The Future of Tabletop Games

by

Greg Borenstein

B.A. Reed College, 2002

M.P.S. New York University, 2011

Submitted to the Program in Media Arts and Sciences, School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of
Master of Science in Media Arts and Sciences
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 2015

Signature of author:____

Program in Media Arts and Sciences
May 8, 2015

Certified by:____

Kevin Slavin
Assistant Professor of Media Arts and Sciences
Thesis Supervisor

Accepted by:____

Pattie Maes, PhD
Academic Head
Program in Media Arts and Sciences

The Future of Tabletop Games

by

Greg Borenstein

Submitted to the Program in Media Arts and Sciences,

School of Architecture and Planning, May 8, 2015 in Partial Fulfillment of the

Requirements for the Degree of Master of Science in Media Arts and Sciences

Abstract

Tabletop games and digital games offer designers divergent affordances and provide players with different kinds of pleasures. Tabletop games excel at creating rich and immediate social interactions. They require players to enact game mechanics themselves resulting in powerful internalization of a game's dynamics. Digital games, on the other hand, can perform complex simulations to let users interact with emergent systems. Where tabletop games require painstaking reading of convoluted written rules, digital games teach their rules through interactive play. They track and adapt dynamically to player actions to ensure competitiveness. And they can connect to the net, allowing remote play, constantly updated content, and even the dynamic integration of real world data. This thesis explores the novel game mechanics and play patterns made possible by including digital devices in a traditional tabletop game setting. It presents a new framework for designing hybrid digital-physical tabletop games based on four areas of focus in which existing digital and tabletop game design practices currently come into conflict: player perception of randomness, the cost of simulation, methods for employing hidden information, and the role of bookkeeping. To illustrate this framework, this thesis describes the design for a novel digital-physical hybrid game called “Sneak”. Sneak is a tabletop stealth game for 2-4 players about deception, evasion, and social intuition. Sneak’s development, playtesting process, and design decisions are used as a test case for validating the described design framework.

Thesis Supervisor: Kevin Slavin

Title: Assistant Professor of Media Arts and Sciences

The Future of Tabletop Games

Greg Borenstein

The following served as a reader for this thesis.

Hiroshi Ishii

Professor of Media Arts and Sciences

MIT Media Lab

The Future of Tabletop Games

Greg Borenstein

The following served as a reader for this thesis.

Frank Lantz
Director
NYU Game Center

Acknowledgements

This thesis would not have been possible without my advisor Kevin Slavin, Rael Dornfest who acted as game design partner and software project manager, my Playful Systems group mates, Che-Wei Wang, Taylor Levy, Julie Legault, Kamal Farah, and, particularly, Jon Bobrow who was incredibly patient and generous with his help on the game's fabrication. I would also like to thank my readers, Hiroshi Ishii and Frank Lantz, for their patience and sage advice and my friends Jack Schulze and Tom Coates for their help in making Sneak visually beautiful, and Elliott Hedman for his advice and welcoming leadership of the tabletop game playing community at the Media Lab. I am endlessly grateful to my family, particularly my dad Mort and my cousin Max, for their loving support and writing advice. Finally, thank you to Kate Tibbetts without whom I would never have come to the Media Lab and who sacrificed more for my work than should have been asked.

Table of Contents

1. Introduction

In December 1961, Steve Russell, a computer scientist in MIT's Computer Science and Artificial Intelligence Laboratory, started work on a new program to demonstrate the graphical capabilities of the lab's PDP-1. The project started from an idea Russell and his peers had been kicking around for a few months, but the MIT hackers had just received a tape with some new sine and cosine routines from DEC, their computer's manufacturer, that would finally make this idea practical.

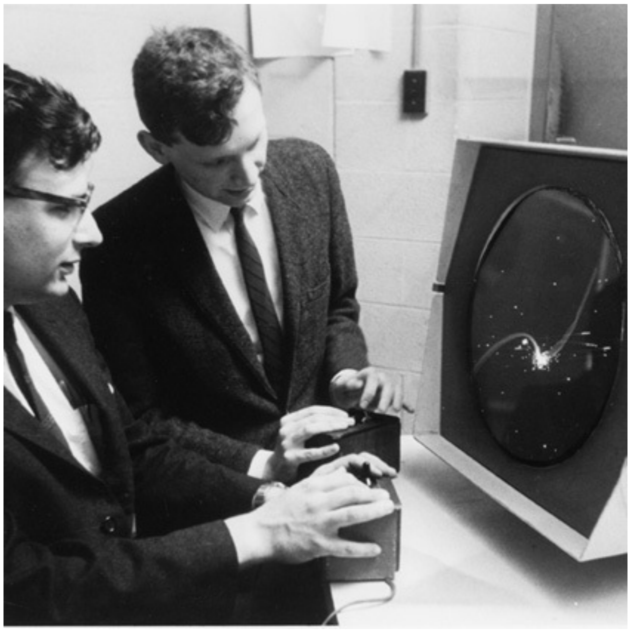


FIG 1-1 STEVE RUSSELL, ITS INVENTOR, PLAYING SPACEWAR, THE FIRST EVER VIDEO GAME, AT MIT IN 1962.

Russell had received the nickname "Slug" by his peers for his slow speed at programming. The coding work took about 200 hours and the program went live in April.

The program Russell created was called "Spacewar!" and it was the first graphical video game of any substance.[1] Spacewar was a local multiplayer game. Each player controlled a spaceship, originally using the computer's front-panel test switches before the engineers built custom-purpose controllers. The spaceships maneuvered around a star firing missiles in an attempt to destroy each other (Fig. 1-1).

The game quickly became a phenomenon around the MIT computer science lab. It spawned endless variants and ports to other computer systems at universities and corporations around the world. It was eventually canonized in Stewart Brand's famous Rolling Stone article, "Spacewar: Fanatic Life and Symbolic Death Among the Computer Bums".[2]

Spacewar (and the other pioneering video games of that time) represented the beginning of a deep fissure in the field of game design. Before Spacewar all parlor games were what we now call tabletop games: games played with cards, dice, cardboard chits, miniature figures, and

[1] John Markoff, "A Long Time Ago, in a Lab Far Away…". The New York Times, February 22, 2002.

[2] Stewart Brand, "Spacewar: Fanatic Life and Symbolic Death Among the Computer Bums". Rolling Stone, December 1972.

other physical pieces played by groups of people sitting around tables. Spacewar inaugurated a new field of digital games, which has, over the intervening 50 years dramatically diverged from its analog predecessors.

It is telling that Spacewar did not begin life as an attempt to translate an existing tabletop game into a digital format. Instead, it started as a benchmark for the graphical and computing capacities of its host platform. And it immediately did things which are rare or downright impossible in tabletop games: simulating the gravity of a star, responding in realtime to the input of multiple players, etc. Even in that nascent form at CSAIL, Spacewar created a social environment quite different from that associated with existing parlor games: “Four intense hours, much frenzy and skilled concerted action, a 15-ring circus in ten different directions, the most bzz-bzz-busy scene I’ve been around since Merry Prankster Acid Tests,” as Stewart Brand described it. In the 40 years since Brand wrote that, video games have grown into a $100 billion worldwide entertainment industry.[1] Simultaneously, they have developed a rich design language of their own including a number of diverse genres providing a variety of different play experiences.

In the last decade or so, tabletop games have also undergone a commercial and creative renaissance. The New German Board Game movement has produced a number of globally successful games including The Settlers of Catan[2], which has sold 25 million copies[3], more than any board game since Monopoly (1933)[4] or Risk (1957).[5]

These two divergent game formats offer designers dramatically different affordances and excel at creating different kinds of social interaction. From Spacewar to SimCity[6], digital games are incredible platforms for simulation, offering players the chance to interact with complex procedural models of real world systems. They also create rich real time interactions giving

[1] Gartner, “Gartner Says Worldwide Video Game Market to Total $93 Billion in 2013”, October 29, 2013. http://www.gartner.com/newsroom/id/2614915

[2] The Settles of Catan, Klaus Teuber. 999 Games. 1995.

[3] Scott Keyes, “Settlers of Catan: How a German Board Game Went Mainstream”. The Atlantic. June 7, 2011. http://www.theatlantic.com/entertainment/archive/2011/06/settlers-of-catan-how-a-german-board-game-went-mainstream/239919/

[4] Monopoly, Elizabeth Magie and Charles Darrow. Parker Brothers. 1933.

[5] Risk, Albert Lamorisse. Parker Brothers. 1957.

[6] Sim City, Will Wrght. Maxis. 1987.

players a visceral experience of master of virtual worlds.[1] They can adapt to player behaviors to create smooth learning curves and balanced competitive environments.

Tabletop games, on the other hand, offer immediate interpersonal interactions and diverse patterns of social interaction. Their modest production costs means they can be created by small groups, iterated rapidly, and distributed cheaply. This has resulted in a shockingly diverse body of games that cover a broad variety of themes and are widely available to a large number of people.

With the rise of portable digital devices, the possibility arrived to combining these two game formats into a new type of hybrid digital-physical tabletop game. Such a game could incorporate the best of both of these two divergent formats, including the systems complexity of digital games and the rich social interaction of tabletop games. It could start to bridge the divide that began back in 1962 when Spacewar lit up its first CRT display.

1.1 Prior Work

This great promise of hybrid digital-physical tabletop games has drawn a number of researchers and there is already a significant pre-existing literature on their design. However, nearly all of this literature focuses on the interface between physical game pieces and the digital platform. This work has explored a wide variety of techniques for integrating physical game pieces with digital tabletop displays. These range from an array of infrared sensors to detect game pieces[2] to simultaneous multi-touch input from multiple players[3] to including RFID tags in game pieces[4] to computer vision and microphone processing.[5]

[1] Steve Swink, Game Feel: A Game Designer's Guide to Virtual Sensation. Morgan Kaufmann. 2008.

[2] Regan L. Mandryk and Diego S. Maranan. 2002. False prophets: exploring hybrid board/video games. In CHI '02 Extended Abstracts on Human Factors in Computing Systems (CHI EA ’02).

[3] van Loenen, Evert, et al. "Entertaible: a solution for social gaming experiences." Tangible Play workshop, IUI Conference. 2007.

[4] Carsten Magerkurth, Maral Memisoglu, Timo Engelke, and Norbert Streitz. 2004. Towards the next generation of tabletop gaming experiences. In Proceedings of Graphics Interface 2004 (GI '04). and Saskia Bakker, Debby Vorstenbosch, Elise van den Hoven, Gerard Hollemans, and Tom Bergman. 2007. Weathergods: tangible interaction in a digital tabletop game. In Proceedings of the 1st international conference on Tangible and embedded interaction (TEI '07).

[5] Hiroshi Ishii, Craig Wisneski, Julian Orbanes, Ben Chun, and Joe Paradiso. 1999. PingPongPlus: design of an athletic-tangible interface for computer-supported cooperative play. In Proceedings of the SIGCHI conference on Human Factors in Computing Systems (CHI '99).

There are also a number of examples of existing commercial projects that use related techniques. Animal Planet: Wildlands by Nukotoys[1] provides physical cards embedded with conductive traces designed to extended the users touch to interact with the capacitive screens of iPads in order to detect and identify cards. The upcoming World of Yo-Ho by Volumique[2] users the accelerometers of a smartphone to track device position on top of a game board.

All of this work focuses on techniques for getting the state of the physical components into the digital system and displaying digital information back to the players rather than on the design of the novel possibilities for game systems in hybrid digital-physical games. Further, many of these systems focus on precisely reproducing existing play patterns from tabletop games in these new hybrid environments. Wallace et al[3] showed a specific example of converting the game Pandemic into a digital from in order to simplifying its user interface.

Some examples of novel game systems do exist in hybrid digital-physical games that make good precedents for the work undertaken here. For example, RealTimeChess[4] presents a novel real time variant of chess enabled by a touchscreen interface. An even better example is Spaceteam[5], a smartphone game played by a series of in-person players over a local network. Players must collaborate to share asymmetrically distributed information under time pressure by shouting at each other. Further, XCOM: The Board Game[6] incorporates a tablet app as a kind of game master to drive and organize the players' real time collaborative decision making.

[1] Animal Planet: Wildlands, Nukotoys. 2014 http://www.nukotoysinc.com/games/wildlands

[2] World of Yo-Ho, Volumique. 2014 http://yoho.io/

[3] James R. Wallace, Joseph Pape, Yu-Ling Betty Chang, Phillip J. McClelland, T.C. Nicholas Graham, Stacey D. Scott, and Mark Hancock. 2012. Exploring automation in digital tabletop board game. In Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work Companion (CSCW '12).

[4] Jonathan Chaboissier and Frederic Vernier. 2009. RealTimeChess: a real-time strategy and multiplayer game for tabletop displays. In Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS '09).

[5] Spaceteam, Henry Smith. Sleeping Beast Games. 2012.

[6] XCOM: The Board Game, Eric Lang. Fantasy Flight. 2015

1.2 Contribution

In their landmark paper, “Build It to Understand It: Ludology Meets Narratology in Game Design Space,”[1] Michael Mateas and Andrew Stern argue that game design is a “wicked problem” in the sense first coined by Rittel and Weber.[2] Wicked problems have no definitive problem statement, they lack a stopping rule, each variation of the problem is unique, and there is no ultimate test by which to judge a solution. In response to the wicked problem of game design, Mateas and Stern propose “a simultaneous process of research and artmaking” as a probe to improve understanding of open questions in the field. They suggest a process including the creation of new games along with careful playtesting to study the effects those games have on players. They specifically identify this approach as being most necessary when exploring new areas of game design (in their case, interactive dramas): “As a wicked problem, only by actually trying to build an interactive drama could we have ever identified this design region.”

This is exactly the approach I have taken in this thesis. This thesis contributes to the emerging field of hybrid tabletop game design in two ways. First, I present the design for a novel hybrid digital-physical tabletop game called “Sneak”. The game demonstrates a new play pattern for integrating digital devices into a tabletop game environment in a way that focuses on the new game systems made possible rather than the interaction between physical components and digital devices. More specifically, it explores the new relationships these hybrid games can create both amongst players and between players and the game’s digital systems.

In addition to this game, I present a new framework for designing hybrid tabletop games. This framework emerged out of the design challenges I discovered doing in preliminary work in this area.[3] The framework presents four areas of game design in which tabletop and digital games diverge: The Perception of Randomness, The Cost of Simulation, The Use of Hidden Information, and The Role of Bookkeeping. It uses the design of Sneak and the lessons learned from the game’s playtesting to explore how the design space of hybrid tabletop games diverges from traditional tabletop and digital games with regards to these questions.

[1] Michael Mateas and Andrew Stern, “Build It to Understand It: Ludology Meets Narratology in Game Design Space”. DiGRA. 2005.

[2] H. Rittel and M. Webber, “Dilemmas in a General Theory of Planning”, in Policy Sciences 4, Elsevier Scientific Publishing, Amsterdam, pp. 155-159, 1973.

[3] Heroes and Villains, Greg Borenstein and Rael Dornfest. Unpublished prototype. 2014.

2. Sneak: A Hybrid Digital-Physical Game About Deception, Stealth, and Social intuition

In this section, I present “Sneak”, the game I designed and implemented as a test case for my design framework. The current Sneak prototype was created through regular playtesting between January and May, 2015. The game consists of a digital app, a set of physical pieces, and a modular board.

After summarizing the game’s premise and gameplay, I will provide a detailed description of its mechanics including setup, turn phases, and victory conditions. I will also describe some of the dynamics that have emerged between players during playtesting sessions. Finally, I will describe the game’s technical implementation.

This summary is meant to provide a basis for the reader to understand the discussion of my design framework in the following chapters. While Sneak’s rules and play experience will be described in some detail, discussion of the underlying design decisions will largely be held for those later chapters where they can be put in the context of that framework.

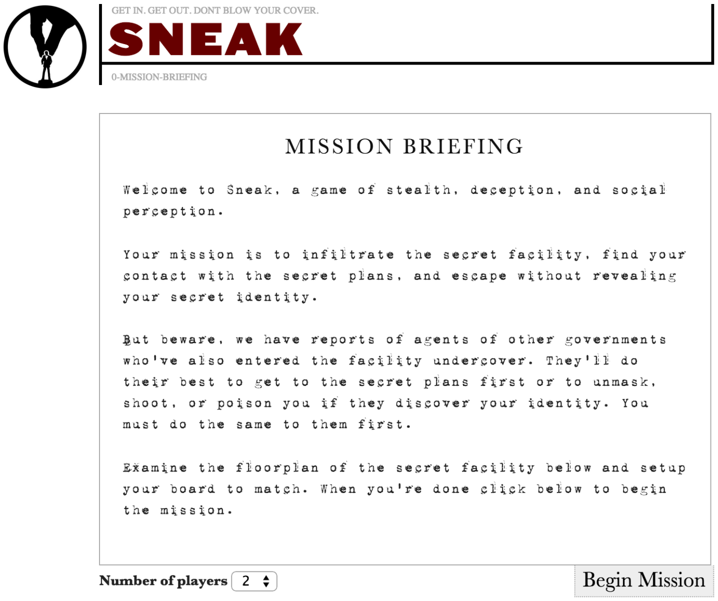
2.1 Overview

Sneak is a hybrid digital tabletop game for 2-4 players about deception, stealth, and social intuition. Each player secretly controls one agent in a procedurally-generated super villain lair. Their mission is to find the secret plans and escape without getting discovered, shot, or poisoned by another player. To accomplish this players must interact and blend in with a series of computer-controlled henchmen while keeping a close eye on their human opponents for any social cues that might reveal their identity.

Players submit their moves through a digital app in order to keep their secret identities private. The app computes the behavior of the non-human henchmen and instructs the players in how to move all of the pieces around the board. Whenever two characters land on the same square, they share their information about who possess the secret plans. This information spreads amongst the human- and computer-controlled characters until one or more human players discovers the identity of the character with the plans. If a player can navigate onto the same square as the character with the plans they will then take possession of the plans. If they can subsequently make it to the “Exit” square safely they win the game.

Players can also win the game by identifying and killing their opponents. Four guns are placed around the board. A player who arrives at a square with a gun can choose to kill any character on the board. If the shot character belongs to a player then they are eliminated from the game. Shooting a gun reveals the character firing the gun as being controlled by a player. Players also each begin the game with one dose of poison. During any turn on which their character occupies the same square as another character, a player can decide to poison that other character. The poisoned character will die after a randomly determined number of subsequent turns. Unlike shooting, poisoning a character does not reveal the killer's identity. Dead characters no longer move or share their information, but their corpses can still be rifled for items (such as the secret plans). If only one player-controlled character remains alive that player is declared the winner.

2.2 Setup



MISSION BRIEFING

Welcome to Sneak, a game of stealth, deception, and social perception.

Your mission is to infiltrate the secret facility, find your contact with the secret plans, and escape without revealing your secret identity.

But beware, we have reports of agents of other governments who've also entered the facility undercover. They'll do their best to get to the secret plans first or to unmask, shoot, or poison you if they discover your identity. You must do the same to them first.

Examine the floorplan of the secret facility below and setup your board to match. When you're done click below to begin the mission.

FIG 2-1. MISSION BRIEFING. THE FIRST SCREEN OF THE SNEAK DIGITAL APP WHERE THE NUMBER OF PLAYERS IS SELECTED.

Players begin the game using the digital app. On the app's starting screen they are presented with a "Mission Briefing" that introduces the premise of the game. They can select the number of players who will be participating and then tap to begin the game (Fig. 2-1).

On the following screen players are presented with a blueprint describing the board layout generated for this specific game (Fig. 2-3). The players follow this blueprint to set up the physical board. Modular wall and door pieces can be snapped into the board's base in order to match the layout shown in the blueprint (Fig. 2-2).

This screen also provides players with the starting positions of all 10 of the game's characters as well as the four guns and the exit. The board is labeled with letters for columns and numbers for rows similar to a chess board. Players place each of the miniature figures on

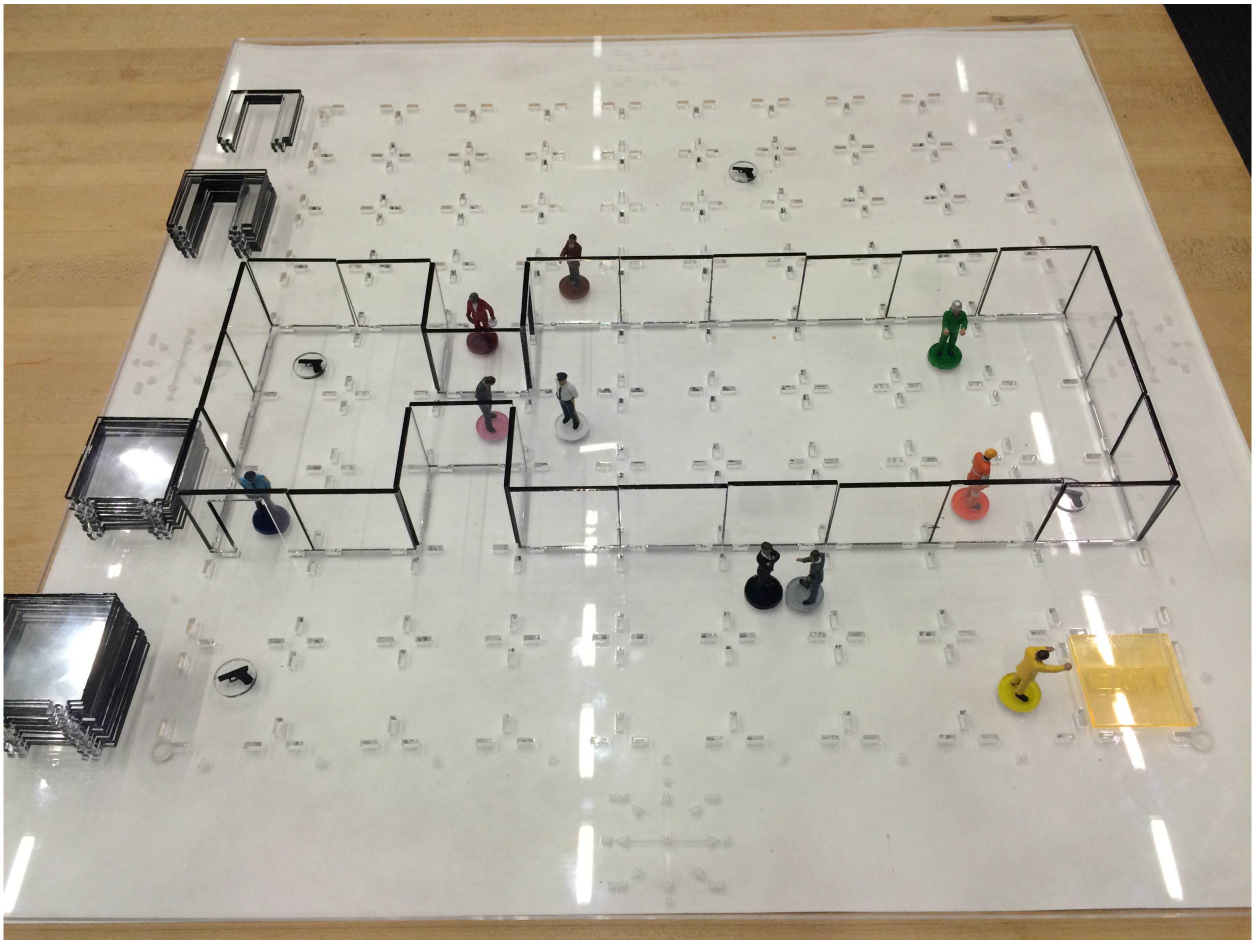


FIG 2-2. A COMPLETE BOARD SHOWING SNAP-FITTED WALL PIECES, ALL 10 CHARACTERS, GUNS, AND THE EXIT SQUARES.

the grid square listed on this screen. They then do the same for the tokens representing the guns and the exit square.

In addition to their colors, characters are each referred to by a military rank. This rank is used to determine who picks up an item when multiple characters arrive at the same square simultaneously (more about this below in the Turn Phases section when we discuss gun pickup). This setup screen presents the players with the order of the ranks as well as an explanation of their use.

