AI Design Lessons for Social Modeling at Scale

Ethan Robison,[1] **Matthew Viglione,**[2] **Robert Zubek,**[2] **Ian Horswill**[3]

[1] Naughty Dog
[2] SomaSim
[3] Northwestern University
ethan@ethanrobison.com, robert@somasim.com, matt@somasim.com, ian@northwestern.edu

Abstract

City of Gangsters is a commercial strategy game in which the player is principally challenged by navigating a complex and large-scale social network, in which every action resonates through the network and holds associated risks and rewards. Our procedurally-generated city results in a large social network with randomized and non-prescriptive configurations. Gameplay is oriented around social reciprocity and engaging in the same via an understanding of a concise set of social norms. Addressing these problems in a video game required a close unity between AI design and game design.

We present four key AI design and implementation lessons learned in developing and shipping this game: the paramount need to make social actions and their consequences legible, the need for reversible actions, the need for modeled social norms to comprise a succinct set, and the need for individuals to be fungible with one another vis-a-vis social actions. We conclude with a description of the design affordances of this approach.

Introduction

City of Gangsters (abbreviated: *CoG*) is a forthcoming criminal syndicate simulator tycoon game set in the 1920s Prohibition-era United States, made by SomaSim and collaborators. Its gameplay centers principally on the maintaining of large distribution networks of illegally procured (or produced) alcohol, the sale of which is profitable by virtue of its black-market status. In addition to an economy of money and goods, *CoG* is a game built on an economy of social exchange. Whom you know, and how they feel about you, is a key determining factor in short- and long-term success. In designing the game, we sought to create a city which felt vibrant and lived-in—every street corner holding a potentially valuable or challenging opportunity or frustration. We sought as well to provide the player with a handful of principles as to how best to play the game, but not to limit them to any specific approach on any given playthrough. The city, its residents, their relationships; and their proclivities, idiosyncracies, and business opportunities; are all procedurally-generated. This heavily informed the design of the game, as it meant we could not rely on hand-tuned starting configurations; rather, we had to provide players with general-
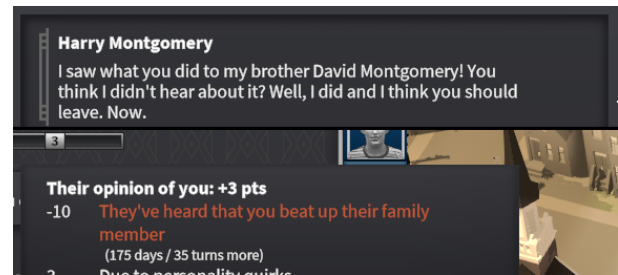
Figure 1: NPCs in *City of Gangsters* pay attention to the player's actions, especially those which impact their friends and family. Top: An NPC comments on the player's mistreatment of their brother. Bottom: The player may inspect their relationship with this NPC to learn more.

purpose knowledge for approaching any particular situation. This means providing all the information they might need to turn any situation profitable.

In addition to feeling vibrant, we made our procedurally-generated cities feel vast by populating them with over 1000 individually simulated non-player characters (NPCs), each of whose family and personal history and relationships are tracked, and used extensively during gameplay. As a result, the social economy (which itself supports a material economy) is a complex web of relationships, grudges, favors, and quid-pro-quo exchanges where everyone—hopefully—walks away a little richer. Each relationship holds a valence (bad to neutral to good) which is impacted by actions that agents (human- or AI-controlled) perform. Those actions have consequences on other relationships, as well: the game actively simulates the propagation of social consequences beyond just the agent unto whom an action is performed. Players must be careful, lest careless mistreatment of a seemingly unimportant individual ruin their carefully laid plans elsewhere in the city.

On the one hand, we have one set of challenges: the social economy, the tracking of relationships, and the presenting of this information to the player. On the other hand, we have a wholly-different set: 1200 agents, an "open city" with no set progression path; and highly variable, procedurally-generated layouts. The former set requires providing the player with substantial and specific information for making

meaningfully tactical and strategic decisions. The latter set requires that this information be tracked and surfaced in a general way, divorced of the particulars of any city.