2.3 Turn Phases

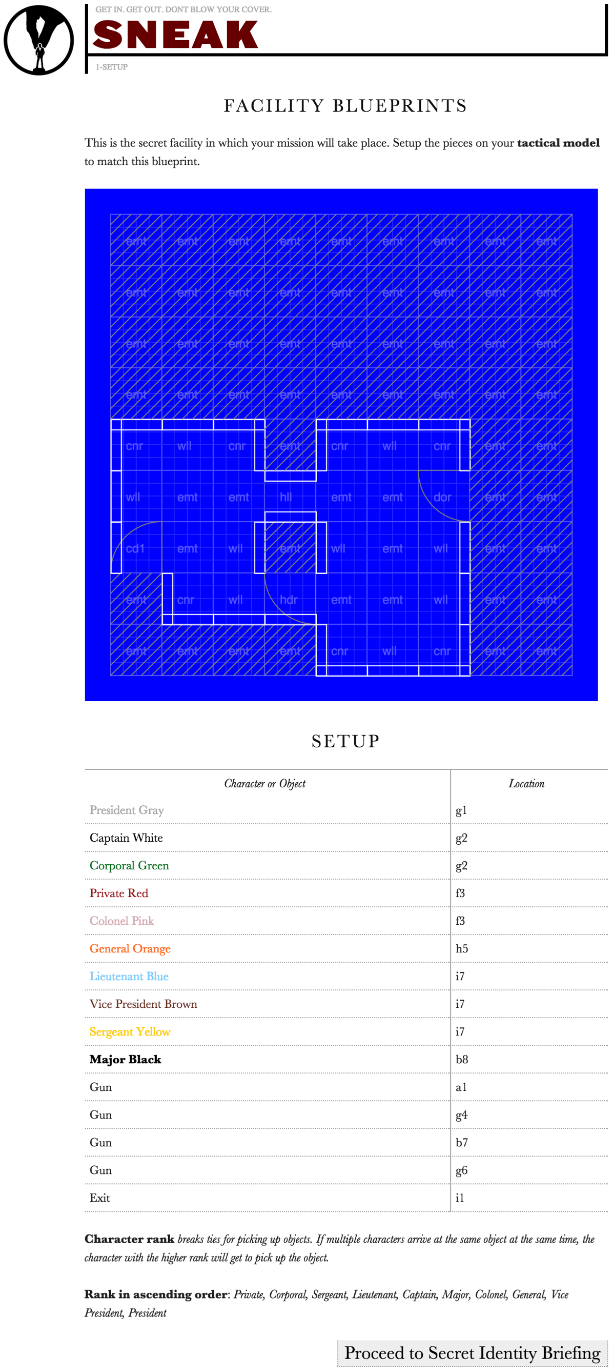
After the players have set up the board play begins. Each turn of the game consists of three phases: action selection, move resolution, and information review.

The first phase of each turn is action selection. During this phase, each player is presented with an interface for selecting their character's next action (Fig. 2-5). Depending on the context three different types of actions are available to players: movement, shooting, and poisoning.

On most turns, players can choose to move their character one square in any of the cardinal directions or hold at their square (more about the exceptions shortly). Their move options are limited by the layout of the board since characters cannot move through walls. Each player sits on one of the four sides of the board and so views the board from a different orientation. The move input buttons are arranged to match this view and the board includes a compass rose on each side to help players orient themselves (Fig. 2-7).

Beyond movement, this input screen also allows players to shoot and poison other characters when game conditions permit.

If a player lands on a square containing a gun they are presented with the shooting interface instead of the movement interface. In the shooting interface players can select whether to drop the gun or fire it. If they choose to fire they can then pick the character they'd like to target. Their target will be killed this turn and their death revealed in the resolution phase. Each gun on the board can only be used once and is removed from the board after being fired.



GET IN. GET OUT. DONT BLOW YOUR COVER.

SNEAK

1-SETUP

FACILITY BLUEPRINTS

This is the secret facility in which your mission will take place. Setup the pieces on your **tactical model** to match this blueprint.

emt	emt	emt	emt	emt	emt	emt	emt	emt
emt	emt	emt	emt	emt	emt	emt	emt	emt
emt	emt	emt	emt	emt	emt	emt	emt	emt
emt	emt	emt	emt	emt	emt	emt	emt	emt
cnr	wll	cnr	emt	cnr	wll	cnr	emt	emt
wll	emt	emt	hll	emt	emt	dor	emt	emt
cd1	emt	wll	emt	wll	emt	wll	emt	emt
emt	cnr	wll	hdr	emt	emt	wll	emt	emt
emt	emt	emt	emt	cnr	wll	cnr	emt	emt

SETUP

Character or Object	*Location*
President Gray	g1
Captain White	g2
Corporal Green	g2
Private Red	f3
Colonel Pink	f3
General Orange	h5
Lieutenant Blue	i7
Vice President Brown	i7
Sergeant Yellow	i7
Major Black	b8
Gun	a1
Gun	g4
Gun	b7
Gun	g6
Exit	i1

Character rank *breaks ties for picking up objects. If multiple characters arrive at the same object at the same time, the character with the higher rank will get to pick up the object.*

Rank in ascending order: *Private, Corporal, Sergeant, Lieutenant, Captain, Major, Colonel, General, Vice President, President*

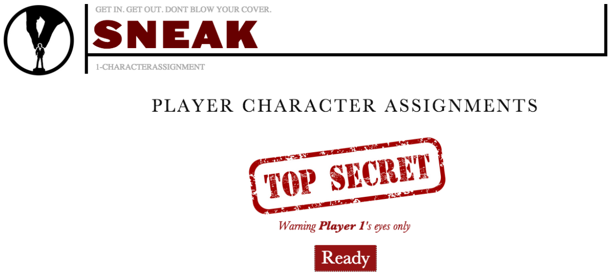
Proceed to Secret Identity Briefing

FIG 2-3. PLAYERS ARE SHOWN A BLUEPRINT TO CONSTRUCT THE BOARD AS WELL AS INSTRUCTIONS FOR PLACING ALL CHARACTERS AND ITEMS.

If players choose to drop the gun instead of firing they hold still on their current square until the next round. Non-player characters (NPCs) who land on a gun square also pick up guns and hold for a turn, but do not fire.

If a player finds themselves on the same square as another character, they can select to secretly poison that character. The poisoned character will then die after a randomly determined number of turns.[1] Each player can only poison one character in the course of any game. The poison interface shows up on the same screen as the move input interface when one or more characters are present on the same square as the current player and the player has not yet used their poison. When players poison a character they also hold instead of moving for that turn.

After all of the players have submitted their actions for the turn, the game enters the move resolution phase. The app presents the players with one move instruction for each character (Fig. 2-6). Move instructions are given in cardinal directions and characters who pick up guns are noted. In addition to the move instruction text, the app generates spoken audio instructions so that the players can move the characters without needing to look back-and-forth between the device and the board.[2]



GET IN. GET OUT. DONT BLOW YOUR COVER.

SNEAK

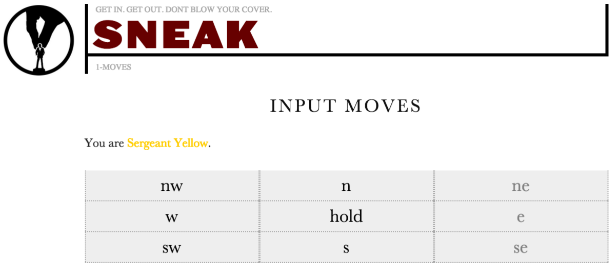
1-CHARACTERASSIGNMENT

PLAYER CHARACTER ASSIGNMENTS

TOP SECRET

*Warning **Player 1**'s eyes only*

FIG 2-4 THE PASS SCREEN IS USED TO KEEP PRIVATE INFORMATION SECRET FROM OTHER PLAYERS.



GET IN. GET OUT. DONT BLOW YOUR COVER.

SNEAK

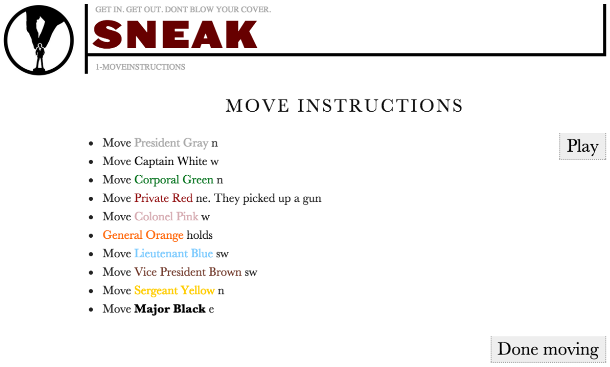
1-MOVES

INPUT MOVES

You are Sergeant Yellow.

nw	n	ne
w	hold	e
sw	s	se

FIG 2-5 THE MOVE INPUT SCREEN IS ALWAYS SHOWN ORIENTED FOR THE CURRENT PLAYER.



GET IN. GET OUT. DONT BLOW YOUR COVER.

SNEAK

1-MOVEINSTRUCTIONS

MOVE INSTRUCTIONS

- Move President Gray n
- Move Captain White w
- Move Corporal Green n
- Move Private Red ne. They picked up a gun
- Move Colonel Pink w
- General Orange holds
- Move Lieutenant Blue sw
- Move Vice President Brown sw
- Move Sergeant Yellow n
- Move **Major Black** e

FIG 2-6 THE MOVE RESOLUTION SCREEN INSTRUCTS THE PLAYERS IN HOW TO MOVE THE CHARACTERS.

[1] Between two and five turns in the current prototype as of this writing, though this value may be tuned through further playtesting.

[2] See https://vimeo.com/124847236 for documentation of the audio instructions.

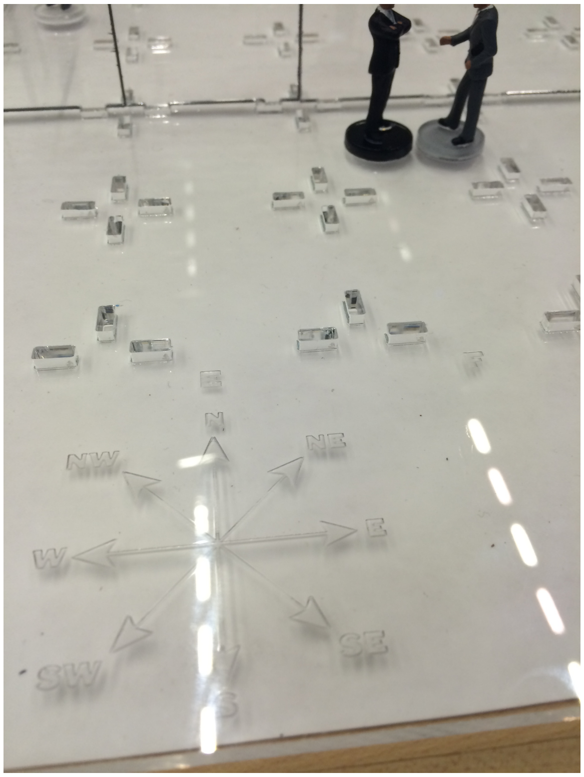
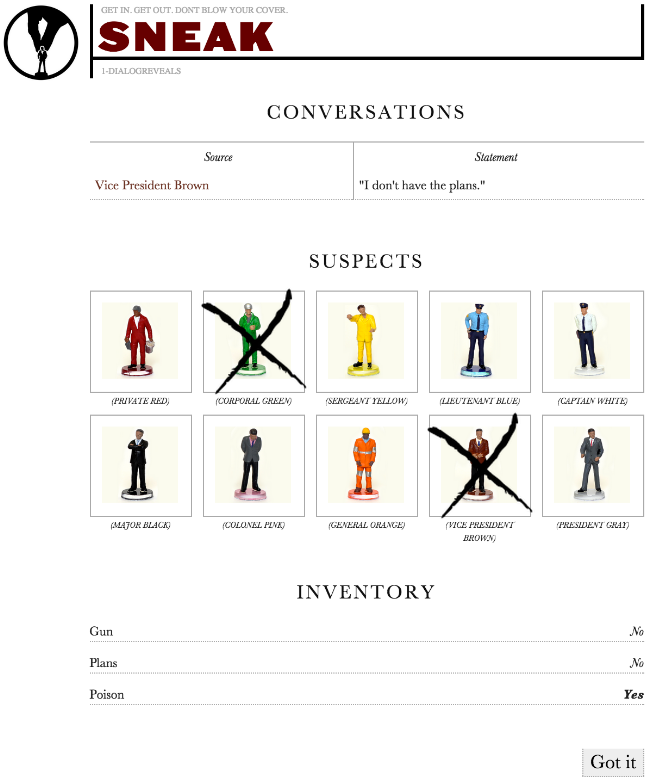


FIG 2-7 EACH SIDE OF THE BOARD HAS A COMPASS ORIENTED FOR THAT PLAYER.



GET IN. GET OUT. DONT BLOW YOUR COVER.

SNEAK

1-DIALOGREVEALS

CONVERSATIONS

Source	*Statement*
Vice President Brown	"I don't have the plans."

SUSPECTS

(PRIVATE RED) (CORPORAL GREEN) (SERGEANT YELLOW) (LIEUTENANT BLUE) (CAPTAIN WHITE)

(MAJOR BLACK) (COLONEL PINK) (GENERAL ORANGE) (VICE PRESIDENT BROWN) (PRESIDENT GRAY)

INVENTORY

Gun	*No*
Plans	*No*
Poison	***Yes***

Got it

FIG 2-8 THE INFORMATION REVIEW SCREEN SHOWS PLAYERS DIALOG RESULTS AND INVENTORY.

If any characters have been killed this turn, either by shooting or poison, that is also revealed at this point. When a character is shot the player-controlled character who fired the gun is also disclosed.[1] If a player has won the game by killing their last surviving opponent the app declares that victory here.

The players follow the instructions to move all of the characters to their new positions. The characters controlled by the players move according to the directions submitted by the players. The NPCs move according to pathfinding and destination selection algorithms implemented in the app's software (see later chapters on The Perception of Randomness and The Cost of Simulation below for a thorough discussions of these algorithms).

Once all characters have arrived at their new squares on the board, the players tap to move onto the information review phase. This phase relates to one of the game's two victory conditions: finding and escaping with the secret plans. At the start of the game one NPC is randomly assigned the plans. Whenever two characters arrive at the same square they engage

[1] The player controlling the character who fired is not revealed by the app. Obviously in a two player game the identity of the shooter will be immediately known, but in three or four player games the ambiguity remains and interesting dynamics can result.

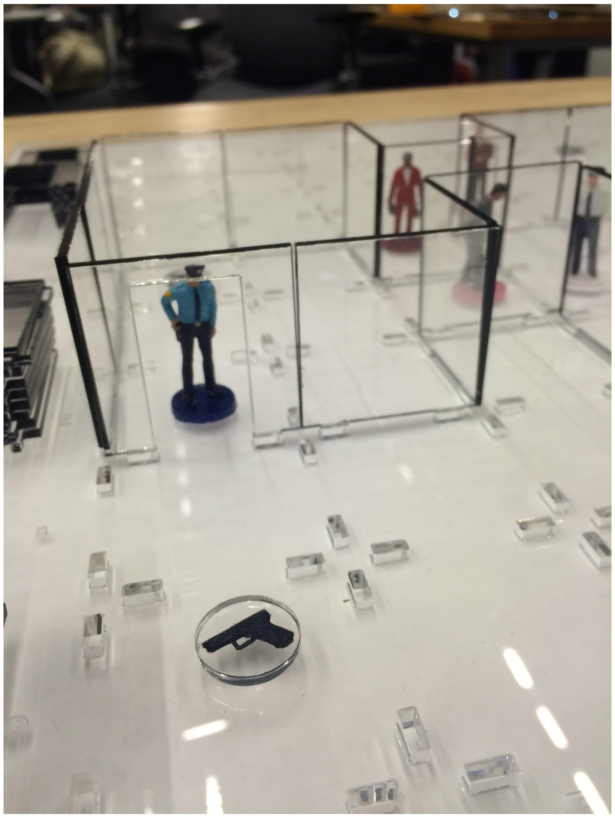


FIG 2-9 EXAMPLE GUN PLACEMENT TOKEN

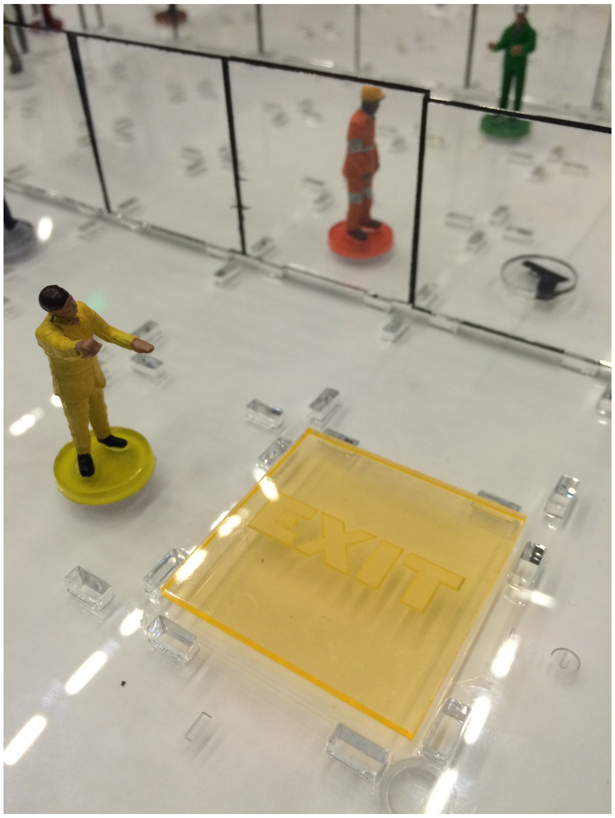


FIG 2-10 PIECE MARKING EXIT SQUARE

in a dialog during which knowledge is transferred between them. Through these dialogs each characters learns everything their dialog partner knew about the possession of the plans: whether or not they themselves have them as well as which other characters they previously encountered and whether or not those characters had the plans. While the game starts with each character only knowing about their own possession of the plans (or lack thereof) information propagates quickly as dialogs occur.

The information review phase of each turn begins by summarizing all of the current dialogs occurring on the board. A dialog occurs between any two characters who are on the same square. If more than two characters are on the same square a dialog occurs between each possible pair of characters.

Next, each player is presented with a screen showing their dialogs for the round as well as a summary of their current knowledge about which characters may have the plans (Fig. 2-8).[1] This screen also reminds the player of whether or not they currently have a gun, a dose of poison, or the plans in their possession.

[1] As you can see in Fig. 2-8, characters that are known not to have the plans are marked with an ‘x’. Not shown here, characters that have been reported to have the plans are circled.

Once all players have seen their information review screen, the turn is over and the next turn begins at the action input phase.

For the action input and information review phases it is important that each player see the game interface in secret. If other players saw these screens they would find out the player's character identity or learn private information about the location of the secret plans. While Sneak may eventually be played using multiple mobile devices. The current prototype uses a single device which is passed back-and-forth between the players.[1] Hence, between each player's private interface screen, the game presents a "pass screen" (Fig. 2-4). This pass screen returns the app to a neutral state that does not reveal any private information as well as telling the players who should receive the device next.

After the device has been passed to the next player, the player who received it taps the "done" button to visit the appropriate private interface.

2.4 Victory Conditions

As mentioned in the overview, there are two ways for players to win at Sneak. They can find the secret plans and escape to the exit or they can identify and kill all of the other players. Players pursue this first objective by participating in dialogs with other characters in order to discover who has the plans. They pursue the second objective by watching the behavior of their fellow players (both the social cues they emit around the board as well as the quality of their moves on the board).

When the game begins one of the NPCs is randomly assigned possession of the plans. As the game progresses knowledge about which character has the plans spreads to the other NPCs and to the players through dialogs. Players identify characters that might have the plans and attempt to intersect with them. Eventually, one of the players successfully finds the plans-holder and takes possession of them from him. If that player can then reach the exit square before they are killed by any of the other players, they win the game.

Players can identify human-controlled characters in two ways. First, they can observe the movement of characters on the board and notice behavior that stands out from the algorithmic

[1] In addition to aiding prototyping, this single device mode has a number of inherent advantages. It does not require a network connection and, most importantly, it means that players not currently moving can keep their focus on the other players rather than being distracted by other apps on their mobile device.

movement patterns of the other characters. As will be discussed extensively in later chapters, NPCs move according to a series of algorithms for selecting destinations, navigating around the map, and even balancing the competitiveness of the game. Attentive players will gradually learn to understand these behaviors and be able to detect character movements that don't conform to their logic. Such movements can reveal that a particular character is controlled by a player and lead a competitor to shoot or poison them.

Secondly, players can determine the identity of their competitors through social cues. Players watch each other input moves, react to character movement, and receive information. They also talk about the game while it progresses. All of these interactions yield subtle hints that players use to form theories about which of their opponents control which characters.[1] And, of course, far more obvious hints occur when players shoot characters revealing their identity.

During the course of the design process I also experimented with two additional game mechanics designed to aid player identification. Neither of these have yet made it into the current prototype though for different reasons in both cases. These are Spoken Dialogs and Character Bugs. I will discuss both of these mechanics extensively (particularly why the first was excluded despite being prototyped and the second should be a high priority for future design work) in the chapter on The Role of Bookkeeping.

2.5 Game Dynamics

In order to give the reader a more complete picture of Sneak this section describes the dynamics that emerge during a typical game.

The term "dynamic" is used here in the sense defined by Hunicke, Le Blanc, and Zubek in their landmark paper "MDA: A Formal Approach to Game Design and Game Research".[2] The MDA framework divides game design into three components: mechanics, aesthetics, and dynamics. Mechanics describes the "particular components of the game": the rules, data, and algorithms, on which we have thus far focused in our discussion of Sneak. Aesthetics describes

[1] This was a common experience reported during playtesting. See Appendix A containing my playtesting notes for a number of examples.

[2] Robin Hunicke, Marc LeBlanc, and Robert Zubek. "MDA: A Formal Approach to Game Design and Game Research." Proceedings of the Challenges in Games AI Workshop, Nineteenth National Conference of Artificial Intelligence. 2004

the “desirable emotional responses evoked in the player” during play, which will be the focus of much of the following three chapters. Dynamics “describes the run-time behavior of the mechanics acting on player inputs and each others’ outputs over time.” A game’s dynamics emerge from the strategies concocted by players and the way players interact with each other and with games systems in order to pursue these strategies.

Sneak is a competitive game. It pits the players against each other in a winner-takes-all format. Hence, its dynamics center on the strategies players use to defeat each other. In order to illustrate these dynamics I’ll describe a specific games that occurred during playtesting.[1]

The game was held on March 9, 2015. Its participants were two volunteer MIT undergraduates. For the first few turns, both players moved their characters around the board attempting to encounter as many other characters as possible. After turn eight, one of the players became convinced that he had figured out the other player’s identity. “Oh, you’re orange!”, he announced after a movement phase. He proceeded to move his character onto a gun and shoot the orange character. But in the movement phase the game revealed that orange had been an NPC. The second player was, in fact, controlling the white character and had already acquired the plans and was only a few squares from the exit. Player two was able to move the few squares left to the exit to win before player one could come close to reaching a second gun to fire again. The play session took about 20 minutes.

In a post-game interview, player one explained that he’d narrowed down player one’s identity to two candidates based on the way characters moved on the board. He didn’t think a player would move towards the edge of the board and stay still for long. He assumed that the other player would employee a strategy similar to his own of attempting to engage in as many dialogs as possible with other characters in order to gain information. Once a sufficient number of turns had passed and he had failed to discover which character had the plans, he became nervous that the other player might be near victory. Hence, he decided to shoot one of his suspects despite not being positive of his deduction.

This game illustrates the “three act” structure that generally characterizes the dynamics of a game of sneak. At the start of the game, neither player has any information about either the plans or their opponents’ identity. They spend the first turns attempting to engage in as many dialogs as possible in order to find the location of the plans.

After five to 10 turns have passed, the second act begins. By this point, at least one of the players knows which NPC has the plans and sometimes one of the players has even acquired

[1] For detailed playtesting notes, see Appendix A.

the plans. Players who do not know the location of the plans begin to become nervous that they are falling behind. Both players have developed suspicions about their opponent's identity, often narrowing the field down to two or three suspects.

At this point the game enters its final act. If a player has the plans, they attempt to subtly make their way towards the exit. If one or more players has identified the NPC carrying the plans, they each race to chase that NPC down. Players who have not discovered the location of the plans try to narrow down their suspects and start to think about taking a shot at them. The game resolves when one player reaches the exit with the plans or by an exchange of gun fire.

2.6 Technical Implementation

The current Sneak prototype contains extensive software and hardware components. The software was implemented using web technologies for speed of development, flexibility, and availability on a range of mobile devices. The hardware was created using laser cut components for the board base, modular walls, and gun and exit tokens. The character pieces were created using modified O-scale model train figures. In this section, I'll discuss the specifics of each of these components as they stand in the current prototype.

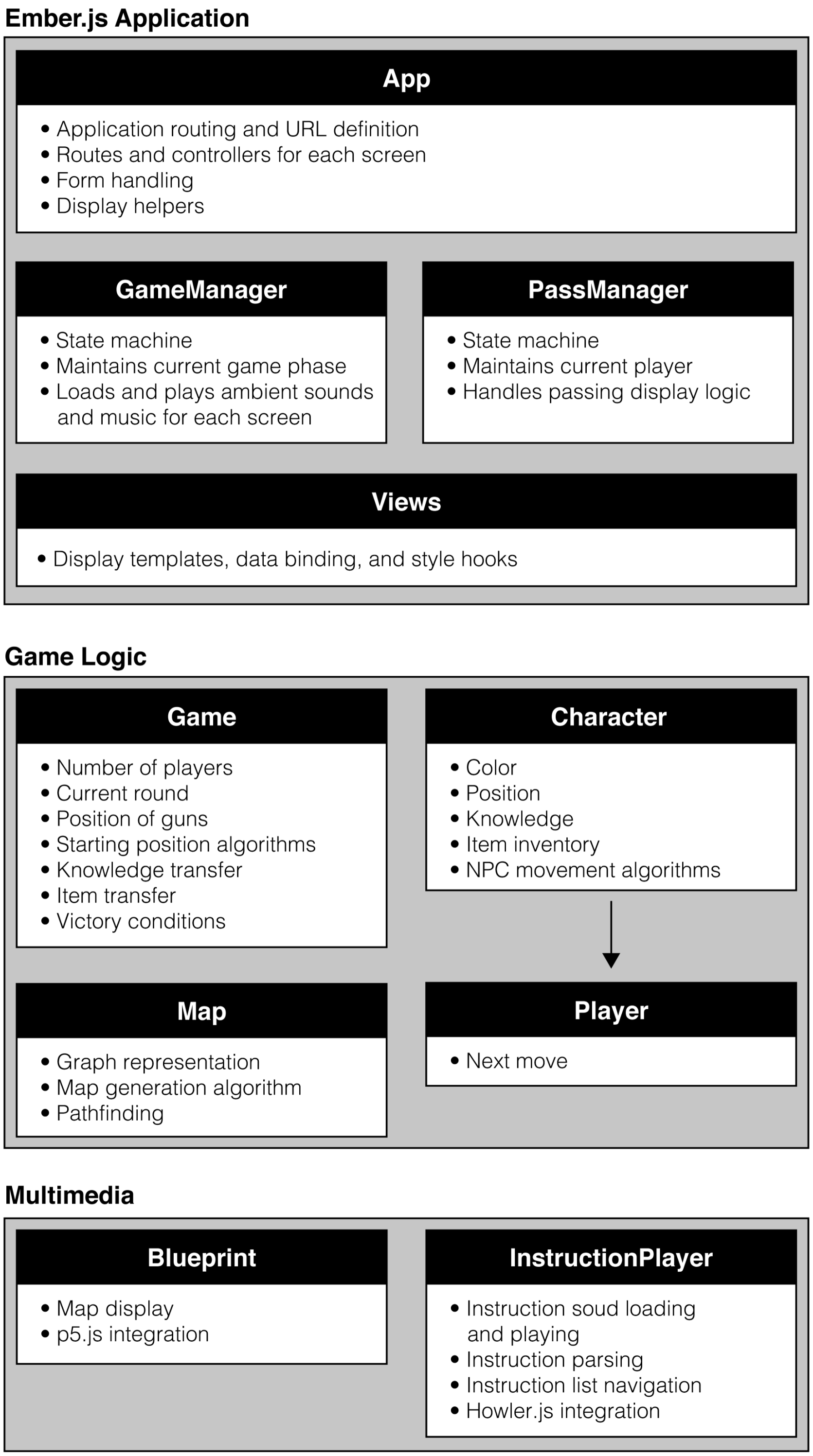
Sneak's current software[1] is a client-side web application implemented using the Ember.js javascript framework.[2] The game logic is implemented in a series of plain javascript objects representing entities such as players, characters, maps, and the state of the current game (Fig. 2-9). These objects are responsible for maintaining character and game state as well as map generation and pathfinding algorithms. Both pathfinding and map generation use a simple adjacency graph for the squares. Pathfinding is accomplished using depth-first search over this graph.

The flow of the game's interface is implemented in a pair of state machines managing turn phase and player turns (GameManager and PassManager). The HTML and CSS are constructed in a series of views using Handlebar templates[3] per the Ember.js framework.

[1] All of Sneak's software is available in this git repository: https://github.com/atduskgreg/sneakgame

[2] http://emberjs.com/

[3] http://handlebarsjs.com/



Ember.js Application

App

- Application routing and URL definition
- Routes and controllers for each screen
- Form handling
- Display helpers

GameManager

- State machine
- Maintains current game phase
- Loads and plays ambient sounds and music for each screen

PassManager

- State machine
- Maintains current player
- Handles passing display logic

Views

- Display templates, data binding, and style hooks

Game Logic

Game

- Number of players
- Current round
- Position of guns
- Starting position algorithms
- Knowledge transfer
- Item transfer
- Victory conditions

Character

- Color
- Position
- Knowledge
- Item inventory
- NPC movement algorithms

Map

- Graph representation
- Map generation algorithm
- Pathfinding

Player

- Next move

Multimedia

Blueprint

- Map display
- p5.js integration

InstructionPlayer

- Instruction soud loading and playing
- Instruction parsing
- Instruction list navigation
- Howler.js integration

FIG 2-11 SOFTWARE ARCHITECTURE OF THE CURRENT PROTOTYPE. ALL PORTIONS OF THE APPLICATIONS ARE IMPLEMENTED IN CLIENT-SIDE JAVASCRIPT.

The game includes a blueprint-style display of each game's procedurally generated map. This is accomplished using the p5.js javascript framework.[1] This framework acts as an abstraction above browser canvas APIs. The small javascript application written in it receives the Map data structure representing the layout of the current game and uses that to display a blueprint-style map (Fig 2-3.).

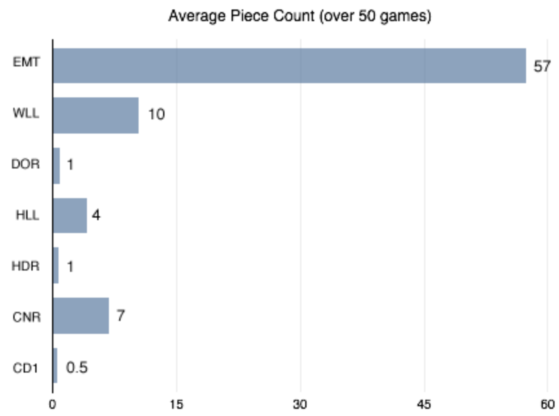


FIG 2-12 AVERAGE PIECE USAGE COUNT FOR EACH SQUARE CALCULATED OVER 50 SIMULATED GAMES.

Finally, the game uses the Howler.js[2] library for cross-browser HTML 5 audio playback. Howler is integrated in two different places in the application. The GameManager uses it to play music and background ambience during appropriate phases of the game. The InstructionPlayer uses it to play spoken audio versions of move instruction. In the latter case, textual versions of these instructions are parsed out of the DOM and sequences of audio recordings are assembled into complete instructions for each player.[3] This limits the number of audio files that are necessary for the game and reduces page load times and audio buffering.

The chief technical challenge for Sneak's hardware is to provide a physical board with modular wall pieces that can be easily and quickly assembled by the players at the start of each game to match the blueprint outputted by the procedural map generation algorithm. After experimenting with a number of different approaches (see the chapter on The Materialization of Theme for a discussion of these alternatives) I arrived at a system that uses laser cut plexiglass to create wall and door pieces that snap fit into a grid of holes in a base (Fig. 2-13 and Fig. 2-14).

This method was selected after analysis of the distribution of walls and corners on a typical board generated by the map creation algorithm (Fig. 2-12). That analysis showed that squares without walls dramatically outnumbered those that had one or more walls. Thus, any physical

[1] http://p5js.org/

[2] https://github.com/goldfire/howler.js/

[3] Individual audio segments were recorded for each different color character moving and holding. The cardinal directions were recorded separately. The two appropriate pieces are then assembled by the InstructionPlayer as needed.

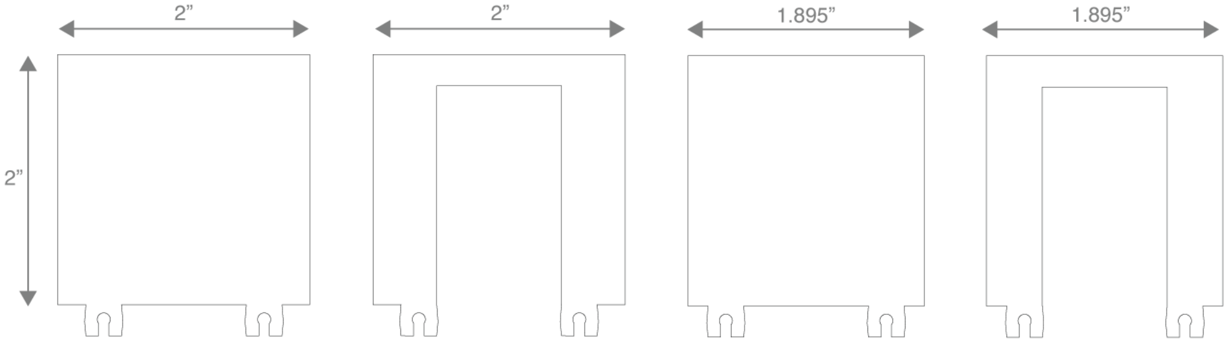


FIG 2-13 WALL AND DOOR PIECE DESIGN FOR LASERCUT WITH SNAP FIT FEET. BOTH DOOR AND WALL PIECES COME IN TWO DIFFERENT WIDTHS TO ALLOW FOR INTERSECTIONS AT CORNERS.

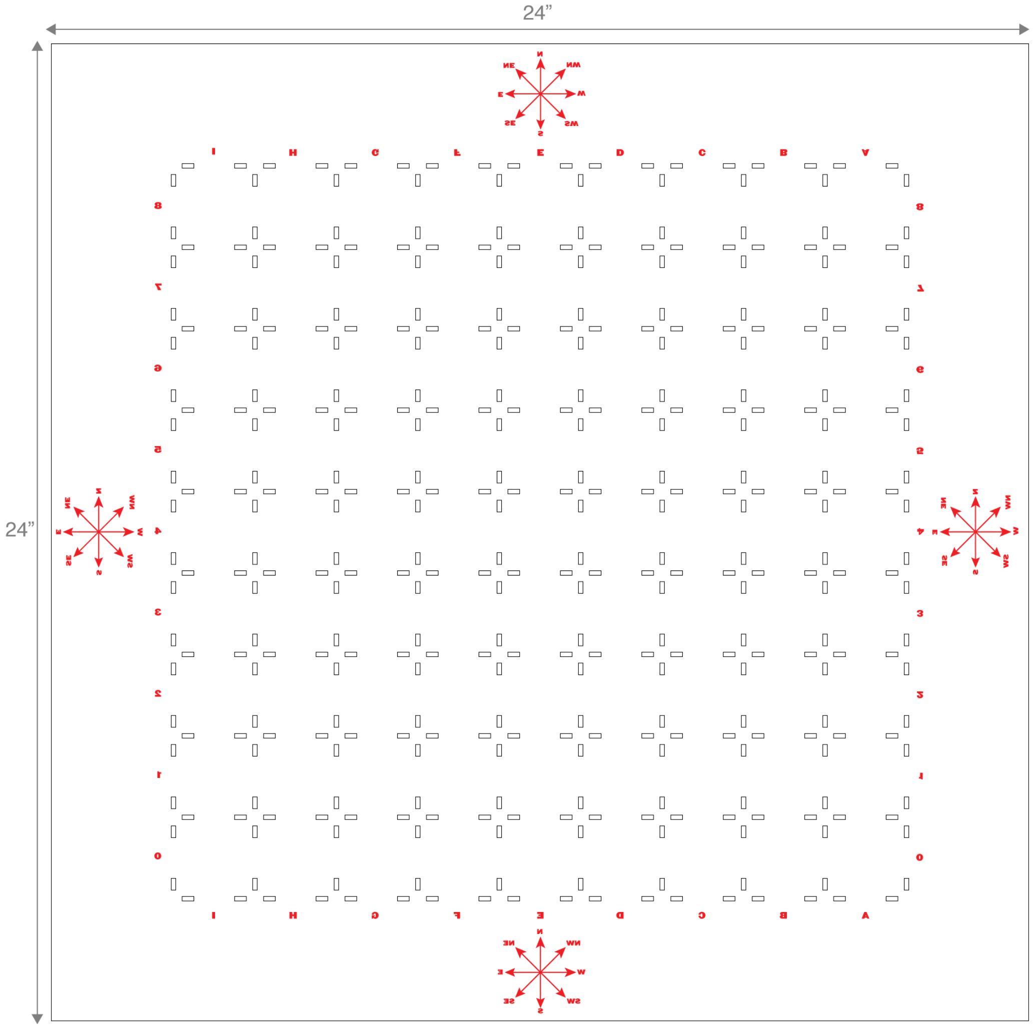


FIG 2-14 DESIGN FOR BOARD BASE. INCLUDES LASER CUT HOLES TO RECEIVE SNAP FIT FEET AS WELL AS COMPASS ROSES AND ROW AND COLUMN LABELS WITH INVERTED TEXT TO BE ETCHED ON THE UNDERSIDE OF THE BOARD.

design that required placing pieces on these empty squares would increase player setup time dramatically. After this analysis, I quickly settled on the snap fit design described here which required no pieces to be used on empty squares and hence minimized the number of pieces needed to set up any given board while still providing the desired aesthetic.

To implement this approach I designed a foot mechanism that allows each vertical piece to snap fit into a pair of adjacent holes in the base. Each foot contains a small cut and a round gap to allow the foot to flex enough to snap into place (Fig. 2-13). The current dimensions of this foot mechanism were determined through extensive trial and error to find measurements that produced a firm fit without breaking the plexi during insertion or being too difficult for players to insert or remove.

I cut the wall and door pieces out of clear 1/8” plexi and the base out of clear 1/4” plexi. In order to deal with corners, I provided two variations of both the door and wall pieces. The standard pieces are 2” wide to match the base grid. The second set of pieces are 1.895” wide in order to allow for the thickness of the plexi where pieces abut (Fig. 2-13).

In addition to holes cut to receive the wall and door pieces, the base also includes four compass roses, one on each side of the board for each of the four potential players, as well as row and column labels (Fig. 2-14). The text for these was oriented to face each respective p layer and inverted so it could be etched on the underside of the base’s clear plexi.

For the character figures, I repurposed O-scale model train figures. I needed figures in 10 different colors to match the colors used in the software, so I selected and repainted existing figures to match (Fig. 2-15). I also laser cut clear plastic bases and painted them with the matching color for each figure.

Finally the gun and exit tokens (Fig. 2-9 and Fig. 2-10) were also cut and etched out of plexi with the gun icon being painted on the inverse side to make it more visible.

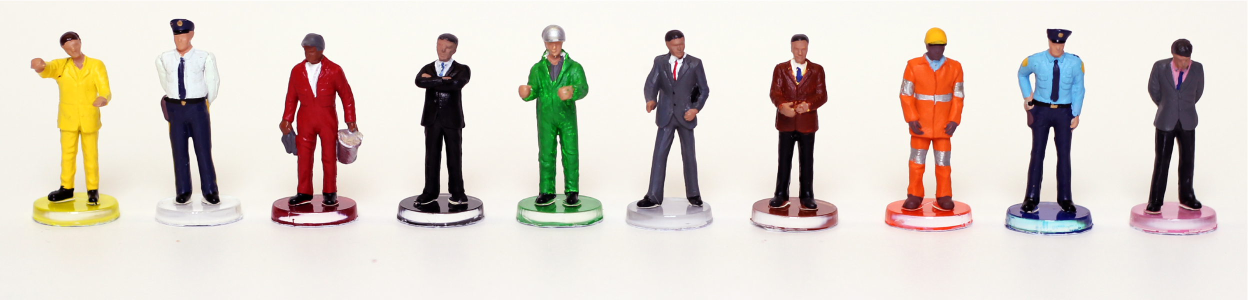


FIG. 2-15 CHARACTER FIGURES REPURPOSED AND REPAINTED FROM O-SCALE MODEL TRAIN FIGURES.

3. The Four Questions: A Framework for Hybrid Tabletop Game Design

My preliminary investigations into the field of hybrid tabletop games began in the summer of 2014 with a collaboration with board game designer Rael Dornfest. Over the course of a two week prototyping session we designed Heroes and Villains, a collaborative two-player real time press your luck game with a superhero theme (Fig. 1).

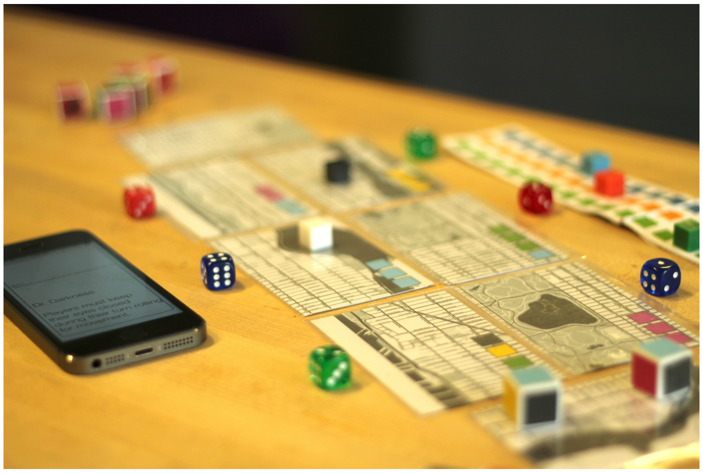


FIG 3-1 HEROES AND VILLAINS, A HYBRID DIGITAL-PHYSICAL GAME EXPLORING THE PERCEPTION OF RANDOMNESS AND THE USE OF REAL TIME MECHANICS.

During this prototyping session, a series of tensions between the methods and goals of traditional tabletop and video game design repeatedly emerged. After analyzing these tensions and making a number of additional prototypes, I came up with a design framework based on four key areas in which the design languages of digital and tabletop games come into conflict when designing a hybrid digital-physical game.

These areas are: the perception of randomness, the cost of simulation, the use of hidden information, and the role of bookkeeping. In each of these areas, the cognitive limitations of players, their in-built biases, and the affordances of digital systems and physical components intersect to create a series of challenges and opportunities for designers of digital tabletop games.

In the following four chapters I explore these questions. In each chapter I begin with a discussion of the different ways these game design elements work in traditional tabletop and digital games. Then I describe the possibility space they present for hybrid games and communicate what I learned about each of these questions in designing and playtesting Sneak.

4. The Perception of Randomness

The vast majority of games integrate randomness in one form or another. From dice rolls to shuffled decks of cards to spinning wheels, luck has been part of play since our earliest games.[1] Use of randomness ranges from the simple — take Monopoly[2] where a player's movement is entirely determined by a series of die rolls — to the extremely complex (think of Dungeons and Dragons's[3] use of sophisticated rolls with multiple dice to model the likelihood of uncertain events).

Considering solely the odds of their outcome, there is no appreciable difference between rolling a six-sided die and generating a random number in the same range through software. However, from the player's perspective, these events feel extremely different, particularly with regard to where they place the blame or credit for the effect of the random number on the result of the game. In games requiring the rolling of physical dice it is impossible for players not to attribute agency to the player who actually rolls the die.

For example, this is the central basis of the design of the common casino game, Craps. In Craps, a single "shooter" conducts a sequence of rolls with a pair of dice while other players bet on the outcome. Even though individual shooters do not influence the result of the roll, players cannot help but blame each other for the outcomes of their rolls, attributing skill or hot streaks to a particular shooter.

On the other hand when the random number is generated by a computer, players attribute it to the systems or characters portrayed by the game rather than their own actions. For example, in the Medieval dynasty simulation game Crusader Kings II,[4] the heritable characteristics of a player's children are determined partially by random number generation and players perceive them as the result of fate or genetics rather than their own actions.

These differences in the perception of randomness make the choice of mechanics for random number generation within a digital tabletop game highly expressive, allowing a designer to encourage players to blame or credit themselves for results of some numbers and feel that others are imposed on them by the game's world, characters, or events.

[1] Kevin Slavin, "Debunking Luck", Pop Tech 2013. https://vimeo.com/78829799

[2] Monopoly, Elizabeth Magie and Charles Darrow. Parker Brothers. 1933

[3] Dungeons & Dragons, Gary Gygax and Dave Arneson. TSR. 1974

[4] Crusader Kings II, Henrik Fåhraeus. Paradox Interactive. 2012

A hybrid digital-physical game finds itself in an interesting position in relation to this issue. Nearly all of the mechanics for generating random numbers in tabletop games rely on players manipulating physical tokens in some way: drawing a card from a shuffled deck, rolling dice, spinning a wheel, flipping a coin, etc. A hybrid game that uses such physical random number generators is presented with the problem of synchronization: physically-generated random numbers must be communicated to the digital app in some way.

Such a game could use some technological fix in order to transmit the random number to the app. For example, “Dice+” is a bluetooth dice product[1] specifically designed to transmit the result of dice rolls to tablet and smartphone games. Computer vision-based approaches for detecting play cards have also been demonstrated, for example by Martins, Reis, and Teófilo.[2] Unfortunately, such fixes are expensive and constraining. Bluetooth dice currently cost upwards of 40 dollars as opposed to 10 cents for the conventional version. Computer vision solutions depend on adequate lighting, a clean background, and high quality cameras, materials that may not all always be available in every play context and can make for rigid play setups even when available.[3]

Another potential alternative beyond automatic sensing of dice and cards is to require players to enter card and dice results into the application through some kind of interface. This is the approach taken by pioneering hybrid digital-physical board game “XCOM: The Board Game”.[4] The significant downside of this approach is the duplication of bookkeeping effort required. Random numbers must be generated using cards or dice rolls and then separately entered into the device. This problem will be discussed extensively in the chapter on The Role of Bookkeeping.

The final and simplest option is to provide an on-screen interface which players publicly and explicitly tap in order to receive a random number. This can be thought of as the skeuomorphic solution, designing a digital interface to reproduce in detail the workings of its physical

[1] http://dicepl.us/

[2] Paulo Martins, Luís Paulo Reis, and Luís Teófilo, “Poker Vision: Playing Cards and Chips Identification based on Image Processing”. Pattern Recognition and Image Analysis Conference. 2011

[3] For example Osmo is a computer vision-based play platform for tablets. It relies on the tablet standing upright in supporting hardware on a stable surface so that its mirror mechanism can point the tablet’s camera at the compatible play pieces. https://www.playosmo.com

[4] XCOM: The Board Game, Eric M. Lang. Fantasy Flight. 2015

analogue.[1] This approach may be the most promising of the three listed here. It is one I used in a prototype game in the early phases of this research.[2]

Sneak uses none of these approaches. It uses randomness more like a video game than a traditional tabletop game. There are two reasons for this result. First, as explained in the introduction, one of Sneak's chief design goals was to create a game that retained the social dynamics of a tabletop game but used its digital technology to include as many mechanics as possible that were wholly novel to tabletop games. Given that the technique for user-generated randomness I found most viable reproduces existing board game mechanics quite closely, I chose to pursue video game-style use of randomness in order to maximize the possibility of finding novel mechanics.

The second factor was that stealth and deception turned out to be core to Sneak's gameplay. As the game developed, this emphasis arose naturally from the intersection of the game's mechanics and theme as well as the overall project goal of finding novel social interactions and play patterns enabled by this new play environment.

Designing for stealth and deception naturally lead to a format where players receive information and make decisions in secret. Perusing the catalog of tabletop games listed above that make players feel responsible for random number generation (Craps, D&D, Monopoly) it quickly becomes clear that the public performance of these dice rolls and card draws is key to these games' effects. It is the hope of the gamblers around the Craps table or the other members of the D&D party that forces on players a sense of ownership for the outcome of their dice roll. This kind of peer pressure is incompatible with the asymmetries of information necessary for stealth- and deception-driven games like Sneak. (Further, it may be incompatible with competitive games more generally. It is striking that the most prominent examples of this kind of "dice psychology" arise in collaborative games.)

Hence, Sneak is a tabletop game that uses random number generation in ways more analogous to a video game: to produce procedurally-generated content, to guide AI behavior, and to balance chances of victory between the players. Crucially, however, player perception of randomness in Sneak does not simply collapse into being identical to what would be found in a

[1] Once you add tap-for-random number to a game mechanics, it is never long until you start reaching for a design and animation vocabulary to sweeten that experience. Even a cursory glance at the large number of gambling games on both Apple and Google's mobile app stores shows that the near ubiquitous solution to this problem is to show virtual dice or shuffling cards.