In this paper, we discuss the design process that enabled us to meet this onerous amalgamation of desiderata. We present lessons applicable to projects whose design goals center around social simulation with a substantial number of agents and meaningful relationships with and among them; where the player can make meaningful predictions about what is going to happen when they perform a social action; and this happens against the backdrop of a run-specific and unpredictable game configuration. Each lesson is presented as a conclusion that we reached during the process of developing the game. For developers with similar design constraints and desiderata, following them will help in achieving unity between the design of the game and the design of the AI. We present the following four conclusions:

- Legibility of information: The pre- and post-conditions of a social action must be clear to the player.

- Reversibility of Actions: For any given action, the player must be able to—through some means—work to return the social network to the state before that action.

- Succinctness of Social Norms: The modeled social rules must comprise a succinct and genre-appropriate collection of social maxims.

- Fungibility of Individuals: For a given action in a given context with a given result, applying that same action in a different but similar context must have a similar result.

We discuss each conclusion at length, grounded in examples from *CoG*, but with large-scale social simulation games outside of the tycoon sub-genre in mind. We discuss a design framework for games involving such scale which also use social inference and simulation as core elements of their design, dubbing this intersection "social economy" games.

Delightful Complications in *City of Gangsters*

Systems-heavy simulation games are known for providing players with an endless supply of surprising experiences, what might be called "delightful complications". It is much the same in social economy games. To provide a concrete example from *CoG*, and to motivate the creation of such intricate and large-scale social simulation games more generally, consider the following gameplay example:

The player has established a substantial bootlegging operation. This involves purchasing liquor from multiple sources, transporting it to a central location, and operating a speakeasy from that location. Of consideration are the amount of suspicion drawn from local law enforcement officials, maintaining money in the pockets of the delivery drivers, and ensuring that excess product from other parts of the criminal enterprise are appropriately relocated.

Then, in an effort to expand their territory, the player decides to oust a nearby goon from an auspiciously-placed corner. Simple enough for the player to intimidate them with the threat of violence, and the goon leaves the area, allowing the player to swoop in take over.

After a few turns, however, the player learns that their speakeasy no longer receives enough product to turn a profit anymore. To their surprise, three of their suppliers were relatives of the victim of their earlier violence, and all of them have decided that the player is no good, and they refuse to sell to them henceforth. The player is left high and dry, and will have to work hard to repair the damage done to their crucial business relationships.

This example highlights the consequences of social inference on seemingly unrelated aspects of gameplay. Combat and the resource economy are linked together through the social simulation, providing delightful complications to the player's plans. Indeed, the material economy (buying, selling, establishing trade routes, processing raw ingredients, etc.) and progression mechanics (unlocking new resources, learning new skills, improving physical structures, etc.) are linked through the social simulation. Nearly *every* aspect of gameplay is mediated through the social economy.

Related Work

City of Gangsters brings to mind several games, commercial and academic, with analogous mechanical or technical trappings. First are games involving large numbers of agents with heavy simulation elements. *Dwarf Fortress* (Adams and Adams 2006) involves the creation and forward simulation of huge numbers (hundreds or even thousands) of agents for the purposes of generating NPCs used in gameplay. However, moment-to-moment gameplay within *Dwarf Fortress* is not centered around the social consequences of player choice; NPCs hold opinions about happenings in the world, and these opinions may change their behavior in consequential ways, but the player is rarely performing an action with a social result specifically in mind. Similarly, *Bad News* (Samuel et al. 2016), involves a player navigating a social network comprised of numerous agents, each of which is generated and forward-simulated to create a rich and interesting social graph. However, the social simulation in *Bad News* stops before gameplay begins; the player is navigating a relatively static social graph and exploring its layout rather than using social inference mechanics to change the graph's structure.