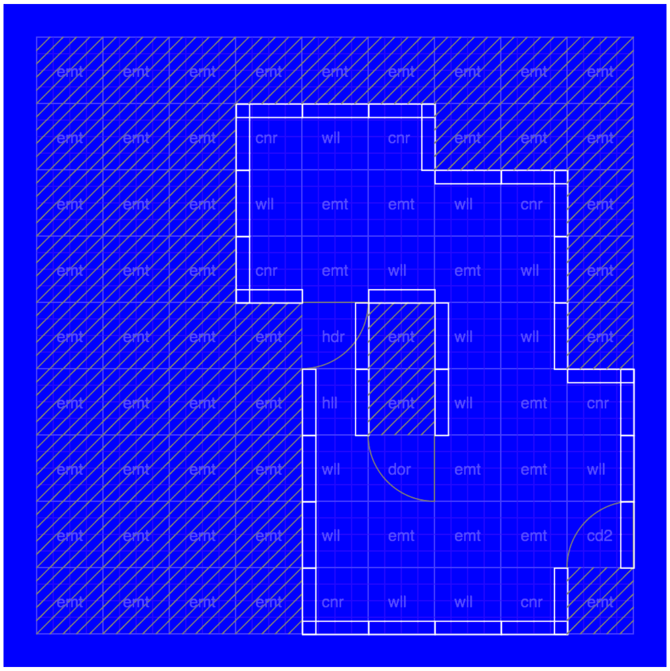
[2] Heroes and Villains, Greg Borenstein and Rael Dornfest. Unpublished prototype. 2014 http://gregborenstein.com/games/heroes_and_villains/

video game. Because of the game's hybrid format players perceive the components of the game which employ randomness in a wholly new way.

In this chapter, I divide consideration of Sneak's use of randomness into two sections. First, I examine how Sneak uses randomness to shape the environment in which players compete in the game. This section includes a discussion of Sneak's procedural map generation, item placement algorithms, and NPC pathfinding systems with an eye towards how they affect player cognitive load, strategy formation, re-playability, and game balancing. Secondly, I will examine how random elements interact with the game's theme to produce player stories that explain and make vivid randomly produced outcomes.

4.1 Procedural Playfields

There were three goals for Sneak's map generation algorithm: to provide variety between games, to make it easier for players to emulate NPC movement, and to add perceived narrative to character movements. In this section, I'll describe the algorithms that implement procedural map generation and NPC pathfinding and then I'll discuss their effects on players and how they interact with the tabletop context to produce these effects.[1]



emt	emt	emt	emt	emt	emt	emt	emt	emt
emt	emt	emt	cnr	wll	cnr	emt	emt	emt
emt	emt	emt	wll	emt	emt	wll	cnr	emt
emt	emt	emt	cnr	emt	wll	emt	wll	emt
emt	emt	emt	emt	hdr	emt	wll	wll	emt
emt	emt	emt	emt	hll	emt	wll	emt	cnr
emt	emt	emt	emt	wll	dor	emt	emt	wll
emt	emt	emt	emt	wll	emt	emt	emt	cd2
emt	emt	emt	emt	cnr	wll	wll	cnr	emt

FIG 4-1 A TYPICAL SNEAK MAP. OUTDOOR SQUARES ARE INDICATED WITH DIAGONAL LINES, DOORS WITH CURVES, AND WALLS WITH WHITE RECTANGLES.

It is worth noting that Sneak's approach in this area builds on recent developments in analog board games. the New German Board Game movement has produced a number of tabletop games with procedural elements. For example, in The Settlers of

[1] I discuss these two techniques together because the former is not possible without the latter. In the absence of pathfinding algorithms that can navigate around walls, NPCs would not be able to function on a board that had them.

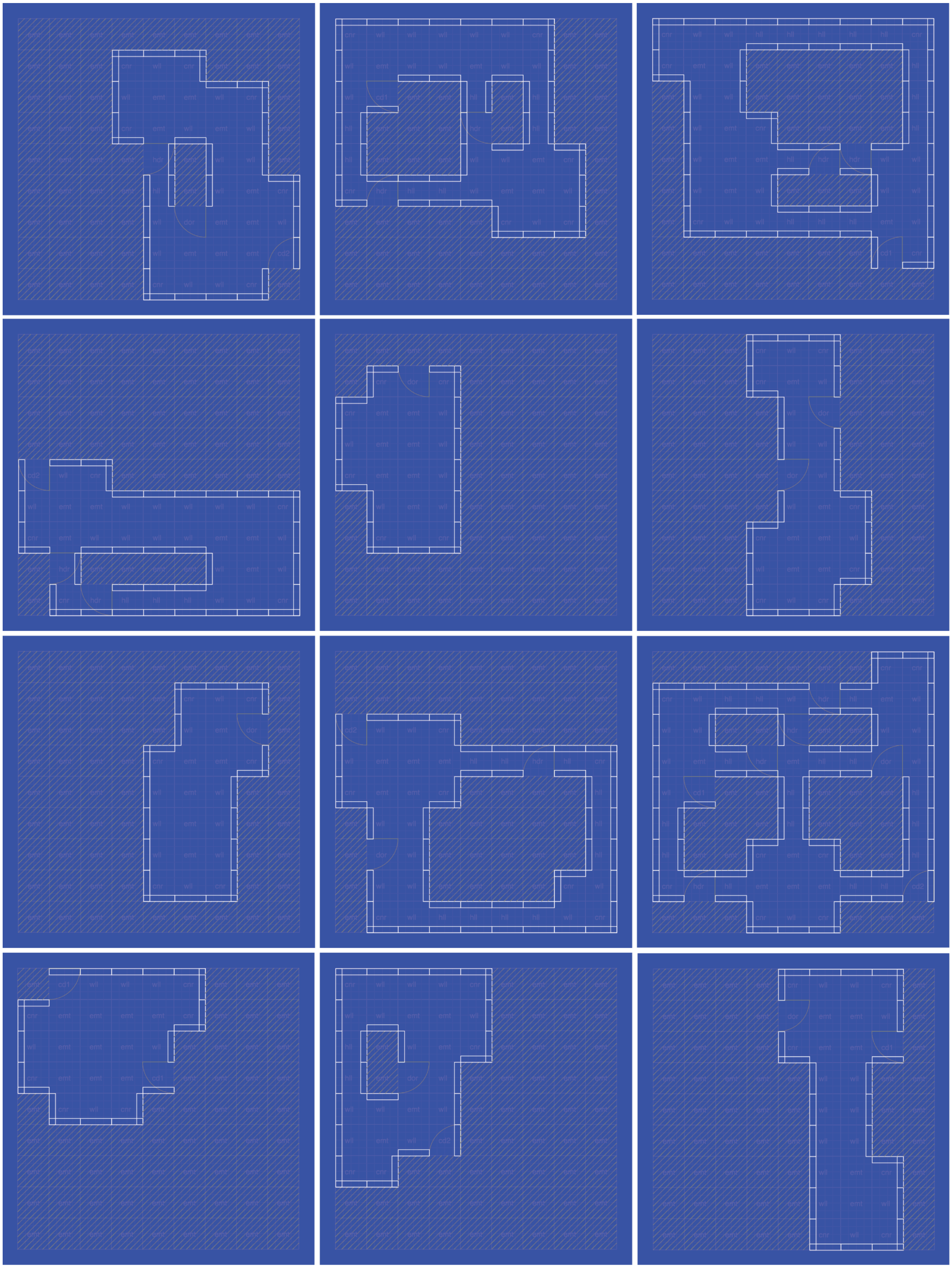


FIG 4-2 EXAMPLE OUTPUT FROM THE PROCEDURAL MAP GENERATION SYSTEM.

Catan[1], players construct the board at the start of each game by arranging a series of hexagonal tiles. The arrangement of tiles yields a different distribution of resources in the game, varying level of difficulty and type of game. Betrayal at House on the Hill[2] presents an even more interesting case. In this haunted house exploration game, every time players pass through a door to enter a new room, they draw a tile from a shuffled stack to represent the new room. This design seems inspired by the many video games such as The Legend of Zelda[3] in which players explore an unknown map which is gradually revealed as they visit each area. The mutual influence of video games and tabletop games is one of the key factors of the current game design environment in which hybrid digital tabletop games are emerging.

Sneak's map is a 9-by-9 grid that is divided into indoor and outdoor squares. Indoor and outdoor squares are separated by walls except for a few places in which they are connected by doors (Fig 4-1). Characters can move between any adjacent squares in all orthogonal and diagonal directions unless a wall separates them. All outdoor and indoor areas are connected by doors. Hence it is always possible to navigate from any square on the board to every other.

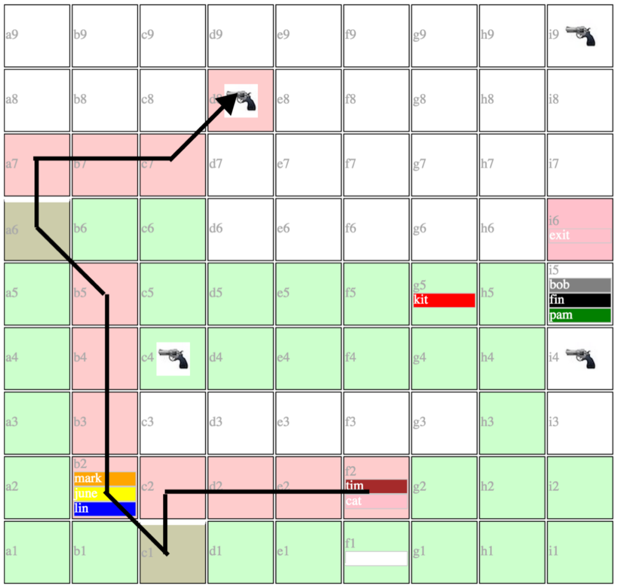

a9 b9 c9 d9 e9 f9 g9 h9 i9
a8 b8 c8 e8 f8 g8 h8 i8
a7 b7 c7 d7 e7 f7 g7 h7 i7
a6 b6 c6 d6 e6 f6 g6 h6 i6
exit
a5 b5 c5 d5 e5 f5 g5 h5 i5
kit
bob
fin
pam
a4 b4 c4 d4 e4 f4 g4 h4 i4
a3 b3 c3 d3 e3 f3 g3 h3 i3
a2 b2 c2 d2 e2 f2 g2 h2 i2
mark
june
lin
tim
cat
a1 b1 c1 d1 e1 f1 g1 h1 i1

FIG 4-3 AN EXAMPLE PATH FROM SQUARE F2 TO D8 CALCULATED BY THE PATHFINDING ALGORITHM.

Sneak's map generation system is based on the idea of "rooms". It implements a reductive simplification of real buildings as a series of large open areas ("rooms"), which are connected by narrow paths ("hallways"). This approach was chosen because of the variety of board layouts it produces. As you can see in Fig. 4-2, this approach produces board layouts that range from simple and compact (the board in the bottom left) to strongly vertically (bottom right) or horizontally oriented (second from top at left) to complex and sprawling (the two rightmost boards in the top and third rows). Further, its

[1] The Settles of Catan, Klaus Teuber. 999 Games. 1995.

[2] Betrayal at House on the Hill, Rob Daviau, Bruce Glassco, Bill McQuillan, Mike Selinker, and Teeuwynn Woodruff. Avalon Hill. 2004.

[3] The Legend of Zelda, Shigeru Miyamoto, Takashi Tezuka, Eiji Aonuma. Nintendo. 1986.

results are legible as plausible building layouts.

This system is implemented in two passes.[1] The first pass separates the board into indoor and outdoor areas (see Fig. 4-4 for a formal presentation of this algorithm). It starts by randomly selecting three squares to act as the centers of rooms. These squares, and the eight immediately surrounding, them are marked as being indoors. Importantly, these rooms are allowed to overlap and seeds are allowed to be placed at board edges, increasing the variation of both room shape and overall number of indoor squares generated on each map. Finally all of these rooms are connected by hallways, which are single width paths of indoor squares that turn at right angles. Any square not included in a room or hallway is considered to be outdoors.

```
Mark all squares on the board as outdoors.
Select three random squares on the board.
For each selected square (S1):
   Mark S1 as indoors.
   Mark S1’s neighbors as indoors.
   For each of the other two selected squares (Sn):
      Starting at S1
      Move horizontally to Sn’s column marking each traversed square indoors.
      Continue vertically to Sn’s row marking each traversed square indoors.
```

FIG 4-4 INDOOR AREA GENERATION ALGORITHM. DIVIDES THE MAP INTO INDOOR AND OUTDOOR AREAS.

After the room creation pass, the board has been split into at least two (and possibly several) distinct areas. There is a single contiguous indoor area, but the presence of hallways may have split the outdoor area into multiple isolated regions. In order to ensure that characters can navigate to every square on the board we perform a second pass to add doors to the map connecting all distinct indoor and outdoor areas.

The door assignment algorithm looks at each distinct outdoor area and adds a door to connect it to an adjacent indoor region see Fig. 4-5 for a formal presentation of this algorithm). It iterates over all of the isolated outdoor areas. In each outdoor area it finds the squares that border on indoor squares. It picks one of these edge squares at random and creates a door connecting it to an orthogonally adjacent indoor square.

[1] Both of the algorithms described here (as well as the pathfinding implementation discussed below) are based on a representation of the map as a graph of adjacent squares implemented in the Map object (see Fig. 2-11).

```
Make a list of all outdoor squares.
While there are squares in the list:
   Pick a random seed square from the list.
   Get the set of squares connected to the seed square.
   Select the squares in this set that have at least one indoor orthogonal
      neighbor.
   Pick a random square from this subset (call it S1).
   Pick a random orthogonally-adjacent indoor neighbor of S1 (call it S2).
   Add the pair (S1, S2) to a list of pairs of squares.
   Remove the seed square and the set of connected squares from the list of
      outdoor squares.
For each entry in the list of pairs:
   Mark the indoor square as having a door to the outdoor square.
```

FIG 4-5 DOOR ASSIGNMENT ALGORITHM. ENSURES COMPLETE MAP CONNECTIVITY.

The indoor/outdoor/door data produced by these algorithms is used to instruct the players in board setup (by displaying the blueprints seen in Fig. 4-1 and Fig. 4-2), to determine legal moves for players (any move that would navigate a player between indoor and outdoor squares without passing through a door is removed from the move input options), and as constraints in the NPC pathfinding algorithm.

NPC movement in Sneak is based on a series of destination squares. I will explain how these destinations are selected in the next section of this chapter. For the sake of this discussion, suffice it to say that during most turns NPCs select their move by calculating the most efficient path from their current square towards the square they have selected as their destination (see Fig. 4-3 for an example path). Paths cannot pass between indoor and outdoor squares except by passing through two squares connected by a door.

The pathfinding algorithm is based on a breadth-first search over the graph of connected squares (where two squares are considered connected if they are both indoors, both outdoors, or a door exists between them). Breadth-first search has been used widely in computer science since it was first invented in the the late 1950s.[1] Today, most video games use the more efficient

[1] Breadth-first search actually seems to have been invented twice independently. First in E. F. Moore, “The shortest path through a maze”. In Proceedings of the International Symposium on the Theory of Switching, Harvard University Press. 1959. And then again in C. Y. Lee, “An algorithm for path connection and its applications”. IRE Transactions on Electronic Computers. 1961

A* algorithm for pathfinding due to the performance constraints of real time games.[1] However, Sneak's turn-based format and the particularly small dimensions of its board make the simpler breadth-first search approach more than adequate.

The topology that results from Sneak's map generation and the AI behavior that results from its pathfinding approach interact with player perceptions to shape gameplay in a number of ways.

The first and most important impact of procedural map generation is that it increases the legibility of the board for the player. Sneak's map generation algorithm breaks up the undifferentiated board into a series of regions of different appearance, affordances, and strategic importance. Before adding procedurally generated maps to Sneak's design, the game used a completely open 8-by-8 grid. During playtesting players frequently complained of difficulties remembering where characters had been in previous turns and had no natural assumptions for where any character might be going against which to judge future moves for signs of human selection.

Procedural generation of maps addresses both of these problems. The shape and positions of rooms and the placement of doors act as landmarks against which to measure character movement. During playtests using procedurally-generated boards, players frequently referred to characters by their relationship to map landmarks in discussions amongst themselves, speaking of "the character near that door" or referring "how long red's been in that room".

Further, by differentiating the board's layout with walls instead of simply letter and number coordinates, procedurally-generated maps employ the player's spatial memory instead of their verbal memory for keeping track of characters' locations. Cognitive psychology has shown that verbal activity is less disruptive of spatial recall than of verbal recall.[2] Playing Sneak requires extensive verbal activity both in the form of reading on-screen information and speaking with other players. Hence, the addition of spatial cues to the board lightens the cumulative cognitive burden the game places on players. On Sneak's procedural maps, this enhancement of recall takes the form of small narratives that players tell themselves about the patterns of NPC movement they observe. During playtesting in post-game interviews, players will frequently explain their mental models by making reference to how characters moved relative to the

[1] P.E. Hart, N.J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths". IEEE Transactions on Systems Science and Cybernetics. 1968

[2] Lee R. Brooks, "Spatial and verbal components of the act of recall", Canadian Journal of Psychology. Vol 22(5), 1968.

board’s features: “I suspected black because they stayed near this central door”, “I was pretty sure it wasn’t red because they went the wrong way down this hall”, etc.[1]

Procedural generation also creates a hierarchy of strategic importance amongst different areas of the board. Doors and hallways become important choke points through which many characters will have to move to reach destinations around the map. These areas also constrain character movement, creating areas where characters options for changing direction are limited making their movements more predictable. Walls also break up the inter-square connectivity of the board creating significantly longer travel times between some areas than others. These attributes of procedurally generated maps act as affordances against which players can construct strategy. Choke points offer areas where players can linger to intersect many characters and gain information rapidly. Hallways create expectations about character movement patterns, drawing attention to characters that violate them by reversing direction. Long travel times between board regions alter the risk-reward tradeoffs of visiting certain destinations.

Importantly, none of these properties are inherent to Sneak’s maps being produced procedurally. A human-designed map could take advantage of all of these properties and, potentially, through careful design work, even achieve a more perfectly composed arena for player competition. However, the use of human level design would limit the number of possible maps available in Sneak. Including only a small number of maps would limit re-playability. For all the reasons described here, map specifics have a major impact on player strategies. If maps recurred, repeat players would learn to exploit their eccentricities. This would induce a very different kind of learning curve, one similar to competitive multiplayer first person shooters such as Counter-Strike: Global Offensive[2] which reward detailed knowledge of map geometry and how it shapes with strategy and tactics.

In Sneak, I want player expertise to focus on social interaction rather than level memorization. I want to present players with a skill curve based in figuring out how your opponent is reacting to the current board situation, observing their responses to game events, and coming up with techniques to exploit an emergent situation to elicit a revealing response from them. These goals argue for procedural map generation as the source of variation rather than level design. Procedural generation can produce an infinite number of maps and by

[1] See Chapter 5, “The Cost of Simulation”, for more detail on how players remember and model character movements.

[2] Counter-Strike: Global Offensive, designer unknown. Valve Corporation. 2012.

refusing to repeat maps it requires players to focus on the game's systems rather than the specifics of any particular map.

This strategy has a strong precedent in video game design in the "roguelike". Roguelikes are a genre of video games named after the 1980 RPG, Rogue.[1] Roguelikes include turn-based movement, permanent death, randomization, and map exploration. This genre has experienced a swift rise in popularity in recent years. Game developer Kevin Forbes (Don't Starve) attributes their popularity to how they offer players the opportunity to "master complex systems", "direct their own experience", and experience "emergent gameplay."[2]

Sneak includes all of the properties of a roguelike. Death is permanent, maps are randomly generated, and play is turn-based. Sneak is a hybrid digital-physical roguelike. Like other roguelikes, balance is achieved differently in Sneak than in non-roguelike genres. In games that make heavy use of procedural elements there are two different questions to ask when it comes to balance: How even are the odds between the players in any individual game? Is the range of outcomes produced by the procedural system balanced between different strategies or player types?

In each game of Sneak, after the map is procedurally generated, the characters, guns, and exit are all placed. The algorithms that select the locations for these game elements use randomness but have been designed to produce outputs that contribute to fair games between players. Take first, the algorithm for placing the game's four guns. This algorithm divides the board up into four quadrants and selects one random square in each of these to receive a gun. This approach ensures that a gun is always within some relatively uniform maximum distance from every square on the board, but it also allows for significant variation between games. For example, games in which all four guns spawn near the center of the board present players with different strategic options than those in which the guns are more widely spread. In such a game, given a particular arrangement of characters, a player might find themselves in a situation where they can shoot a gun and still have the opportunity to reach a second gun before any other character effectively giving them two chances to guess their opponent before being shot themselves.

Initial character placement is another example of how Sneak uses shaped randomness to ensure that variation and fairness coexist. Sneak's algorithm for selecting character starting

[1] Rogue, Michael Toy, Glenn Wichman, Ken Arnold, Jon Lane. Various publishers. 1980.

[2] Quoted in Christian Nutt, 'Roguelikes': Getting to the heart of the it-genre, Gamasutra. 2014. http://www.gamasutra.com/view/feature/218178/roguelikes_getting_to_the_heart_.php

positions starts by picking a small number of seed squares (six in the current implementation). It then randomly places each character on one of these seed squares with the exception of the character holding the plans. After all other characters have been placed, the character holding the plans is placed randomly onto one of the seed squares that includes neither of the players. This approach has been tuned so that most seed squares will begin with two or three characters on them with the occasional square having four or one. While this leads to player starting situations that are unequally advantageous, the most imbalanced possibilities (of one player having the opportunity to pick up the plans on the first turn) has been avoided.

These various uses of randomness, and particularly the procedural map generation system, can lead to games in which one player is likely to gain significantly more information than other. Players spawn in locations near a greater number of NPCs or, on the contrary, at the end of a long winding hallway far from the heart of the action. While Sneak's design does its best to minimize these situations, they can arise. In these cases, Sneak is "balanced" at a meta level by the existence of its multiple victory conditions. A player's chance to win is not solely linked to their opportunities to gain information about the plans. Players can also win by detecting their opponent's identity and the cues necessary to do so are not tied to their board position or the results of the game's other random systems. In fact, in some ways, players in a good position relative to those systems end up receiving more and more interesting information through the app, creating more chances for them to express surprise or otherwise emit social cues that the other player can detect. In this way, the being able to win by shooting your opponent acts as a "catchup mechanic" to keep players who have fallen into an unfortunate corner of the range of outcomes possible in Sneak's procedural generation system.

4.2 Player Stories from Random Outcomes

The uses of randomness covered so far present themselves to the player proudly as random systems. At the start of the game players are simply presented with a map, character, and item positions. These results are not contextualized or explained in any way. They simply make up the starting situation in which they players find themselves this game. Sneak also uses randomness in a couple of subtler ways behind the scenes: to select destinations for the NPCs and to assign NPC identity. That randomness governs these elements of the game is not obvious to the players. In absence of that knowledge players have a tendency to attribute meaning to random outcomes in a way that enhances their experience of the game.

The two mechanics that use randomness invisibly are NPC destination selection and NPC identity assignment. In the previous section, I explained how NPC pathfinding works, but I did not explain how NPCs select the destination square to which they use pathfinding to navigate. NPCs select destinations at random from all of the squares on the board. When they reach that destination they stay there until they select a new destination. On each turn they generate a random number to see if it is time to select a new destination. Each character has a different threshold determining how likely they are to select a new destination. These thresholds are also selected randomly for each NPC at the start of the game. The combination of these two uses of randomness produces surprisingly complex variations in behavior between NPCs. NPCs with a high threshold for new destination selection seem to move in a single continuous snakelike path, stopping here and there but sometimes also seeming to reverse or change direction out of nowhere. Those with a low threshold seem to largely lay low, stopping for long periods during the game and then suddenly breaking out and going on a long path from one side of the board to the other.

This variation was not present in early prototypes, but since I introduced it I noticed that it has a number of interesting effects on players' perception of the NPCs. Even though this was a simple change, it made it significantly more difficult for players to reverse engineer the algorithm driving character movement (see The Cost of Simulation below for a complete discussion of how players perform this task). When asked about it in post-game interviews, players stories about and descriptions of NPC movement became more significantly more diverse. Some players began to attribute complex logic to NPC movement, mentioning everything from the density of other characters (one player described a particular NPC as "agoraphobic"[1]) to the position of guns.

These complex stories give players cover to make their desired moves with their own characters. In earlier prototypes, when NPC destination selection was more consistent, players often complained of knowing where they wanted to go but feeling like they could not make the necessary moves to get there without making their identity obvious. With this new destination selection algorithm, players can now interpret NPC behavior broadly enough to make them feel justified in moving more boldly. Simultaneously, player's ability to intuit each other's identity has not declined so this new freedom has not actually helped them hide their behavior from their each other.

[1] I think they meant "enochlophobic" (fear of crowds) rather than "agoraphobic" (fear or open spaces). The NPC in question had stirred itself from a long series of holds to move away from a gather crowd of other characters.

The other system where randomness leads to player stories is less mechanically meaningful and more simply evocative. As explained in Chapter 2, Sneak uses repurposed O-scale model train figures as pieces to represent its characters. The characters were selected (and modified) to match the 10 colors of the different characters in Sneak. But because they come from model train supplies, the characters are dressed in garb indicating professions common to that environment: construction workers, businessmen, policemen, etc. These different character costumes have no mechanical role in sneak whatsoever. They simply help players see the different color characters. The relationship between the profession depicted by the figures and the color of the character in the game is arbitrary. Regardless, these professions shape player perception of the characters' movements. They react differently to a construction worker who stops in a doorway than they do to a cop. Players have referred to these identities in their post-game conversations about how they understood NPC behavior.

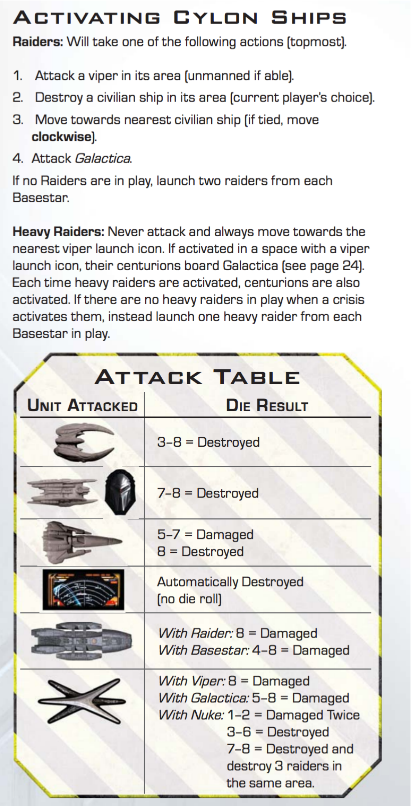
It is possible that this involuntary process of story formation is not helpful given Sneak's current design. Any assumptions it builds in the players' minds about the functioning of the system are spurious at best. However, this relationship between the game's theme and how players interpret its behavior, even when the link is completely random, is a powerful illustration of the apophenia which players bring to random systems in games and the ability of even their mistaken interpretations of random behavior to generate meaningful stories.

5. The Cost of Simulation

Simulating complex systems is the bread and butter of video games. Perhaps the canonical example is SimCity[1], a game that positions the user as a manager of the transportation, construction, and regulatory systems of a simulated city. The simulation of physics and human behavior is a ubiquitous part of most games, but particularly central to the first- and third-person shooter genres. For example, Far Cry 2[2] is well-known for its simulation of unusual properties such as guns jamming from dirt, characters contracting malaria, and trees catching on fire due to gunfire.

Both of these (otherwise quite different) applications of simulation use it to produce rich and emergent gameplay with a high degree of variability, increasing game re-playability. The complex calculations required to simulate such varied systems have become easily within the reach of modern computers and game consoles. An extreme example of the use of this computational power for simulation is Dwarf Fortress[3], a construction and management simulation game with only ASCII graphics that simulates thousands of years of world geology and history to create its game world before play even begins.

Tabletop games, on the other hand, are restricted in the complexity of system they can simulate since they depend on players to do all of their calculations. While players with dice and counters are Turing complete, the time, effort, and tedium necessary to conduct even a mildly complex simulation would bore and discourage most players.



ACTIVATING CYLON SHIPS

Raiders: Will take one of the following actions (topmost).

1. Attack a viper in its area (unmanned if able).
2. Destroy a civilian ship in its area (current player's choice).
3. Move towards nearest civilian ship (if tied, move **clockwise**).
4. Attack *Galactica*.

If no Raiders are in play, launch two raiders from each Basestar.

Heavy Raiders: Never attack and always move towards the nearest viper launch icon. If activated in a space with a viper launch icon, their centurions board Galactica (see page 24). Each time heavy raiders are activated, centurions are also activated. If there are no heavy raiders in play when a crisis activates them, instead launch one heavy raider from each Basestar in play.

ATTACK TABLE

UNIT ATTACKED	DIE RESULT
	3–8 = Destroyed
	7–8 = Destroyed
	5–7 = Damaged 8 = Destroyed
	Automatically Destroyed (no die roll)
	With Raider: 8 = Damaged *With Basestar:* 4–8 = Damaged
	With Viper: 8 = Damaged *With Galactica:* 5–8 = Damaged *With Nuke:* 1–2 = Damaged Twice 3–6 = Destroyed 7–8 = Destroyed and destroy 3 raiders in the same area.

FIG 5-1 BATTLESTAR GALLACTICA: PEGASUS EXPANSION NPC LOGIC

An example of a mildly complex simulation in a tabletop game can be found in the Pegasus expansion to the

[1] SimCity, Will Wright. Maxis. 1989.

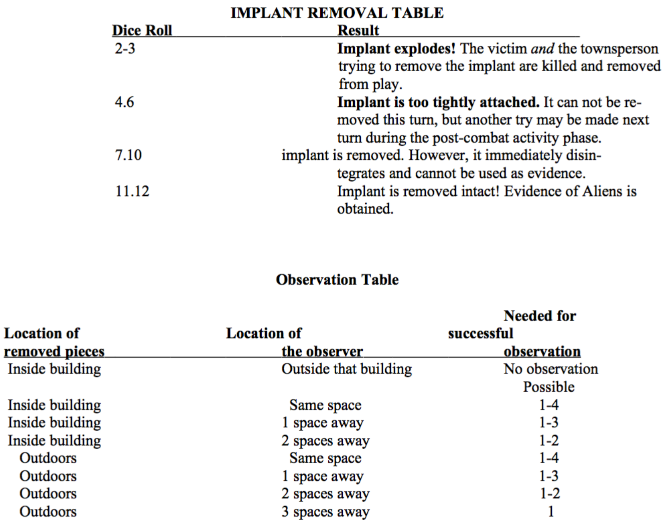
[2] Far Cry 2, Clint Hocking. Ubisoft. 2008.

[3] Dwarf Fortress, Tarn Adams and Zach Adams. Bay 12 Games. 2002.

Battlestar Gallactica board game[1] in which players must simulate the navigation and attacks of a series of Cylon ships threatening their battlestar. While by no means a comprehensive simulation the implementation process requires the players to roll dice repeatedly and look up sets of decision rules in a rulebook (see Fig. 5-1).

Implementing even mildly complex simulations in a tabletop game is awkward and painstaking for players. While this difficulty significantly limits the complexity of system that can be simulated, the tabletop context does have some design advantages for a simulation. Since players have to move all of the pieces themselves and implement the system's rules, they do not just interact with its surface phenomena as in a digital simulation, but instead inhabit its operation. Video game players see the results of a simulation and come to understand its inner workings experimentally by observing how it reacts to their inputs. Tabletop players, on the other hand, are responsible for enacting their games' simulations. They interact not with the simulation's output but with its internal logic.

For example, in They've Invaded Pleasantville[2] one player controls a group of aliens that secretly seize control of a series of human victims by installing implants in them. The other player controls the remaining still-human townspeople. The human player attempts to discover which townspeople have been implanted in order to liberate or interrogate them. Fig. 5-2 shows two of the many lookup tables that govern event outcomes in the game: the "Implant Removal Table" and the "Observation Table". The Implant Removal Table demonstrates how the game simulates the result of an uncertain action by combining a random dice roll with the description of the outcome. Similarly, the Observation Table governs whether or not nearby humans detect when one of their number is seized and implanted by aliens.



IMPLANT REMOVAL TABLE

Dice Roll	Result
2-3	**Implant explodes!** The victim *and* the townsperson trying to remove the implant are killed and removed from play.
4.6	**Implant is too tightly attached.** It can not be removed this turn, but another try may be made next turn during the post-combat activity phase.
7.10	implant is removed. However, it immediately disintegrates and cannot be used as evidence.
11.12	Implant is removed intact! Evidence of Aliens is obtained.

Observation Table

Location of removed pieces	**Location of the observer**	**Needed for successful observation**
Inside building	Outside that building	No observation Possible
Inside building	Same space	1-4
Inside building	1 space away	1-3
Inside building	2 spaces away	1-2
Outdoors	Same space	1-4
Outdoors	1 space away	1-3
Outdoors	2 spaces away	1-2
Outdoors	3 spaces away	1

FIG 5-2 LOOKUP TABLES FROM THEY'VE INVADED PLEASANTVILLE'S INSTRUCTION BOOKLET

[1] Battlestar Galactica: Pegasus Expansion, Daniel Clark, Corey Konieczka and Tim Uren. Fantasy Flight. 2009.

[2] They've Invaded Pleasantville, Michael Price. TSR. 1981.

The process of using these lookup tables (and the many others littered throughout They've Invaded Pleasantville's manual) is painstaking and slow. However, their presence gives attentive players an opportunity to fully understand the rules of the simulations driving the game as they are literally laid out for players to read. This is a representative, if extreme, example for how simulations work in many tabletop games. While players pay a high cost for executing the simulation in working memory and slowed game pace, their presence in the game can serve to give players deep knowledge of the simulated system's structure.

The use of simulation is one of the areas of greatest promise in hybrid digital-physical games. Including software in tabletop games has the potential to raise the ceiling on the complexity of simulation possible in these games while still allowing players to retain their intimate relationship with their workings.

An illuminating early example of this potential is Alchemists by Matúš Kotry.[1] In Alchemists players perform experiments by combining ingredients (represented by cards) in an attempt to discover and publish alchemical laws. The results of various ingredient combinations are determined by a smartphone app. The app recalculates new alchemical laws for each game so that players must experiment to rediscover them anew on each playthrough. It also hides the particular "alchemical logic" of any specific game from the players, enabling the game's core mechanic of discovery. In fact, Alchemists includes an optional smartphone-free mode where a person can take the role of the app, going through a painstaking procedure to generate a results chart and then looking up each proposed concoction on that chart to determine its result. That person does not play in the game, but solely acts in this manual computation role. The complexity (and total lack of fun) required to play the game without a computer is a beautiful illustration of the potential of digital simulation in tabletop games.

This example points towards a novel play pattern which also emerged in the design and playtesting of Sneak and may turn out to be a useful guide for future hybrid digital-physical tabletop games: games that are explicitly about the discovery of the rules behind a simulated system. In both Alchemists and Sneak players interact with a simulated system whose logic is computed by a digital device and, at first, unknown to them. But then, instead of simply seeing the system's outcomes, the tabletop components of each game intimately involve the players in producing those outcomes. Hence, they gradually learn the explicit logic that makes up the simulation. And, further, the game's victory conditions require the player to actively reproduce the behavior simulated by the system in order to compete.

[1] Alchemists, Matúš Kotry. Arclight. 2014.

In Sneak the simulated system that plays this role is NPC movement. Sneak asks players to figure out which of the ten characters on the board is controlled by the other player. The chief evidence it gives them is the characters' movement patterns. Hence, players carefully scrutinize character movement to understand the logic that governs NPC movement. They then use this logic to catch out opposing players who diverge from it. They also try to mimic it with their own character's movements to avoid being caught out themselves. By intermingling human and simulation behavior, Sneak specifically asks players to build a model of the system driving the simulation.

Through Sneak's playtesting I have learned a number of useful lessons about how players go about this process of modeling the simulated system and how a game that asks players to go about this can fail. For the rest of this chapter, I will describe these lessons using notes from playtesting sessions and a description of the design iterations the game underwent. I will explain how players learn to model NPC behavior, the typical mistakes they make in the process, how their learning is constrained by their their working memory and in-built assumptions, and, finally, how this process of simulation comprehension affects Sneak's re-playability. Then, I will examine one of the biggest challenges entailed by this simulation modeling mechanic: the need to keep the digital simulation and the physical board and pieces synchronized. I will describe the synchronization challenges that arose during Sneak's development and explain the design tools I have used to overcome or minimize them.

5.1 How Players Model NPC Behavior

In "Thinking Fast and Slow", Nobel prize-winning psychologist Daniel Kahneman describes a set of systematic errors people make when forming judgments on too little information. "Subjective confidence in a judgment is not a reasoned evaluation of the probability that this judgment is correct. Confidence is a feeling, which reflects the coherence of the information and the cognitive ease of processing it."[1] Kahneman calls this "coherence bias". It derives from the tendency of our reflexive judgments to "automatically and effortlessly identify causal connections between events, sometimes even when the connection is spurious."[2] In other words we instinctively build stories out of whatever information is available heedless of whether

[1] Daniel Kahneman, Thinking Fast and Slow. Farrar, Straus and Giroux. 2011. p. 212

[2] ibid. p.110

that data is sufficient to support such a story and we judge both our own conclusions as well as new incoming facts based on the coherence of the story we tell ourselves. “The measure of success for [our intuitive faculties] is the coherence of the story it manages to create. The amount and quality of the data on which the story is based are largely irrelevant. When information is scare, which is a common occurrence, [our intuition] operates as a machine for jumping to conclusions.”[1]

In playtesting Sneak, I observed this coherence bias repeatedly in how players understood the behavior of the game’s NPCs. From their earliest interactions with the game, players began to form a model to explain the characters’ movements and, crucially, to attempt to distinguish NPC movement from behavior indicating the control of their opponents. This model typically consists of a story players told themselves about what moves an NPC would be likely to make and what kinds of moves humans would prefer.

For example, there was one movement pattern that reliably produced the assumption that a character was an NPC across all playtests. Characters that move away from the center of the board, or other areas in which large groups of characters are congregating, are assumed by players to be NPCs. When asked in debriefing interviews how they narrowed down the candidates for which character their opponent controlled, players repeatedly cited this behavior as indicative of being an NPC. They explained this conclusion by talking about why they would not make such a move with their own character: moving to the edge of the board creates fewer opportunities for dialogs with other characters and hence denies them of information about the location of the plans. It also eliminates possibilities for future good moves, requiring players to spend multiple turns to reach other characters.

This example points towards a more general trend in the assumptions and biases that shape players’ judgment when attempting to distinguish NPCs from players. Lacking information on the algorithms that control NPC movements, players substitute the heuristic of “not something I would do.” Similarly, lacking information about the other player’s strategy, they substitute with the opposite heuristic: “seems like something I would do”. As in the example given above about characters that moved to the outside of the board, if players could come up with a story of why they would make an observed move for a character then they would suspect that character of being human-controlled. If they couldn’t come up with such a story, they would write that character off as an NPC.

[1] ibid p.85. Kahneman refers to our snap decision-making, intuitive faculties as “System 1” in his two system model of the psychology of judgment. I have substitutive out that phrase here to enhance clarity for readers who are not familiar with Kahneman’s system.

Players' confidence in conclusions like these dramatically exceeded the evidence they were actually able to perceive and remember. Specifically, players reported being tightly constrained by the limits of their working memory in keeping track of the sequence of moves made by each character in order to examine them for patterns that might reveal NPC- or player-like behavior. Players reported that they could remember moves made by individual characters for no more than two turns and often struggled to remember moves that long when many character moved in the same turn. Early on in playtesting, struggling with this limitation myself, I introduced an optional mechanic in which players can orient the pieces representing each character to face in the direction that character last moved to act as a memory aid. In playtesting sessions, I offer this to players as an option that they can employ if both players in a game agree. Players always choose to use this optional mechanic.

Interestingly, players' stories tended to see NPC movement and player-controlled movement as starkly contrasting despite the fact that Sneak provides players with very little intrinsic information about the identity of their opponents. Players watch nine character move only one of which is controlled by their opponent. They can barely remember the sequence of moves made by each character. But they still maintain a strong belief that they can distinguish NPC from player movement. This gap fits perfectly with the coherence bias described by Kahneman, which predicts that people are inclined to hold a greater degree of confidence in their conclusions that corresponds with the power of the story they have told themselves rather than the observations they have actually made.

Some of the most impressive strategic thinking in Sneak playtests came from players who consciously noted this gap and attempted to exploit it. For example in one playtest session,[1] after their first game two players insisted on playing again a second time. In their second game, after the initial rounds, the game entered a phase where most of the NPCs held their positions for extended periods with only two or three moving each round. The two players reacted to this situation quite differently. Player one decided to hold his position for multiple consecutive rounds as well. In a post-game interview he explained that he was afraid that moving would call attention to his character and seem like an obvious divergence from NPC behavior. Player two, on the other hand, took exactly the opposite tack. He moved his character repeatedly during this phase of the game, collecting a large amount of information and eventually discovering which character held the plans. When asked, player two explained that he thought that moving a lot seemed very unnatural and so would be something only an NPC would do. Hence, he decided

[1] See Appendix A for complete playtesting notes.

to do it in order to act like an NPC. Player one revealed that this gambit had completely worked on him and he had not suspected player two's character at all.

One of the fascinating things about this story is that it shows players have different criteria for their own actions versus those of other characters. Player one held still because moving felt vulnerable and likely to reveal him as a player. However, the movements of player two, which were exactly like those he was afraid to make, seemed NPC like to him and did not arouse his suspicion.

This interplay of player stories also points to how Sneak might be able to both achieve a meaningful skill curve and retain re-playability even after players have reverse engineered the algorithms that drive NPC movement.

One of the challenges facing a game like Sneak is what happens after people have played enough games to successfully reverse engineer the behaviors whose logic is hidden by the game's digital system. Since that reverse engineering is exactly the challenge the game sets out for players Sneak faces a danger of collapsing in interest after players have accomplished it.

In the play test session just described, the players played two games in a row. After each game, I asked the players to explain their understanding of the logic governing NPC movement. After the first game their explanations were significantly incomplete. While they had correctly observed some patterns, they had only been partially successful in reasoning out the full logic. However, by the end of the second game, their understanding had dramatically improved. When asked the same interview questions again their new explanation was nearly completely correct.

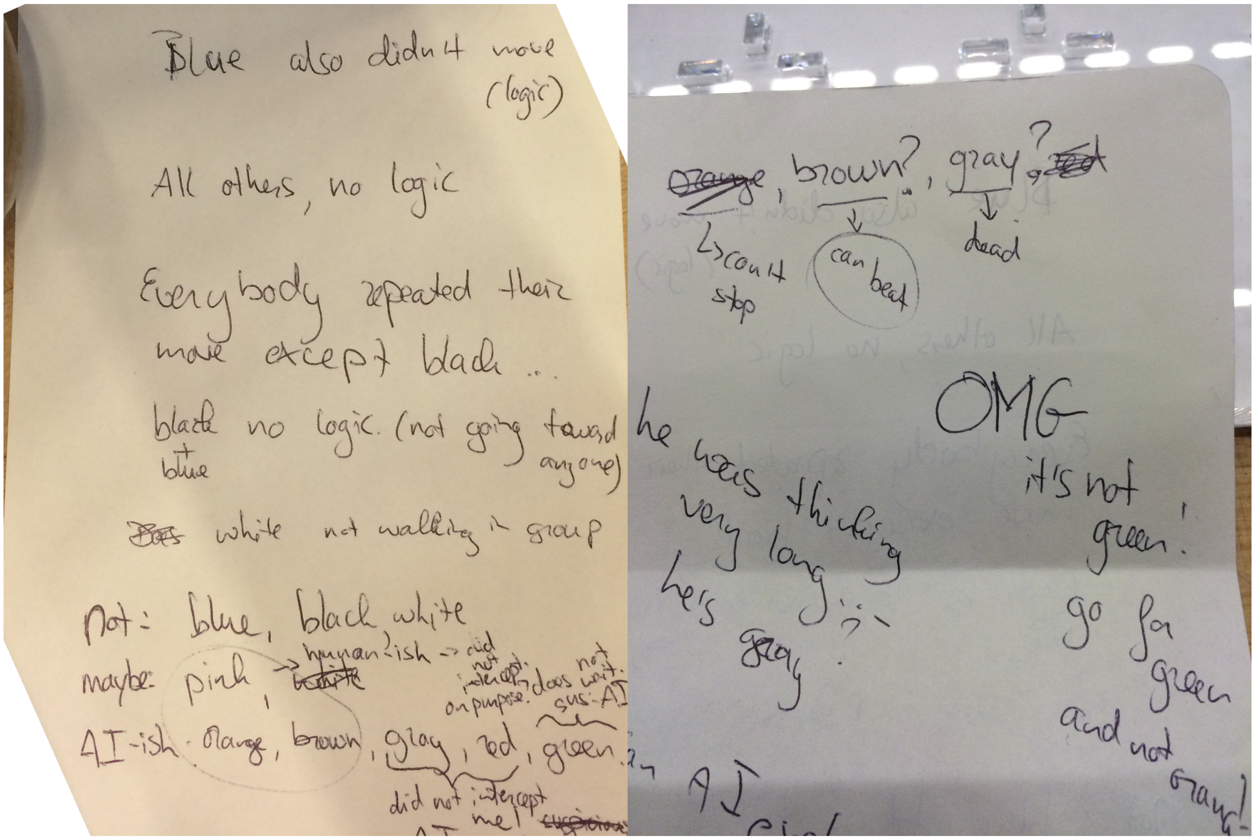
Since that playtest, I have introduced procedurally generated maps and more sophisticated NPC pathfinding to the game. These mechanics have somewhat slowed players in reverse engineering the logic of NPC movement, largely because there are simply fewer points at which NPC might veer off of paths that would be taken by human players.

The danger of repeat players eventually completely understanding the algorithm is still a design concern. However, one other recent playtest pointed towards a dynamic that might emerge as players gain more experience to allay that fear.

This playtest pitted a first-time player against a player who has played multiple times previously. When the game started the more experienced player was situated in a significantly disadvantageous position, only near one or two other NPCs. Instead of immediately moving towards the central cluster of characters, this player set out in the opposite direction, moving into a corner of map even further from the action and wasting a turn picking up a gun. In the post-game interview, this player explained that they had intentionally done this because it would not seem like something a player would do and would hence throw off the other player's suspicion. This strategy proved successful and their opponent ended up as the only playtester so far to not even suspect the correct identity of their opponent (more about this particular playtest shortly).

Surprisingly, despite the biases that effect players' reasoning about NPC behavior they are still consistently able to effectively narrow down the identity of their opponents. In post-game interviews, all but one of Sneak's playtesters included their opponent's real identity amongst their two or three top candidates when asked to list which character they suspected was controlled by the other player. This accuracy is particularly striking given that it often arose from incorrect observations and conclusions about how those candidates' movements diverged from the other characters.

I believe this seeming paradox is explained by the fact that players use extrinsic social cues picked up from their opponents to form these beliefs to a far greater extent than they are aware. They over-credit their observations and reasoning, just as Kahneman's coherence bias would suggest they do, while simultaneously undervaluing their ability to pick up on subtle social hints.



Blue also didn't move (logic)

All others, no logic

Everybody repeated their move except black ...

black + blue no logic. (not going toward anyone)

~~Bes~~ white not walking in group

Not: blue, black white

maybe: pink, → ~~white~~ human-ish → did not intercept on purpose. does not wait. sus-AI

AI-ish: orange, brown, gray, red, green.

did not intercept me! ~~suspicious~~

~~Orange~~, brown?, gray?, ~~red~~

↳ can't stop

can beat

dead

OMG

he was thinking very long ;-`

he's gray.

it's not green!

go for green and not gray!

FIG 5-3 NOTES KEPT BY A PLAYER ATTEMPTING TO DEDUCE THE IDENTITY OF THEIR OPPONENT.

This hypothesis is supported by one of the most striking games that occurred during playtesting. At the start of game, during the dialog phase of the second round, player two asked if it was acceptable for them to keep notes on paper. I said the game included no rules against it and so it was allowed if their opponent consented. The other player agreed and so player two began to take notes.[1] On each turn, when their side of the dialog phase arrived, player two spent a couple of minutes noting down observations about character movement and their own suspicions as to their opponents identity (see Fig 5-3 for some images of their notes).

This action had two effects that impacted the game. First, it significantly slowed the rate of play. Where all previous games of Sneak had ranged between 20 and 40 minutes, this one stretched out to more than 80. Second, taking and reviewing notes occupied nearly all of player two's attention, meaning she spent significantly less time looking at or talking to player one. In other words, player two paid for their analytical depth with a reduction in attention to social cues. How little player two was watching player one even because a joke between the two players during the course of the game with player one pretending to tap outlandishly on the tablet during their turn in a comical attempt to throw of player two who was not even watching.

As described above, at the start of the game it seemed that player two was in a dominant position. They participated in more dialogs and discovered which NPC had the plans relatively quickly. However, player two's analysis lead them to a false conclusion about player one's identity. Namely, that player one was green, the character with the plans. At one point, player two took me aside to ask if there was even any point in continuing the game since green had the plans and could make it to the exit before she could intervene or reach a gun. I indicated that there was.