Second are games with heavy social inference mechanics, where a mastery of the designers' choice of social norms is critical to success. *Prom Week* (McCoy et al. 2012), built using the work done on *CiF* (McCoy et al. 2011), concerns itself with the rich social simulation of a small number of agents, tracking the state of each relationship and changing that state according to player decisions. *Versu* (Evans and Short 2013) contains similar mechanics, with player choices being "judged" by third parties, and with social consequences mediating later gameplay affordances. Additionally, though it is also concerned with drama management and story beat arrangement, *Façade* (Mateas and Stern 2003) is a game with heavy relationship simulation. All three games involve a model of social relationships impacted by player choices. The important distinction with *CoG*, however, is the number of agents simulated. *Prom Week* and *Versu* gameplay sessions involve on the order of 10s of agents and *Façade* involves precisely 2. By contrast, *CoG*

simulates 1200 agents in a typical gameplay session. The sheer number of agents involved implies a different set of design and UI considerations.

Four Lessons

Having described the game's design and gameplay features, we return to the AI design lessons learned during this project.

Legibility of Information

One cannot get anything done in the Prohibition-era fiction of *CoG* without a connection. "You gotta know a guy," was a key design principle of every aspect of the game, and knowing which guys you did—and did not—have a good relationship with was as crucial a part of the design of the user interface as knowing how many barrels of whiskey were in your storeroom. For a complicated social simulation to work as the backbone for a game, it needs to be legible to the player.

First and foremost, the inner workings of the social simulation need to be accessible to the player. The first way that *CoG* presents information about a given relationship is through an aggregate number, a relationship level. A relationship which is a floating-point value typically in the range from -50 to +50. From the player's perspective the relationship level is used ubiquitously as a parameter in their moment-to-moment interactions. Everything from how politely an NPC greets the player, to whether or not the NPC will sell illicit product to them, is gated behind the relationship level (in addition to other mechanics not germane to this discussion). If things get bad enough, the NPC may stop selling to the player entirely, or, if they are members of a hostile crew, they may attack the player. On the other hand, if the player gets their relationship level with an NPC high enough, they garner favors, which can be used as social captial to coax someone to go above and beyond the usual degree of obligation.

There is a resulting elegance in having a single number represent each relationship (and in having a single "variety" of relationship): the challenge of *CoG* comes principally from the sheer volume of connections the player must maintain, so reasoning about any individual connection must be straightforward. Several numbers, or special "flavors" of relationship with their own rules, would be unnecessary cognitive burdens.

Furthermore, having something as complex as a relationship, with its many ups and downs and complicated characteristics, boil down to a single number holds advantages: it provides the player with a dead-simple means of knowing how things are going between them and an NPC. This number correlates with the presence/absence of other mechanical affordances or flavor elements, but the player always has an unambiguous answer to the question of how they should regard a relationship's status.

The second task in providing a meaningful interface into the numbers underpinning a simulation is allowing the player to ask *why* a given number is what it is. It is all well and good to see that something is in a certain state, but unless the player is afforded good access to the history that
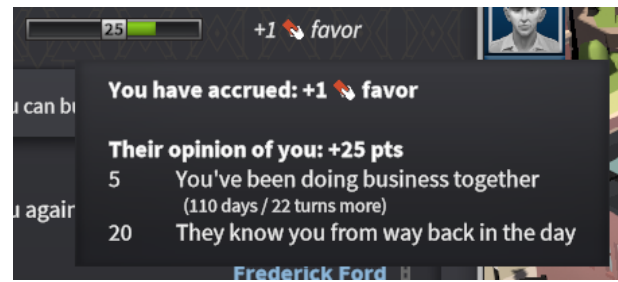

Figure 2: UI for displaying detailed information about a relationship level
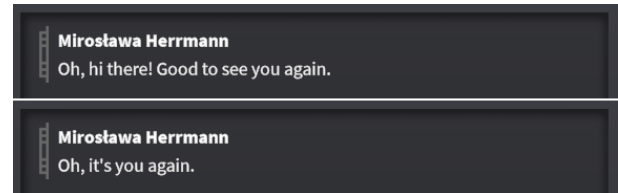

Figure 3: Top: Greeting with a high relationship level
Bottom: Greeting with a low relationship level

led to the present moment, they will be unable to willfully recreate (or avoid) similar situations elsewhere. In *CoG*, the relationship level is a single number, but the player may dig deeper in the UI to find a list of all of the in-game events that contibuted to the relationship's current status. Figure 2 shows the UI element which reports a current relationship's level, and the explanations behind it.