When green went past the exit without ending the game, player two was completely shocked. See the page on the right in Fig 5-3 where they write "OMG it's not green!". After that point, player two scrambled to figure out what to do. They poisoned a gray, one of their other (also incorrect) candidates. But the game ended shortly thereafter when player one shot them. In the discussion after the game, player two listed their other candidates for player one's identity all of which were also wrong. They never even suspected the correct choice. Player one, on the other hand, had successfully narrowed down the options to two and the poisoning death of gray clinched it for them.

The nature of player two's notes is fascinating. When she asked for permission to take notes, player two's rationale centered on the need to keep track of how characters moved

[1] See Appendix A for my complete notes from this playtest.

across multiple turns, to supplement limited working memory. However, if you look at the actual notes she took, they do not chiefly consist of direct observations of character movements. There are some descriptive phrases — “blue also didn’t move”, “everybody repeated their move except black”, “white not walking in group” — but the notes mainly log the history of player two’s own suspicions, conclusions, and assumptions. One of the first notes that appears lists eliminated options: “not: blue, black, white” as well candidates thought to be “humanish” and “AI-ish”. Listed as reasons for being “AI-ish”, gray and red are noted as “did not intercept me”. This is again the common player assumption that seeking out conversations is a core human behavior in the game.

Rather than lending them greater observational clarity, player two’s notes acted as a powerful amplifier for coherence bias. The notes acted as a venue for player two to reinforce and provide illusory authority to to their own invented stories.

Sneak’s design requires players to simultaneously learn the rules of a simulation and intuit the intentions of another person. Intermingling these two types of intuitive judgments leads to Sneak's most interesting dynamics. This combination seems a promising area for future research in hybrid board game design.

5.3 The Challenges of Human/Software Synchronization

So far in this chapter, we have seen looked closely at how players think about Sneak’s simulation and how that simulation leads them to new interactions with other players. However, these new dynamics promised by hybrid digital-physical tabletop games are only accessible if those games are able to overcome the inherent challenge entailed by this format. The heart of that challenge is the difficulty of keeping the game's digital system and its physical pieces in sync.

The game’s software maintains an internal representation of the state of the game that includes everything from the game’s map to the position of all of the characters to the state of each NPC’s knowledge of the plans to which player has the current turn. Many of these elements have a parallel physical representation in the state of the board and game pieces. There are many opportunities for these two representations to diverge as play proceeds. Players may setup the board incorrectly, they may forget to move a character, they may move a character incorrectly, they may pass the device to a player outside of their turn, etc. All of these failures have the potential to ruin a playthrough of the game. Even worse, many of them are not immediately apparent, but only show up after a number of turns when the app issues a move instruction that is clearly out of sync with the board (like instructing the players to move a character through a wall). This makes players very frustrated with the game when it occurs both because the reason for the problem is invisible to them, there is nothing they can do to successfully continue the game, and a significant amount of their time has been wasted.

As I explained in Section 1.1, nearly all of the prior work on digital tabletop games consists of technical fixes for this kind of problem. Many researchers have devised systems that synchronize digital and physical game components through the use of various sensors and physical interfaces. Sneak takes a different approach to the problem, using the players themselves to achieve this synchronization. While this choice enables many of Sneak's key qualities (like the dynamics explored in the previous section) it also means that I had to address the synchronization problem constantly while designing it.

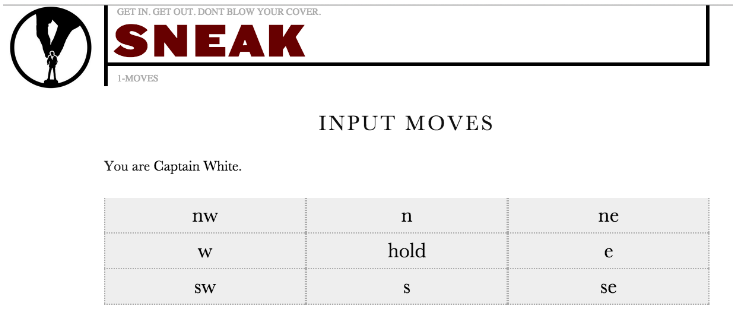
In this section, I will outline some of the specific synchronization challenges I encountered during Sneak's design process and explain how I addressed them. While its dangers have been significantly mitigated, synchronization in Sneak is not a completely solved problem. I will end the chapter by discussing one major area of synchronization in Sneak that is still an open problem: how to help players detect if something has gone wrong and correct it.

Synchronization in Sneak is fundamentally a user interface problem. From the initial generation of the map and the starting placement of the characters, information about the state of the game originates with the app and then is communicated to the players so they can move the physical pieces to match. Problems occur when users misunderstand this information. In order to avoid problems I have systematically sought to reduce the amount of information that needs to be communicated to players and to simplify and clarify communication that is required.

The central venue for synchronization in Sneak is the interface through which the game instructs players in how to place and move the characters. Early on in Sneak's design I hit upon the idea of using a coordinate system to refer to the squares on the board, similar to that found in chess. On each turn, the game would give players 10 instructions in the form: “move the red character on b3 to c4” or “the blue character on d2 holds”. The game always used absolute coordinates to refer to both the starting and ending positions of each character. In order to execute such an instruction, players had to refer to the coordinate system printed on the board twice, both to locate the character to move and to find the square to which they should move. This process proved slow, painstaking, and error prone for players.

I improved this interface by making two changes, one structural and one linguistic. First, I changed the color assignment of characters so that each character had a unique color. That way instead of referring to the moving character by their starting square instructions could refer to them by their color, which is much easier for players to find by simply glancing at the board. Secondly, I reformulated these instructions so that they used cardinal directions instead of absolute coordinates, i.e. “move the red character north” or “move the blue character southeast”. Thusly, as long as players keep their orientation relative to the board straight they can move a character without having to lookup row and column coordinates at all. To ensure that players understand this orientation I etched the appropriate compass rose into each side of the board in front of each player (see Fig. 2-7).[1]

Another source of synchronization errors during the movement phase came from the need for players to look back and forth between the board and the screen of the digital device to read the instructions. This movement often caused to players to repeat a single instruction or skip one or more instructions leading to synchronization errors. Many players addressed the difficulty of this interface by having one player read the instructions while the other player moved the pieces. This was a significantly suboptimal solution because it denied both players an equal chance to see patterns in character movement and, importantly, their opponent’s reaction to moves.

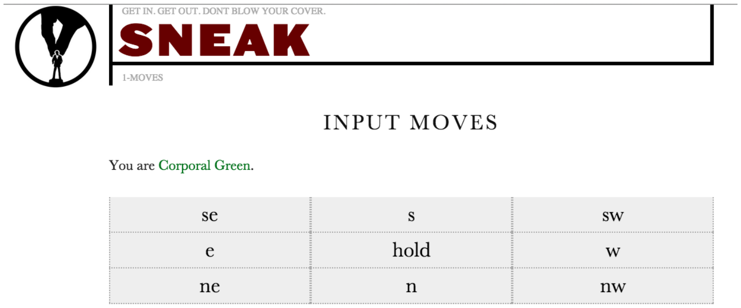


INPUT MOVES

You are Captain White.

nw	n	ne
w	hold	e
sw	s	se

FIG 5-4 DIFFERENT MOVE INPUT ORIENTATIONS BASED ON PLAYER BOARD POSITION. ABOVE: PLAYER ONE. BELOW: PLAYER TWO



INPUT MOVES

You are Corporal Green.

se	s	sw
e	hold	w
ne	n	nw

In order to overcome this problem I added spoken instructions to Sneak’s app. On each move phase, the game generates audio versions of each move instruction and plays them in sequence for the players. The audio instructions are assembled on the fly in software from a

[1] Absolute coordinates are now only used during setup when players are looking at a visual map of the board.

series of fragments pre-recorded by an announcer. Those audio instructions enable players to move the characters while both of them keep their eyes on the board and on each other.

Together these changes have dramatically reduced the frequency of synchronization problems caused by players moving pieces incorrectly.

Synchronization errors can also occur in the opposite direction: players can make mistakes when inputting their moves into the app. This type of problem arises when the spatial orientation of UI widgets does not match each player's perspective on the board. Specifically, in Sneak, each player sits on a different side of the board. Players move their characters by clicking on one of nine buttons representing the eight cardinal directions and the opportunity to hold (see Fig. 5-4, top).Initially, Sneak's interface presented all players with the same set of buttons for selecting moves. These buttons were oriented with north at the top of the screen. Since players sat at opposite sides of the board this interface would be upside down from the perspective of at least one player. Players often submitted moves other than those they intended when using this interface. To correct this problem, Sneak now presents a different move input screen to each player depending on which side of the board they are sitting. Players sitting on the north side of the board use an interface that matches their perspective on the board (see Fig. 5-4, bottom).

This change has virtually eliminated mistaken move inputs. However, I believe it could still be improved. Instead of providing buttons labeled with cardinal directions, the interface could show the local region of the map around the player's position, offering the chance to submit the move simply by clicking on the square to which they wanted to move their character. This proposed design would offer an immediacy between interface and in-game action that is common to good video game UI design.

Despite the significant improvements caused by these UI changes, the holy grail of synchronization still remains elusive in Sneak: a system that regularly checks to ensure that the physical board is in sync with the state of the app. Sneak does currently provide an optional "debug view" that shows an onscreen representation of the placement of all walls, characters, guns, and other game pieces. However, players have to actively choose to consult this view when they worry something has already gone wrong. In that circumstance it can act as an emergency fix to get the game back on track, but it is an ugly hack extrinsic to the game's systems and only useful as a measure of last resort.

An ideal "sync check" system would be integrated organically into the game's mechanics. Instead of requiring players to switch out of playing the game into a separate mode, it would lead players through checking synchronization within the game's systems. Perhaps it would incentivize players to catch out their opponents making an incorrect move or otherwise fold synchronization check into the game's competitive mechanics. I have not thus far found a mechanic to serve this purpose in Sneak, but such a pursuit could potentially lead to discovering designs for entirely different hybrid digital physical games.

6. The Use of Hidden Information

Asymmetries of information between players are one of the strongest sources of game dynamics in both digital and tabletop games. In tabletop games this often takes the form of bluffing or hidden traitor mechanics designed to create social pressure by requiring players to lie to each other.

Two essential, and very different examples, of this use of hidden information are Werewolf[1] and Poker. Poker is a family of card games involving varying combinations of private and public cards and turn-based betting. Bets are always public and in some variations of the game some public cards are shared amongst all players, but players also always posses a hand of cards known only to them. Gameplay consists of deciding bet amounts and attempting to bluff other players in regards to the quality of your hand. Werewolf (also known as Mafia) is a party game where a minority of players with a hidden traitor role attempt to disrupt the collaboration of the wider group to identify and eliminate them. Gameplay consists of conversations amongst the entire group in which the secret traitors attempt to influence the group's decision to avoid suspicion cast on them.

Both of these games involve hidden information and a public performance on the part of the players whose interpretation is informed by other players' suspicions about their hidden information. In addition to hidden roles, Werewolf players' actions are also hidden adding to the complexity of this performance and interpretation process. Hidden information is central to these games because of the social dynamics it creates. Players pay close attention to the social cues emitted by other players in the game. They watch other players receive information, attempt to mask their own reactions, and say things designed to elicit a telling response. The physical setups of these games are optimized around social interaction: players sit around a table looking at each other. The number of physical components is minimized to keep the focus on the people.

In digital games, hidden information usually operates significantly differently. For example in competitive online multiplayer shooters such as Call of Duty[2] players' positions and movements are hidden from each other not through concealed social information but through the vision constraints of the 3D game world. Online players are often connected through voice chat in

[1] Werewolf/Mafia, Dmitry Davidoff. 1986.

[2] Call of Duty series. Infinity Ward, Treyarch, and Sledgehammer Games. Activision. 2003-present.

these kinds of games leading to trash talking as the chief form of social interaction rather than deception. Local multiplayer variations of this formula such as Goldeneye 007[1] and Pikmin 2[2] add an interesting twist. Since players play on a shared screen in these games they could easily gain information about their opponents' intentions simply by looking at their interface. Hence sophisticated player strategies have evolved to manipulate the look direction of their characters in order to deny other players this information.

One intriguing new pattern of hidden information that has emerged in recent video games can be found in games like Spyparty[3] and the Assassin's Creed: Brotherhood multiplayer.[4] In both of these games, a small number of human players share a game space with a larger number of NPCs. Players hide themselves from each other by emulating the actions of those NPCs in order to avoid standing out and getting targeted by their human opponents. As explained in detail above in both the Perception of Randomness and Cost of Simulation chapters, Sneak creates a similar dynamic to these games, rewarding players for emulating NPC movement and for observing deviations from it by other players. This element of Sneak's design was explicitly inspired by these video games.

Despite this specific relationship, when it comes to the use of hidden information hybrid digital-physical games have more to learn from tabletop games than digital games. Tabletop games that include digital devices are still, fundamentally, tabletop games in their format. They involve two or more people sitting around a table talking to each and looking at each other. Hence, like other tabletop games, hidden information in these games is deeply intertwined with the social dynamics that emerge between players. Bluffing and hidden traitor mechanics (and other mechanics that derive from hidden information) are natural fits for digital tabletop games.

Further, tabletop games that include digital devices potentially posses a major advantage when implementing these mechanics over conventional tabletop games. Depending on the design of the game's interaction, the digital system will have a record of some or all of the information known by each player. Such games can use this information dynamically to balance individual player advantages or otherwise intervene by giving or denying additional information to one or more players. For example, Sneak knows exactly what information each player has

[1] Goldeneye 007. Martin Hollis, Rare. Nintendo. 1997.

[2] Pikmin 2. Shigefumi Hino and Masamichi Abe. Nintendo. 2004.

[3] SpyParty. Chris Hecker. Currently under development.

[4] Assassin's Creed: Brotherhood. Patrick Plourde. Ubisoft. 2010.

received about which character posses the plans. It can therefore detect if one player has a major informational advantage and, potentially direct NPC movement (or other game systems) to help the other players catch up, ensuring a competitive (and therefore exciting) outcome to every game. This kind of “rubber banding” is widely used in video games, particularly in racing titles where AI competitors adjust their speed in order to prevent players from getting too far ahead or behind. For example, the patent for the Nintendo GameCube game MarioKart: DoubleDash!! describes exactly such an algorithm.[1] Such systems are obviously impossible in conventional tabletop games, but become straightforward when digital devices have sufficient information about the game state. I will discuss a proposed design for an additional system in Sneak to implement exactly such a rubber banding balancing pattern below.

Additionally, including digital devices in tabletop games creates the potential for new patterns in how information is transmitted to and between players. These new types of games might involve one device that is passed between players or multiple devices each held by one player. Both of these formats allow players to pass information between each other without any physical token or spoken word that would reveal that action to other players in the game.

Because of its central use of secrecy Sneak does not explore this potential for inter-player communication, but it does use a similar mechanic for communicating between NPCs and players. Players receive information from the simulation of NPCs knowledge in private on each turn by viewing the digital app. Since players view the knowledge interface each turn regardless of whether or not they have learned anything new that round, this use of the device serves both to hide their identity and, ironically, to give other players an inflated impression of how much knowledge their opponent may have gained. In designing this portion of Sneak I wrestled with the differences between using one or multiple devices as well as other details in how players receive this information. I will discuss these design decisions in detail in this chapter.

The rest of this chapter will detail how Sneak’s design uses hidden information to shape the social dynamics between its players. That discussion is separated into two sections. First, I will cover the techniques I discovered for hiding information from players in a digital tabletop game.

I will describe how Sneak’s dialog system creates and transmits information to players and how its temporal dynamics have been designed to balance the game. In the previous chapter on The Cost of Simulation I discussed in detail how players react to hidden information in the game, both social and factual. I showed that players have a strong tendency to jump to incorrect

[1] Yasuyuki Ohyagi and Katsuhisa Satou. Racing Game Program and Video Game Device. Nintendo Co., Ltd., assignee. Patent US 7278913 B2. Oct.-Nov. 2007. Print.

conclusions when presented with a factual deduction problem (figuring out who has the plans) and to perform amazing acts of intuition when confronted with a parallel social problem (guessing which character is controlled by their opponent). In this chapter, I will revisit this issue with an eye towards how Sneak could do a better job supporting active deduction of opponent identity. To that end I will describe a proposed additional mechanic that would give players the ability to narrow down their suspicions of their opponents' identities. In discussion that mechanic I will highlight the risk and reward tradeoffs that I have found to be key to successful hidden information mechanics in Sneak.

Following that section, I will review the ways in which Sneak's design uses hidden information to shape social dynamics between players. I will examine the design issues of using one versus multiple devices. I will share the design process that lead to the addition of the poisoning mechanic to the game, which introduced additional hidden information in order to give the players more freedom of action. Finally, I will propose two additional mechanics that could improve Sneak's design in this area in future iterations: a rubber banding mechanic for dynamic balancing and an addition to the game's interface to allow players to explicitly track their suspicions about the other player's identity.

6.1 Hiding Information

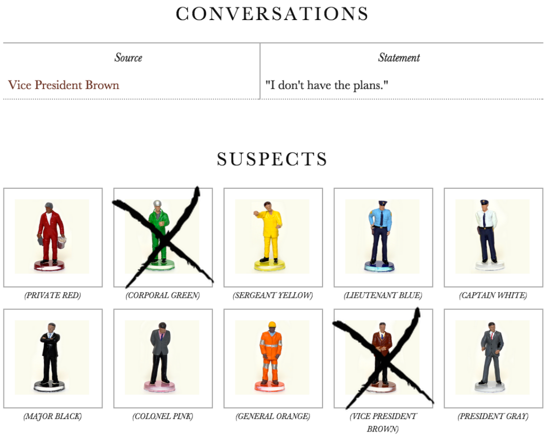
In Section 2.5, Game Dynamics, I described the "three act" structure which Sneak's design is meant to induce in each game. In the first act, players move around the board engaging in dialogs to try to discover which NPC has the plans. After ten or so turns, at least one player has found out who has the plans. Possibly one player has even picked up the plans. At this point players who do not know the location of the plans become nervous. Both players have begun to develop suspicions about their opponent's identity. Act three brings things together for a dramatic conclusion. If a player has the plans they attempt to sneak towards the exit. Players who know which NPC has the plans race to chase down that character. Players who do not know the location of the plans start to think about shooting their chief suspect. The game resolves either by a player escaping with the plans or (more frequently) with an exchange of gunfire and poison and dead bodies on the board.

The main tool Sneak uses to produce this act structure is the hiding, and eventual revealing, of information. Information about who has the plans propagates through the character population via dialogs. The frequency and effect of dialogs depend on Sneak's movement and knowledge systems. These systems are designed and tuned to produce a diffusion of information between characters which produces the particular pace (and inter-player dynamics) that make up the game's three act structure.

In a two-player game of Sneak there are only three pieces of information that are hidden:

- The color of player one's character
- The color of player two's character
- Which character has the plans

All other information information in the game is publicly known.[1] At the start of the game, knowledge about this information is equally sparse. Each player knows only their own character's color. Sneak also models each character's knowledge about the location of the plans. Each character begins the game knowing whether or not they have the plans.



CONVERSATIONS

Source	Statement
Vice President Brown	"I don't have the plans."

SUSPECTS

FIG 6-1 PLAN INFORMATION IS COMMUNICATED TO PLAYERS IN THE FORM OF DIALOGS.

On each turn, characters that end up on the same square participate in dialogs. During a dialog, each character adds their dialog partner's knowledge to their own. For example, say brown and green conduct a dialog on the second turn. Each of them only knows that they themselves do not have the plans. After their dialog, they each have an additional piece of information. Now they both know that brown and green do not have the plans.

If one of these characters is a player, on their turn they will be presented with this information in the form of a conversation (see Fig. 6-1). It will also be summarized for them in the suspect display. NPCs also retain this information even though it is not displayed to the players.

Vitally, with each dialog, every character communicates the entirety of their current knowledge to their conversation partner. This holds for players as well as NPCs. Players have no choice in whether they communicate their current knowledge to characters they encounter.

[1] Discounting here the algorithms that govern NPC behavior, which I discussed at length in earlier chapters. Here I'm referring to specific individual atoms of information with autonomous meaning rather than the rules governing of whole systems.

Given these rules, we can describe the spread of information about the plans through the character population using the Bass Model.[1] Originally created to model the spread of product marketing, the Bass Model is frequently used as a simple model of the spread of contagious infection in a social network. It captures the dynamics of network diffusion as a function of the rate of spontaneous infection, *p*, and the probability that infection is communicated between any two agents, *q*. In Sneak, the "contagion" we are modeling is the information about the plans. Characters become "infected" through dialogs.

The discrete version of the Bass Model computes the percentage of the population infected at any given time step, *F(t), as:*

$$F(t) = F(t-1) + p*(1-F(t-1)) + q*(1-F(t-1))*F(t-1) \quad | \textit{Eq.6-1}$$

In Sneak, the first term, "F(t-1)", corresponds to the number of characters who knew the location of the plans in the previous turn. The second term, "p*(1-F(t-1))", is the "spontaneous infection rate" times the number of uninfected agents. In Sneak this corresponds to the probability of a character who does not know the location of the plans learning it from a source other than a dialog. This only happens at the very start of the game and only for the character actually holding the plans. So, in our model, *p* is actually a function of *t*:

$$p(t) = 1/10 \ (\text{if } t = 0)$$
$$p(t) = 0 \ (\text{if } t > 0)$$

The next term in the model, "q*(1-F(t-1))*F(t-1)" is the "contagion rate" multiplied by the total possible number of encounters between infected and uninfected agents. In Sneak, characters that come into contact always exchange information. So you might assume that the contagion rate would be 1.0. However, the number of characters that come into contact with each other varies depending based on initial character placement, NPC pathfinding algorithm, the connectivity of the generated map, and the movements of the players. Characters with information about the plans can only "infect" characters with whom they exchange dialogs. Therefore, we can treat the average number of conversations per turn as the *q* in our model.

Now, with this model in place we can begin to understand how Sneak's design parameters affect the spread of information in the game. In combination, Sneak's generation and movement

[1] E M. Bass, "A New Product Growth Model for Consumer Durables". Management Science 15(5) 215-227. 1969

systems determine the average number of conversations that will take place in a turn. As just explained, this average determines the *q* parameter of our model. So, by looking at the model's predictions for information diffusion at various values of *q* we can discover how to tune these systems to achieve the information diffusion curve we want in order to achieve the three act structure I described at the start of this section. Fig. 6-2 shows the output from Eq. 6-1 over 25 turns for systems with an average number of conversations per turn ranging from one to five.

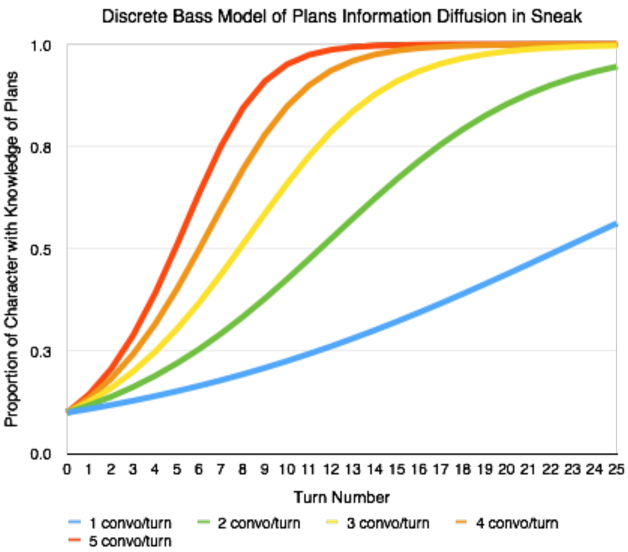


FIG 6-2 BASS MODEL OF INFORMATION DIFFUSION IN SNEAK SHOWING EFFECT OF VARIATION IN NUMBER OF CONVERSATIONS PER TURN.

As you can see in that graph, when the number of conversations per turn is low (zero or 1) the information diffusion curve is so flat that its inflection points are barely noticeable. On the other extreme, systems that produce four or five conversations per turn rise so fast and reach saturation so quickly that they also look nearly linear.

The three conversations per turn curve is the one that best reproduces the three act structure we are seeking. It has a slow growth of information for the first three to five turns that then accelerates rapidly until about turn 13 where it flattens out again after the information has nearly completely spread.

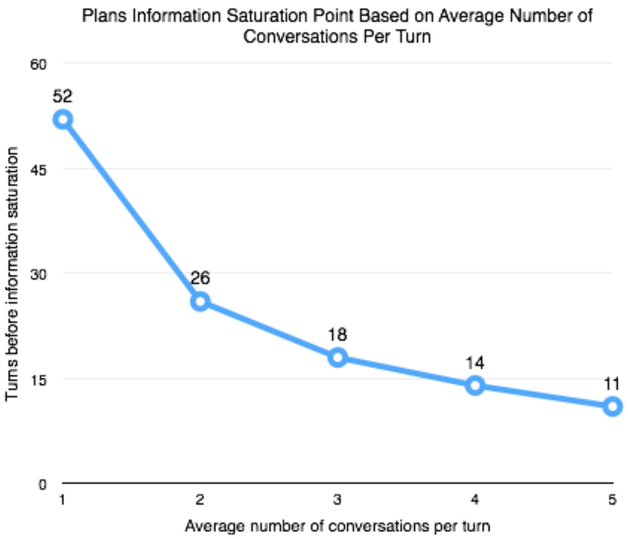


FIG 6-3 EXPECTED TURN AT WHICH ALL CHARACTERS KNOW THE LOCATION OF THE PLANS BASED ON THE AVERAGE NUMBER OF CONVERSATIONS PER TURN

Another useful metric produced by these curves is the estimated turn number at which all 10 characters should have information about the plans (Fig. 6-3). This number acts as something like an upper bound on the length of the game. Even if the players are the last two characters to find out the information, by this point they will have entered the final act of the game. Games that average three conversations per turn reach this point after 18 turns. If we assume an average of one to two minutes of time to play each round[1] then this sets out an outer bound of 20-40 minutes for the game. Since this is the desired play time for the game and this information curve looks correct, I have attempted to tune Sneak's systems to produce an average of three conversations per turn.

These different diffusion rates also present opportunities which Sneak's design does not yet exploit. Sneak has the ability to speed up or slow down the game by altering the conversation rate. It also has the ability to balance the competition between multiple players by imposing different diffusion rates on each of them. In the next section, I'll describe a mechanic I propose to add to Sneak in the future to take advantage of both of these possibilities.

6.2 Designing Social Dynamics with Hidden Information

Where the previous section dealt with the propagation of information in Sneak from a systems perspective, this section looks at the same problem from the perspective of user interface and social dynamics. It focuses on the specifics of how Sneak communicates information to players through the digital app's interface and through the game's mechanics. I will also discus a number of proposed future mechanics that could be added to the game to potentially improve how it handles hidden information.

In my initial conception of Sneak, I assumed that it would be implemented as a smartphone app, each player would run the app on a separate device, and those devices would communicate over a network. I began to prototype the game without network connectivity simply to shorten the time necessary to produce the first playable prototype. This presented the problem of how to allow multiple players to submit information into the app without learning their opponent's identity. At first I solved this problem by personally acting as the interface to the app. I communicated with the players by passing paper slips back and forth. I entered their moves into the computer myself and used the slips to tell them the results of dialogs.

This primitive system was functional enough to work out the initial outline of the game's design. However, I wanted to remove myself from the equation. So, my next round of prototypes added a simple intermediate screen between each player's turn using the app. This made it

[1] This is the average turn length I've observed in typical playtests with some notable outliers caused by unusual player behavior (see "How Players Model NPC Behavior" in Chapter 5).

possible for two people to play the game by passing a single device back and forth, an unwieldy laptop in these initial playtests.

Initially I still assumed this passing interface was a stopgap measure until I could implement networking. However, when I watched the playtests that used this setup something interesting happened. Firstly, the moment of passing the device back and forth became a fixed point of social interaction between the two players. They were forced to look at each other and perform the basic social negotiation required to carefully pass a heavy, fragile object between two people. This seemed to have an impact on the atmosphere during the game. The players talked to each other more even when not passing the computer.

Secondly, while one player used the laptop to submit their moves or receive information I noticed that their opponent would keep an eye on them. This attention was not intensely focused; players tended to move their gaze back and forth between the board and their opponent. But, as I observed players' uncanny accuracy at intuiting their opponents' identity, I came to believe that it derives from the subtle social cues they pick up through this attention. As I observed in Chapter 4, The Cost of Simulation, further playtests also revealed that this intuition can be disrupted by players' failure to focus sufficiently on their opponent. In that chapter I described a playtest in which a player took extensive notes on game state. Spending their attention in that activity rather than observing their opponent lead that player to the worst failure of social intuition I have observed in any playtest.

Giving each player their own device presents the danger of reproducing that experience for every player of Sneak. Psychological field studies have shown that the use of smartphones during conversations significantly reduces the perceived quality of conversation as well as the level of empathetic concern on the part of each participant.[1] Conversation quality and, especially, empathetic concern, are at the heart of Sneak's gameplay.

Providing each player with a private device would offer them two distractions which could have a deleterious effect on conversation quality and empathy. Players would be tempted to interact with the game's interface during their opponent's turn rather than paying attention to them. Such an interface would naturally give them the ability to review their current knowledge, plan their future turns, and keep tracking of their suspicions. As demonstrated with the note-taking playtester, these actions distract from attending to social cues. Secondly, any private digital device used to play the game would, of course, have other apps and functions present on

[1] Shalini Misra, Lulu Cheng, Jamie Genevie, Miao Yuan, "The iPhone Effect: The Quality of In-Person Social Interactions in the Presence of Mobile Devices". Environment and Behavior. July 1, 2014.

it. Players would be tempted to multitask, leaving the game app to check email, search the web, or respond to notifications. Both of these dangers are dramatically reduced, or even eliminated, by the use of a single device which is shared between the players.

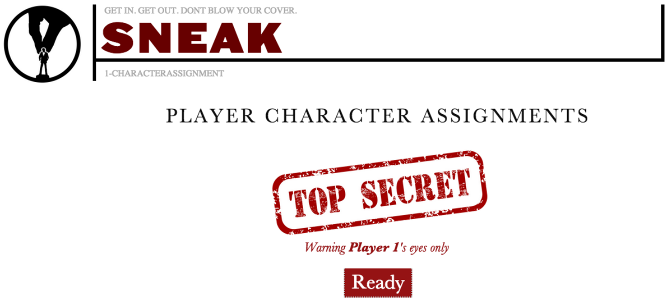

GET IN. GET OUT. DONT BLOW YOUR COVER.
SNEAK
1-CHARACTERASSIGNMENT
PLAYER CHARACTER ASSIGNMENTS
TOP SECRET
Warning Player 1's eyes only
Ready

FIG 6-4 SNEAK'S PASS SCREEN HELPS KEEP INFORMATION HIDDEN.

Designing a hybrid tabletop game around a single shared device does have limitations and challenges. On a tactical level, using a single device poses specific design and implementation problems around the player interface for passing the device between the players. In order to maintain the integrity of hidden information during device passing, the game must present a neutral screen that can be shown to all players while the device is passed. In designing Sneak's implementation of this screen, I found that it is difficult to successfully stop players from accidentally clicking through and seeing their opponents private information. Players reflexively click buttons provided to them without reading the associated text. In response, Sneak's current design for this screen uses aggressive colors and design to attempt to stop them (see Fig. 6-4). This has improved the problem, largely eliminating players simply clicking through it, but players still struggle somewhat with the interaction. Further research is needed in this interaction pattern. Particularly worth exploring is the idea of a more active "unlock" interface for the player receiving the device, perhaps based on a secret code or gesture known only to them.

Shared-device play also imposes a broader mechanical limitation. In games using this format, information and interfaces can only be provided to a single player at a time. This restricts the use of real time or parallel play patterns. In the next chapter, I will describe a mechanic I experimented with during the development of Sneak that would have required the use of multiple devices for exactly these reasons. As I will explain, this mechanic did not make it into the game due to its bookkeeping cost. However, the potential of rich player-player interaction it offered may have made it worth including despite the downsides in player attention described here. Creating direct player-to-player interactions has been one of the biggest difficulties in designing Sneak thus far. Designers of hybrid tabletop games face a continual tension between the deleterious effect multiple devices have on social interaction and the opportunities they offer for creating new social patterns.

Early in Sneak’s development, I discovered that its focus on stealth and deception limited the actions players could take to directly effect each other. The only action the game offered to actively attack an opponent was firing a gun, which came with a very high associated risk as it revealed the player's identity. I wanted to add mechanics that would give players more ways to actively alter the state of the game without necessarily revealing their identity.

To that end, I added the game’s current poisoning mechanic. Poisoning allows players to kill a character on the board without immediately revealing their identity. However, to do so they must be on the same square as the character they wish to kill. Further, the poison takes effect after a delay (randomly determined to be between three and five turns) so it is not completely within their control and cannot stop a player about to reach the exit and win the game. The combination of the proximity requirement and this random trigger time mean that the using poison is still risky for players. If no other characters are near their target, a poisoning death may in fact make their identity obvious to other players. Each player also only gets one dose of poison to use per game.These constraints are meant to keep fear of discovery present for players even while giving them this new tool for attacking their opponents. Poisoning allows players to act on the imperfect information the game has revealed to them while limiting the amount of information about their own hidden identity the introduce into the system. In playtests, I have found that poisoning deaths are dramatic moments that quickly shift the social dynamics between players.

Currently in Sneak, the game’s central dialog mechanic allows players to incremental learn information about the location of the plans. On the other hand, there is no equivalent mechanic that lets them reduce the number of suspects for their opponent’s identity. In future development of the game I propose to add a mechanics designed to do just that: bugging.

In this proposed mechanic, players would have the option, once per game, of adding a “bug” to any character with whom they intersect. A future information learned by bugged character would immediately be reported back to the player that bugged them. However, if a player encountered a character bugged by their opponent. They would detect that fact and the game would present them with the suspects who may have placed the bug: a list of every character with which the bugged character had previously interacted.

Bugging would accomplish a number of things simultaneously: it would accelerate the rate at which players gain information about the plans, giving them some control over the rate of information diffusion in the game. This would let players compensate for games where the coincidence of the generated board and their initial placement denied them of sufficient opportunities for dialogs. Secondly, it would give players another interesting decision to make in which they needed to pick their bugging target by balancing the risk of revealing their identity with the potential to maximize information gain. The earlier a bug is placed the more information it will yield for a player, but also the greater the odds the bugged character will encounter the player's opponent when the list of its contacts is still very short. Finally, when players did encountered characters bugged by their opponents the information they gained would significantly reduce the number of characters who might be controlled by their opponent. This would allow players to focus their observations through the rest of the game, increasing the probability of the social intuition correctly identifying their opponent.

Finally, as mentioned in the previous section, the relationship between information diffusion and NPC pathfinding offer a powerful potential opportunity for Sneak to perform dynamic game balancing. Since all character and player knowledge is known to Sneak's software at all times, Sneak can detect situations in which one player has a major informational advantage or when the rate of dialogs taking place amongst all characters has fallen significantly below the optimal average of three per turn. When these situations arise, Sneak can take action to manipulate the flow of information within the game to rectify them. Specifically, Sneak can alter the pathfinding destinations of one or more NPCs to alter the frequency with which they will intersect with each other, with the trailing player, or with both players.

To take advantage of this possibility, I propose an additional future mechanic for Sneak called "meetings". When Sneak detects an information deficit, it can choose multiple NPCs within a radius of each other and assign them a new, common destination. When the NPCs all arrive on that destination, the information known to each of them will be distributed amongst all of them, increasing the total information diffusion in the game. If the game detects an imbalance amongst the two players, it can simply set the meeting location on or near the trailing player's location and the chances of them catching up will significantly increase.

An interesting side effect of this mechanic is that it will alter NPC behavior and therefore require an appropriate compensation by players in their perception and emulation of that behavior. To prevent meetings from potentially revealing the identity of a player who fails to match the changed movement pattern of surrounding characters, meetings should never involve every character in a given area of the board. There should always be characters who do not participate in meetings in order to give players plausible deniability for their own movements.

7. The Role of Bookkeeping

In tabletop games, none of the pieces or systems move on their own. They always require the player to animate them. Hence, when tabletop games want to keep track of significant amounts of information, they impose a commensurate amount of bookkeeping work on the player. For every quantity that must be tracked, players must move counters, or rotate dice, or take chits, or move meeples, etc.

This bookkeeping requirement imposes a cost on systems in tabletop games paid in player attention and confusion. As systems get more complex and require more numbers to be tracked, players must remember to update more counters and the cognitive load on the players increases. How quickly these costs rise is determined by what we might think of as the 'bookkeeping efficiency' of a game. Bookkeeping efficiency measures how much systems complexity or how many interesting decisions a game is able to include for a fixed cost of bookkeeping.

On one extreme side of the spectrum are games like Sorry![1] and Candy Land[2]. Targeted at young children, these games attempt to minimize bookkeeping and complexity more generally. However, they achieve this simplicity by nearly eliminating decision making altogether. From a certain perspective these games consist solely of bookkeeping. Their designs are highly bookkeeping inefficient so in order to remain simple they are forced to eliminate all but the most rudimentary gameplay.

Good design, on the other hand, increases bookkeeping efficiency leading to higher complexity games that are still playable. For example, take Carcassonne.[3] An icon of the New German Board Game movement, Carcassonne is a tile-placement game with a medieval theme. Players construct a landscape by connecting terrain tiles to a growing board, scoring points based on the roads, churches, and cities they create. Bookkeeping in Carcassonne is very light, consisting solely of a scoring track players advance when they complete features of the landscape and take credit for them. The interplay between collaborative and competitive dynamics and the way tiles are released when features are scored leads to surprisingly deep

[1] Sorry!, Paul T. Haskell, Jr. and William Henry Storey. Basic Fun, Inc. 1929

[2] Candy Land, Eleanor Abbott. Hasbro. 1949

[3] Carcassonne, Klaus-Jürgen Wrede. Hans im Glück Verlags-GmbH. 2000

gameplay given the quite light cognitive load the game's minimal bookkeeping places on players.

An even more extreme example of bookkeeping efficiency is chess. Chess is almost entirely bookkeeping free. Only subtle edge case rules such as castling are not completely revealed simply by looking at the board. There is a near perfect isomorphism between game state and game 'interface'. And, obviously, chess is a very deep game yielding evolving strategies that have kept people interested for millennia.

On the complex end of the spectrum, there are also many games that achieve high systems complexity and rich decision making but at the cost of extraordinary amounts of bookkeeping. No genre of game more fits this description than tabletop war gaming. For example, Fire in the East[1], a war game about World War Two's eastern front includes over 2500 individual counters, six large maps, extensive rulebooks covering everything from Soviet and German rail gauge to weather effects. While games like these have their adherents, their audience will always be strictly limited in size by the intensity of bookkeeping effort required to play them. The sheer time and effort needed to play even a single session these games (let alone learn them) is enough to keep away all but the most devoted players.

If you are tempted to dismiss this problem as only affecting niche hobby games, remember that Risk[2] is a widely-played game that is famous for the high toll in bookkeeping that it imposes on players.[3] And many of the more ambitious of the New German Board Games that have gained such popularity amongst board game devotees are on the complex side of the spectrum.

While good design can improve the bookkeeping efficiency of a game there are limits to the amount of simplification it can achieve and the kinds of systems it can improve. Hybrid digital-physical tabletop games have the potential to increase bookkeeping efficiency even further, beyond what is possible in analog design no matter how clever. Many of the operations typically involved in bookkeeping are also the bread and butter of digital systems: counting things, adding and subtracting numbers, remembering results, etc. By relieving human players of some of this effort hybrid games can enable higher systems complexity and more interesting decisions while remaining at enjoyable levels of bookkeeping effort.

[1] Fire in the East, John Astell, Rich Banner, and Frank Chadwick. GDW Games. 1984

[2] Risk, Albert Lamorisse. Parker Brothers. 1957

[3] When I was young, my parents played in a long-running Risk game with a group of friends. After months of one continued game, the woman storing the board between sessions accidentally bumped, it disrupting the pieces. In an ensuing fit of rage she hurled the pieces to the ground, vacuumed them up, and threw out the board. She could not face the prospect of having to reproduce all of that lost bookkeeping effort a second time.

While we have thus far been considering bookkeeping as a cost, there are also advantages to bookkeeping in that it holds players' attention and keeps them as active participants by requiring them to take a constant series of actions to keep the game moving. Further, while social interaction is the heart of tabletop gaming, without any alternative activity the sustained high-intensity social interaction of party games like Werewolf can be exhausting and stressful for players. Physical bookkeeping provides an alternative attentional focus that can moderate this situation.

Digital games have no direct equivalent of bookkeeping. They typically use the power of computation to invisibly track all but the information most essential for player interaction.

In this chapter, I will explore how bookkeeping works in Sneak. I will look at the bookkeeping tasks from which Sneak saves its players, those I chose to impose on them, and the logic I used to distinguish the two. Specifically, I will argue that game systems that are particularly important for the player to internalize should be the last turned over to digital systems because that would remove them from the players' attention and deny them the ability to inhabit those systems.

I will also describe a prototype for a system that I wanted to add to Sneak — that was, in fact, a major part of Sneak's design as I first imagined it — but that I was forced to discard because the bookkeeping cost it imposed on players was simply too great.

7.1 Designing For and Against Bookkeeping

There are many different ways a hybrid digital tabletop game can encourage, support, or eliminate, bookkeeping. These approaches offer players a range of levels of support in keeping track of in-game information and impose on them complementary quantities of cognitive load. On one end of the spectrum, the game's digital interface can completely ignore a game system, leaving the player to note and recall the relevant information themselves. On the opposite extreme, the game can perform a complete analysis of the information available to the player and present them with conclusions about the appropriate action to take in response. Between these two extremes lies a gradient of intermediary options. The game can record information that was available to the player in a log to relieve them of the cognitive load and working memory costs of retaining it. It can perform analysis of the current state of the game to summarize the costs of move options available to the player to relieve them of having to perform the calculation themselves. The game's interface can highlight the most important current information to aid players in the critical task of discerning it and to help them focus their attention. Each of these techniques, and many others, can be applied to one or more of the game's systems. The more of them the game uses, the lighter the overall bookkeeping cost imposed on the player, the lower the cognitive load they will experience, and the less attention they will spend on the game's systems. By using these tools asymmetrically across the games different systems, designers of hybrid tabletop games can focus the player's attentions on the game's most important systems

In Sneak, there are two general categories of information that players might want to track: factual information about which NPC has the plans and social information about which character they suspect their opponent controls. These two types of information play very different roles in the game and, as I have iterated Sneak's design in response to playtesting, I have discovered that they each need a distinct bookkeeping approach commensurate with that role.

Inter-player fears and suspicions are at the heart of the emotional and social experience of playing Sneak. Building theories about their opponent's identity, fearing that their moves will reveal their own identity, eliminating suspects as NPCs — these actions focus players on their opponent and drive them towards the game's primary dynamics and most effective emotional experiences.

In order to put these social systems at the center of the game's design, I have used bookkeeping elimination techniques more heavily on Sneak's non-social elements such as NPC pathfinding and information about the possession of the plans. I have been careful to avoid relieving players of the responsibility of figuring out their opponent's identity. Ironically, because this task is so central to the game, it is the one whose bookkeeping is least supported by the game's interface.

The first example of of Sneak uses computation to reduce bookkeeping costs is so pervasive in the game it can be hard to see: NPC pathfinding. In chapter four, I explained the NPC pathfinding algorithm in some detail. Since that system's implementation is drawn from existing software approaches it might be counterintuitive to imagine it as something that could be executed by players. But, it is possible to imagine a version of Sneak that requires players to perform the pathfinding calculations while still retaining its other mechanics. Imagine a purely analog version of Sneak that still retained the same mechanics and movement logic for its NPCs. Such a hypothetical game would require a non-playing moderator who secretly rolled random destinations for the NPCs and then performed painstaking calculations to ensure that NPCs followed consistent pathfinding logic while moving towards those destinations. The moderator would also have to receive moves from players in some format and shuffle them into the calculated NPC movements in order to keep player identities secret. Having essentially played the role of that moderator in the earliest prototypes of Sneak before the digital systems were fully in place, I can report that it is a painstaking and boring task and that having it performed by a human radically slows down play.

Since Sneak is not a game that is primarily about the logic of pathfinding, eliminating all of this bookkeeping from player consciousness by turning it over to the game's digital systems is clearly the right design decision. However, that does not mean that the game should do everything it can to remove from players the need to think about NPC movement.

From the earliest playtests of Sneak, it was clear that trying to remember the sequences of moves made by characters was a difficult task for players. Depending on how long each turn of the game took, players reported that they could only remember moves from the previous one to three rounds. In earlier chapters I have discussed the design decisions that effect this memory task in some detail.[1]

It would be easy for the game's digital interface to significantly support this memory task. The game's software has a complete log of every move made by every character. It could present players with movement tracks showing each square visited by each character throughout the course of the entire game. I have chosen not to do this because it would remove attention not from the game's movement system, but from its social systems. A large part of player strategy for figuring out the identity of their opponent is to closely track the movement of all of the characters looking for clues as to which reveal signs of human control. While players may not turn out to be very effective at this task[2] it plays a crucial role in focusing players attention on their opponents actions and increasing their sensitivity to social cues.

My one concession to this difficulty has been to offer players an optional mechanic where, if both players agree, they can chose to orient each figure on the board to face in the direction of that character's last move. This acts as a small aid to memory and lets players select a level of difficulty for that aspect of game and giving them a mechanism to level the playing field of working memory between them.

Another area where the game's digital interface relieves the players of a significant bookkeeping load is in tracking information they have learned from dialogs. On the dialog reveal screen, Sneak presents players with the information they have gained this turn in the form of

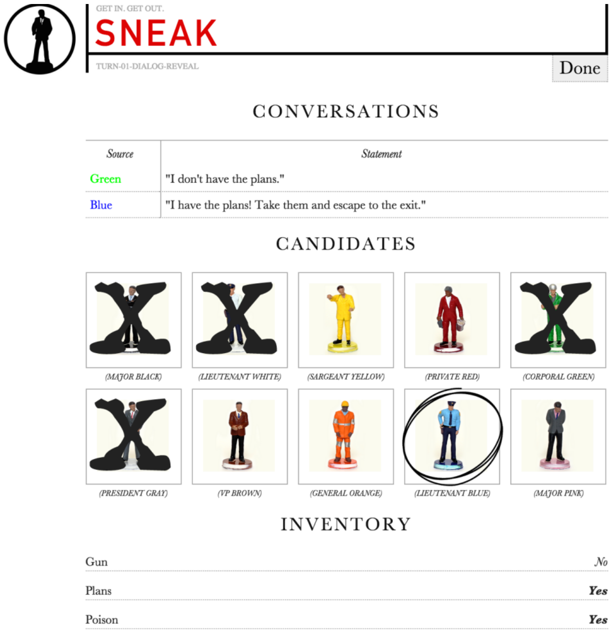
[1] See chapter four, The Perception of Randomness and chapter six, The Cost of Simulation.

[2] See chapter six for an analysis of their failures in this regard.

quotes from the characters providing the information: “Source: Green, Statement: ‘I don’t have the plans’”, “Source: Blue, Statement: ‘Red has the plans.’”

It would be possible for that to be the extent of the interface Sneak provided for tracking information about the plans. The game could leave players to keep track of this information themselves, a process for which they would almost certainly demand the ability to use pen and paper. Instead, Sneak keeps track of this information for players and shows it to them each turn when they visit this screen.

Initially, this information took the form of a list of text statements about the current state of the player’s knowledge: “Blue didn’t have the plans when you saw them five turns ago,” “Green didn’t have the plans last turn according to Red”, etc. However, in an early playtest a player actually missed information about the location of the plans because it was buried in a long list of sentences like these.[1] This made the player appropriately frustrated. They reported that losing the game because they had simply failed to notice this felt arbitrary. To prevent this frustration, I converted the display of this information into a graphical format that makes it impossible for players to miss which candidates they’ve eliminated and which one they know to have the plans (Fig 7-1).



CONVERSATIONS

Source	Statement
Green	"I don't have the plans."
Blue	"I have the plans! Take them and escape to the exit."

INVENTORY

Gun	*No*
Plans	***Yes***
Poison	***Yes***

FIG 7-1 GRAPHICAL SUMMARY OF PLAYER’S KNOWLEDGE ABOUT THE PLANS

While it occupies a large portion of the game’s turn-by-turn mechanics, the chase for the plans is primarily a background against which the social interaction between the players can take place. It gives players a set of actions to take so that they can reveal themselves to their opponents in executing them. In a vast majority of the playtests conducted thus far, players win by shooting or poisoning their opponent rather than actually escaping to the exit with the plans. The game is not about defeating the map and the NPCs in order to retrieve the plans, it is about deceiving your opponents.

[1] See Appendix A for playtesting notes.

Currently Sneak offers the players no systems to aid in keeping track of their suspicions about their opponents identity. However, there are two future changes to the game's design I am currently considering that would change this fact. In section 6.2, Designing Social Dynamics with Hidden Information, I described a proposed additional mechanic called "bugging". This mechanic would provide players with concrete information that would narrow down their opponents identity to a smaller list of characters. When I implement this mechanic, I intend to provide a supporting interface, parallel to that shown in Fig. 7-1, demonstrating to the players which suspects remain. While at first this might seem like reducing the bookkeeping burden in a system where we do want the player's attention, I believe that this would not be the case. The goal of the proposed bugging mechanic is to reduce the list of suspect's for the opposing player's identity so that players can focus their social attention on their opponents interactions with the remaining suspects. Relieving them of the bookkeeping required to remember this list of suspects frees up cognitive resources for this social task.

I am less confident in the design of the second proposed social bookkeeping system. Currently the only information not available to the game's digital systems is the players' current suspicions about the identity of their opponents. If the game offered players an interface for tracking these suspicions it could capture that information and potentially use it in dynamically balancing the game. However, the playtest, described in detail in chapter six, where a player kept their own version of these notes on paper demonstrated that this kind of social bookkeeping has the danger of disrupting players' social intuition by distracting them from paying careful attention to the social cues emitted by their opponents. So, while this system may be worth testing in future prototypes, it should be closely monitored for any deleterious effects it has on the effectiveness of players intuition.

7.2 Systems Limited By Bookkeeping Requirements

From my earliest conception of sneak, I imagined that the dialogs would be the intersection between the game's informational systems and its social systems. In addition to being the mechanism by which knowledge about the plans spread amongst the game's characters they would be performed as actual dialogs between players. My plan was that every time a dialog occurred between characters on the board, the app would provide lines to both players, which they would read. When one of the characters involved in the dialog actually belonged to one of the players, then their opponent would actually be watching and speaking with them as they received important in-game information.

This imagined system was intended to force structured social interaction between players that would knock loose social cues. I hypothesized that receiving information during these dialogs would cause have to hide their reactions while their opponents watched closely to attempt to discern them.

Early in the design of Sneak I built a paper prototype of this dialog mechanic. I found inspiration for the content of the dialogs in the spy movie cliché of the pre-arranged code phrase. For example in the James Bond film, *From Russia With Love,*[1] two spies meeting for the first time use the following code phrase to identity each other:

```
Agent A: Can I borrow a match?
Agent B: I use a lighter.
Agent A: That's better still.
Agent B: Until they go wrong.
```

One of the challenges facing the system was that it needed to communicate information about characters' possession of the plans to one player only without the other players knowing even though the dialog would be conducted in public. To solve this problem, I came up with the idea of giving each player a category of code phrase, each consisting of a category of words, i.e. European cities, types of car, items of clothing, etc. Dialogs could include the listening player's code phrase if the information they provided was true and elide it if their information was false. Players would receive their code phrase at the beginning of the game and keep it secret from the other players. So only they would be able to tell if what they were hearing was true.

To illustrate, here is an example dialog inspired by that exchange from the James Bond movie. For this example, say that red is receiving information from white and that they have been assigned the code phrase category "type of profession":

```
Red:   Can I borrow a match?
White: Sorry mate. I bought a box of lighters, but the black, white,
        and orange ones are all out of fuel.
```

[1] From Russia With Love, Eon Productions. 1963.

```
Red:   No worries. Any idea where I can find a light?
White: Well, the [type of profession] is a smoker. Maybe try them?
```

In this example white is telling red that black, white, and orange do not have the plans. Their final line in the dialog tells red that the information provided here is true. The inverse example would look like this:

```
Red:   Can I borrow a match?
White: Sorry mate. I've got a nice orange lighter at home. If I'd
         planned ahead I would have brought it.
Red:   No worries. Any idea where I can find a light?
White: Why not try quitting smoking? It's worse for you than eating
        a ton of [type of food].
```

In this version white tells red that orange has the plans. But their final line contains “type of food,” which is not red’s code phrase and so the information is false.

Both of these examples include the code phrase as an expression in brackets: “[type of profession]” or “[type of food]”. These are meant to act as placeholders. The player speaking that line is instructed to fill them in with a concrete example of the category: “policeman” or “banker” in the first example, “bananas” or “candy” in the second. This additional twist was meant to add challenge to the speaking player, asking them to do a little bit of improvisation as something to juggle while they tried to watch their opponent closely for any reaction.

I built a paper prototype of this system by writing the variations of dialogs on a series of index cards. I constructed a series of challenges where I would secretly tell one player their code phrase, pass out the complete cards for one conversation variation, and have the players conduct the dialog. After each dialog I would ask the other player to guess whether their opponent heard their true code phrase that time or not.

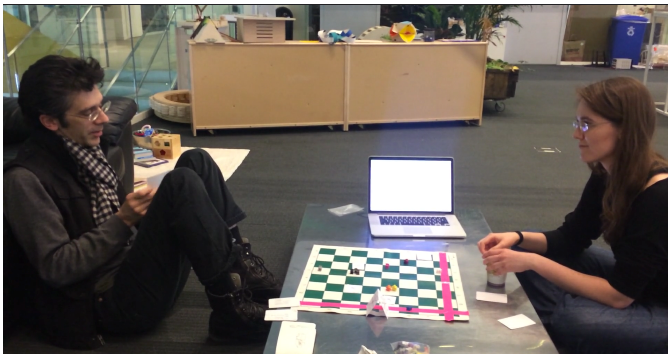


FIG 7-2 PLAYERS TRYING OUT THE FAILED PROTOTYPE OF SNEAK’S DIALOG SYSTEM.

I found that players could not reliably detect when their opponents had, in fact, heard the correct code phrase. This conversation system required them to divide their attention between their own lines and the other player's reactions. This limited their ability to pick up on the social cues emitted by the other player, which were also quite slight since the listening player was themselves struggling to understand the communicated information rather than simply reacting to it. In order to react, the listening player would have to keep in their head the colors of characters that were mentioned, their own code phrase, and the category of the words spoken by their opponent. In order to judge their opponent's reaction, the speaking player would have to keep in mind most of these same elements.

I even specifically told listening players not to worry about retaining the information contained in the messages. My plan had been to reproduce that information in the digital app so that players would not have to recall it. Even with this aspect of the burden removed from them, players were still overcome by the bookkeeping requirements necessary for this mechanic to be effective. The cognitive load of keeping all of that information straight rendered whatever social cues were present muted and unreadable.

Hence, I decided not to pursue this mechanic in Sneak. I think some version of this dialog mechanic, whether connected to Sneak's other systems or as a standalone game of its own, might be interesting to pursue in the future. The attentional challenges and interaction patterns it offers seem promising as the platform for some kind of gameplay, but for now I could not find a way of using digital technology to achieve enough bookkeeping efficiency to make this mechanic viable.

9. Evaluation and Conclusion

As a medium, games are perhaps less complete in themselves as artifacts than any other. They require players to manipulate them in order to come to life. It is possibly more accurate to call what emerges during play “the game” than the mere equipment, the pieces and software, of which the game supposedly consists. And, amongst game genres, this is nowhere more true than in tabletop games that take social interaction between players as their primary material. In a real way, these games do not exist outside of the brief hours when people bring them to life by playing them.

Therefore game design makes continual playtesting a central part of its process. However, unlike a traditional scientific or engineering process, game design cannot simply propose a hypothesis, create an experiment to test it, record results, and then evaluate them to validate or reject the hypothesis. Play and fun are too subjective and contingent as aesthetic qualities to easily yield to reliable objective measure and games are systems of sufficient complexity and integration that controlled comparative changes are nearly impossible.

Some game designers have attempted to overcome these challenges through the vigorous application of methods from experimental psychology,[1] but most game designers have reconciled themselves to the insolubility of their task. In their landmark paper, “Build It to Understand It: Ludology Meets Narratology in Game Design Space,”[2] Michael Mateas and Andrew Stern argue that game design is a “wicked problem” in the sense first coined by Rittel and Weber.[3] Wicked problems have no definitive problem statement, they lack a stopping rule, each variation of the problem is unique, and there is no ultimate test by which to judge a solution. In response to the wicked problem of game design, Mateas and Stern propose “a simultaneous process of research and artmaking” as a probe to improve understanding of open questions in the field. They suggest a process including the creation of new games along with careful playtesting to study the effects those games have on players. They specifically identify this approach as being most necessary when exploring new areas of game design (in their

[1] Mike Ambinder, Valve’s Approach to Playtesting. Game Developer’s Conference. 2009.

[2] Michael Mateas and Andrew Stern, “Build It to Understand It: Ludology Meets Narratology in Game Design Space”. DiGRA. 2005.

[3] H. Rittel and M. Webber, “Dilemmas in a General Theory of Planning”, in Policy Sciences 4, Elsevier Scientific Publishing, Amsterdam, pp. 155-159, 1973.

case, interactive dramas): “As a wicked problem, only by actually trying to build an interactive drama could we have ever identified this design region.”

This is exactly the methodology I have adopted for this thesis. As a nascent design space with few examples to study, the only way I could explore the design region of hybrid digital-physical tabletop games was to create such a game, playtest it extensively, and document the conceptual territory that I discovered in this new region. That is exactly what I have tried to present over the previous five chapters with both the description of Sneak and the four questions of my design framework.

As mentioned, Sneak was regularly playtested throughout its development. In addition to the usual reasons for playtesting a game, Sneak’s playtests acted as the raw material for the design discussion presented throughout this thesis. While these playtests are not a scientific evaluation of Sneak, they acted as a vehicle to improve the game with each iteration, and hopefully they also provide the reader with a glimpse into how players react to hybrid tabletop games and the kinds of specific problems those types of games pose for designers.

There is one additional axis for evaluation that is theoretically possible with this thesis. Beyond Sneak itself, this thesis’s major product is the design framework presented in these pages. Now that I have articulated this framework, it can be shared with other practitioners working in the space of hybrid tabletop games and evaluated by the degree of usefulness it provides to them. Unfortunately, such an evaluation was not possible until the initial project, presented here, of discovering and articulating it in the first place was completed.

That next project begins now.

Bibliography

Ambinder, “Valve’s Approach to Playtesting”. Game Developer’s Conference. 2009.

Bass, “A New Product Growth Model for Consumer Durables”. Management Science 15(5) 215-227. 1969

Bakker, Vorstenbosch, van den Hoven, Hollemans, and Bergman. 2007. “Weathergods: tangible interaction in a digital tabletop game”. Proceedings of the 1st international conference on Tangible and embedded interaction (TEI '07).

Brooks, “Spatial and verbal components of the act of recall”, Canadian Journal of Psychology. Vol 22(5), 1968.

Brand, Stewart, “Spacewar: Fanatic Life and Symbolic Death Among the Computer Bums”. Rolling Stone, December 1972.

Chaboissier and Vernier. 2009. “RealTimeChess: a real-time strategy and multiplayer game for tabletop displays”. Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS ’09).

Hunicke, LeBlanc, and Zubek. “MDA: A Formal Approach to Game Design and Game Research.” Proceedings of the Challenges in Games AI Workshop, Nineteenth National Conference of Artificial Intelligence. 2004

Ishii, Wisneski, Orbanes, Chun, and Paradiso. 1999. “PingPongPlus: design of an athletic-tangible interface for computer-supported cooperative play”. Proceedings of the SIGCHI conference on Human Factors in Computing Systems (CHI '99).

Kahneman, Daniel, *Thinking Fast and Slow*. Farrar, Straus and Giroux. 2011. p. 212

Magerkurth, Carsten, Memisoglu, Maral, Engelke, Timo and Streitz, Norbert. 2004. Towards the next generation of tabletop gaming experiences. In Proceedings of Graphics Interface 2004 (GI '04).

Mandryk and Maranan. 2002. “False prophets: exploring hybrid board/video games”. CHI '02 Extended Abstracts on Human Factors in Computing Systems (CHI EA ’02).

Markoff, John, “A Long Time Ago, in a Lab Far Away…”. The New York Times, February 22, 2002.

Martins, Reis, and Teófilo, “Poker Vision: Playing Cards and Chips Identification based on Image Processing”. Pattern Recognition and Image Analysis Conference. 2011

Mateas and Stern, “Build It to Understand It: Ludology Meets Narratology in Game Design Space”. DiGRA. 2005.

Misra, Cheng, Genevie, and Yuan, “The iPhone Effect: The Quality of In-Person Social Interactions in the Presence of Mobile Devices”. Environment and Behavior. July 1, 2014.

Ohyagi and Satou. “Racing Game Program and Video Game Device”. Nintendo Co., Ltd., assignee. Patent US 7278913 B2. Oct.-Nov. 2007. Print.

Rittel and Webber, “Dilemmas in a General Theory of Planning”, in Policy Sciences 4, Elsevier Scientific Publishing, Amsterdam, pp. 155-159, 1973.

Slavin, Kevin, “Debunking Luck”, Pop Tech 2013. https://vimeo.com/78829799

Swink, Steve, *Game Feel: A Game Designer's Guide to Virtual Sensation*. Morgan Kaufmann. 2008.

van Loenen, Evert, et al. "Entertaible: a solution for social gaming experiences." Tangible Play workshop, IUI Conference. 2007.

Wallace, Pape, Chang, McClelland, Graham, Scott, and Hancock. 2012. “Exploring automation in digital tabletop board game”. Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work Companion (CSCW ’12).

Ludography

Alchemists, Matúš Kotry. Arclight. 2014.

Animal Planet: Wildlands, Nukotoys. 2014.

Assassin's Creed: Brotherhood. Patrick Plourde. Ubisoft. 2010.

Battlestar Galactica: Pegasus Expansion, Daniel Clark, Corey Konieczka and Tim Uren. Fantasy Flight. 2009.

Betrayal at House on the Hill, Rob Daviau, Bruce Glassco, Bill McQuillan, Mike Selinker, and Teeuwynn Woodruff. Avalon Hill. 2004.

Call of Duty series. Infinity Ward, Treyarch, and Sledgehammer Games. Activision. 2003-present.

Candy Land, Eleanor Abbott. Hasbro. 1949.

Carcassonne, Klaus-Jürgen Wrede. Hans im Glück Verlags-GmbH. 2000.

Conspiracy, Eric Solomon. AMIGO Spiel and Freizeit GmbH. 1973.

Counter-Strike: Global Offensive, designer unknown. Valve Corporation. 2012.

Crusader Kings II, Henrik Fåhraeus. Paradox Interactive. 2012.

Dungeons & Dragons, Gary Gygax and Dave Arneson. TSR. 1974.

Dwarf Fortress, Tarn Adams and Zach Adams. Bay 12 Games. 2002.

Escape from the Aliens in Outer Space, Mario Porpora, Pietro Righi Riva, Luca Francesco Rossi, Nicolò Tedeschi. Cranio Creations. 2010.

A Few Acres of Snow, Martin Wallace. Asmodee. 2011.

Far Cry 2, Clint Hocking. Ubisoft. 2008.

Fire in the East, John Astell, Rich Banner, and Frank Chadwick. GDW Games. 1984.

Goldeneye 007. Martin Hollis, Rare. Nintendo. 1997.

The Legend of Zelda, Shigeru Miyamoto, Takashi Tezuka, Eiji Aonuma. Nintendo. 1986.

Monopoly, Elizabeth Magie and Charles Darrow. Parker Brothers. 1933.

Pikmin 2. Shigefumi Hino and Masamichi Abe. Nintendo. 2004.

Risk, Albert Lamorisse. Parker Brothers. 1957.

Rogue, Michael Toy, Glenn Wichman, Ken Arnold, Jon Lane. Various publishers. 1980.

Scotland Yard, Manfred Burggraf, Dorothy Garrels, Wolf Hoermann, Fritz Ifland, Michael Schacht, Werner Scheerer, Werner Schlegel. Bonaparte. 1983.

The Settles of Catan, Klaus Teuber. 999 Games. 1995.

Sim City, Will Wrght. Maxis. 1987.

Sorry!, Paul T. Haskell, Jr. and William Henry Storey. Basic Fun, Inc. 1929.

Spaceteam, Henry Smith. Sleeping Beast Games. 2012.

SpyParty. Chris Hecker. Currently under development.

They've Invaded Pleasantville, Michael Price. TSR. 1981.

Werewolf/Mafia, Dmitry Davidoff. 1986.

World of Yo-Ho, Volumique. 2014.

XCOM: The Board Game, Eric Lang. Fantasy Flight. 2015.

Appendix A - Selected Playtesting Notes

Notes from a playtesting session held on March 9, 2015[1]

First playtest with random strangers recruited from the MIT student body (by my UROP, Ari). And it was actually surprisingly successful!

In the first game, about eight turns in, one of the players suddenly got really convinced he knew which character the other player was. “Oh, you’re orange!” he announced after a movement phase. He proceeded to move his character onto a gun and kill orange. But he was wrong. His opponent wasn’t orange. His opponent was white and he had the plans and was a few squares away from exiting. He proceeded onto the exit winning the game before the first player could get anywhere near a second gun.

In the debrief afterwards, I learned a lot of interesting things from the players. For example they had some sense of the NPC movement patterns, but had also not quite figured it out yet. One of them thought, for example, that the NPCs always moved in straight or diagonal lines, like the Queen in chess.

The first game lasted about 20 minutes and the conversation afterwards at least that long. The playtest had started kind of late so I was starting to thank them and let them go when they asked if they could play again. Of course, I obliged.

The second game went very differently. Seemingly spooked by the gun shot from the first game, both players were much more circumspect about how much information they gave away about themselves and how much they felt like they could deduce about the other player’s identity. This lead to a much slower, more conservative game, particularly after the NPCs started reaching their destinations and moving less about 5 turns in. One player went to ground, holding for a large number of turns to not stand out. The other player assumed that an NPC would be the only one to move and therefore decided to move around a lot to look like an NPC!

After a few turns, that second player had a lot of information, but hadn’t yet found the plans. As more NPCs start to move, the first player (now far behind on information) moved for the first time in awhile and pretty quickly stumbled onto the plans. At that point it seemed like he had a

[1] Originally posted to the Sneak development blog here: http://cardboardmagic.tumblr.com/post/113283309789/first-playtest-with-random-strangers-recruited

clear shot to get to the exit and win. However, he was moving away from the exit at the time he got the plans so he kept going for a turn or two in order to maintain NPC realism.

That path lead him to collide with the second player. I'd told them at the start that if they ended up on the same square with another character who had the plans, but didn't give them over then that was definitely the other player. But the second player didn't react after that conversation. I realized he'd probably missed the information about the plans in his very long list of info. So I slipped him a note reminding him of that rule. On the next turn he checked his info, realized what had happened, and a few turns later he was on a gun and had killed the first player, winning the game.

In the debrief conversation afterwards, I confirmed that player two hadn't seen the information about the plans. He said he felt pressure not to look at that screen for too long so as to not give away to the other player how much information he had. I've known for awhile that info view would need to improve and I came out of this playtest with some specific ideas for how to acheive that.

At this point, I asked them again about how they thought the NPCs moved. This time they had it almost completely right. And they both complained that this time they'd felt constrained by the NPC movement in how they chose their own moves. So more variation in NPC movement (both between NPCs and even within NPCs across time) seems like a good idea now.

In many ways the results of this playtest were lucky and specific to some of the eccentricities of these players. Being extremely careful engineers, they never once came close to screwing up the process of moving the pieces on the board and things never got out of sync between the computer and the board (it looks like I've now truly thoroughly squashed the bugs that were causing that on the game's side). That is definitely not something I could expect of your typical distracted player. Some definite design time needs to go into something like a "checksum" for board positions, a way for the game to double check that everything is where it should be.

The players also expressed some concern about the fairness of the game due to the random nature of where the plans are, who runs into them, etc. But, without my even prompting, they discussed how the different outcomes of their two games showed that while any one game might give a particular player the plans, that wasn't even necessarily a good thing to get, and the random elements meant that neither of them could really think of a dominant strategy that would always work. Classic roguelike dynamics.

All-in-all a very encouraging playtest.

Notes from a playtesting session held on March 22, 2015

Did another playtest last Friday. Ran two full games with two different sets of people.

In both games, players won by figuring out who their opponent was and shoot them rather than by finding the plans.

In discussing our observations, Ari and I came up with ideas for a couple of new mechanics we were excited about:

- Poisoning When a player is on the same square as another character they can choose to poison them, causing that character do die after 1-5 turns (at their player's choice). This gives players a way to take a chance and kill a character without giving away their identity. But it is still highly risky as they have to actually occupy the same square as their target which means they might have to chase them around and will definitely share their information. This would be a one-time action per game for each player.

- Informants Once per game, a player can turn another character into their informant. The player has to be on the same square as the other character to make them their informant. After that point, each piece of information the other character learns also is transferred to the player. If a player has a dialogue with a character that is an informant for another player, they discover that this is the case and are shown a list of every character the informant has talked with (one of whom must be the player).

- Meetings A certain random moments, the game will decide that a subset of NPCs are going to have a meeting. They will all be assigned the same square as a destination and they will all head there and wait until all meeting participants have arrived. This will rapidly increase the spread of information. The idea is to use this as a way to programmatically accelerate the game when players aren't gather enough information on their own. Meetings will be triggered under circumstances where players don't have enough information after a certain number of turns have passed. This will give the NPCs interesting behavior as well as taking advantage of the unique affordances of a digital game in order to create more interesting game dynamics.

Notes from a Playtest Session Held on March 23, 2015

Another playtest yesterday with the generated board and the new poisoning mechanic. This was a fascinating one because one of the players (player 2) took a more analytical approach than I've seen any player attempt before. At the start of the game, she aksed if it was ok if she took notes. I said it was fine as long as her opponent didn't object. On each turn, she wrote down her observations and conclusions about which character her opponent was controlling (see the image above and the flip side of the paper here). This significantly slowed down the pace of the game making this the longest single playthrough yet (about 75 mins).

Fascinatingy, player 2 actually reached incorrect conclusions about their opponents identity using this method. In fact, unlike every other previous playtest participant, none of the candidates to which she narrowed the field were actually the other player (every other player I've seen play the game can eventually name two or three candidates for who the other player was and the right answer has always been included in their candidates).

When the game started player 2 ended up in a location closer to more other characters than player 1 and gained more information. But because of the slower pace of play, the starting position of the players, and the board layout, the "first act" of the game, during which no one knows the location of the plans, lasted longer than usual. Too long. Players become somewhat frustrated during this phase. But, eventually, all the characters converged in the bottom right of the board, an area near the exit, a gun, and the one door on the board connecting the indoor space to the outdoor space.

Player 2 seemed to be in a dominant position. But then after about 10 rounds, when all the characters in the game converged on the same general area (see this photo; yellow is Player 2, black is player 1, and green has the plans) player 1 learned a lot very quickly. Player 2 discovered who had the plans and it happened to be her leading candidate for player 1's character. She thus despaired of winning. She decided to poison a character on her square out of desperation. When that character died it spooked player 1. Player 1 had narrowed player 2's identity down to 2 characters. They decided to move to a gun and shoot. They guessed right and won the game. They said that if the poisoning hadn't happened they would have poisoned instead and the character sharing their square was their other (incorrect) candidate for player 2.

Transcriptions of my notes appear below:

* Player 2 thought the outdoor squares on the blueprint were not accesssible to characters. Suggested using grass instead of lines to indicate outdoor squares.

* Board setup took around 3 minutes, completing almost exactly when the opening theme music ended.

* Players missed the player number in the pass interface. Had to look closely to understand that screen. They also didn't remember their own player number and were a bit confused when it was referred to in that phase.

* After the first turn, the move instructions sound didn't play back This was a bug in the InstructionPlayer. I subbed in and read the instructions since it's my voice in the app anyway.

* Player 2 is keeping extensive notes on which characters she thinks might be human-controlled. (give an interface for this?)

* Player 2 had a theory that green was the opponent, when they learned green had the plans they despaired at winning because green was closer to the exit. But they were actually wrong. Green was an NPC.

* Player 2's theory came from the way green moved towards the exit early.

* Player 2 poisoned one the characters on their square

* Player 2 realized that green wasn't an player since they went past the exit

* Player 1 just missed getting the plans by one square, passing green on the move.

* Player 1 just got a ton of information about who has the plans

* Player 1 was shocked by the poisoning death

* Player 1 asked if guns can be fired more than once

* Player 1 shot and killed Player 2, winning the game

* Player 1 explained that they had narrowed the suspects down to two for the other player. One correct guess and one additional character. If player 2 hadn't poisoned gray, player 1 would have used their poison on their other candidate instead of shooting. But the poisoning spooked them so they decided to fire.

* Player 1 said that after not getting much information early on due to where their character started they decided to closely watch where Player 2 looked at the board in order to try to get a sense of who they might be. "I'll just watch where she's looking."

* Player 2 was frustrated that they couldn't deduce the identity of the other player. In going over her logic, she made some leaps and assumptions beyond the information she actually had (mainly based on assumptions about how an NPC would move) which were what lead to her false conclusions.