This hierarchical presentation of a social simulation's numbers is important, and both layers are necessary. Replace the top-level monolithic number with the list of effects, and the player is left estimating complicated arithmetic in their head in an effort to come to qualitative conclusions about the relationship. Show the list of effects alongside the number at all times, and a quick gloss of a relationship's status is made difficult by the presence of too much extra information. On the other hand, remove the explanatory list, and contextualizing a given relationship's relationship level becomes an impossible task: so many different combinations of numbers add up to any result that deduction of the underlying state is a mathematical impossibility. This is further complicated by the fact that some of the relationship adjustment effects in *CoG* expire after a number of in-game turns, meaning that the list is often on the cusp of changing. Providing this explicit list affords the player the opportunity to know when and in what way a relationship is liable to change.

This is related to the challenge we sought to provide with *CoG*. There are simply so many agents in the social network that it is effortful enough *looking* at all of them and reading their various UI elements. Ambiguity of any sort did not become the experience we sought to provide as an early example of social simulation game at scale.

A final consideration concerns a different form of surfacing information to the player. Interactions between the player and other agents in *CoG* are fictionalized as conver-

sations, and the UI presents them as such. Figure 3 shows an example of the player being greeted differently depending on their relationship level with a given NPC.

In this way, even the tone and diction in a conversation with an NPC serves to subtly inform the player about the state of their relationship with a given NPC. NPCs may also bring up specific social actions (positive or negative) that they have heard about the player doing. For example, if the player has helped a relative of theirs out by completing a quest, they may make mention of it during a conversation. Figure 1 shows an NPC mentioning a social action on the part of the player during a conversation.

Reversibility of Actions

It is important that a given social state have multiple routes of access, so to speak. If the only way to earn the trust of a shopkeeper is to purchase their goods, but they refuse to sell to the player because of a prior infraction, then the player is soft-locked out of a good relationship with them. They are not technically prohibited from having the good relationship that they seek, but there is no means of them achieving it, and therefore it is out of reach.

Soft-locking. Of concern for game designers of many genres, but in particular to games in the simulation genre, is "soft-locking." Soft-locking consists of a game state which is not considered (by the game itself) to be a failure state, but from which only failure states are reachable, regardless of player action. In other words, the player has already lost, they just have not yet realized that fact. Soft-locking is endemic of some older examples of the point-and-click genre, where the player has unwittingly forgotten or omitted a required step, and is able to progress well past the point of returning to a place where they might take that step.

Complex social simulation games are particularly vulnerable to what may be called "death spirals." Death spirals are a form of soft-locking in which the game has not technically ended, but the social state is such that there is no route towards a positive outcome. The player is left waiting for everyone in their social aquarium to die out (literally or figuratively). The worst case comes when the player is unaware that a death spiral has begun, and is working diligently towards a positive resolution while no such result is remotely possible.

With soft-locking and death spirals in mind, it is crucial to afford players multiple means of achieving a given social goal. The first way that *CoG* solves this problem is through "putting in a good word." An agent can offer to improve the relationship between the player and a third party, in exchange for some social capital (itself acquired through any variety of means). Figure 4 shows the result "putting in a good word" in a relationship's UI.

Redundant social strategies. The design of *CoG* is such that the player usually has a plethora of available social options. Figure 5 shows the in-game display of all of the player's direct relationships. The player can improve their relationship with a connection, then use that improved relationship as a springboard to improve the relationship they have with a third party, and so on. Each of these connections
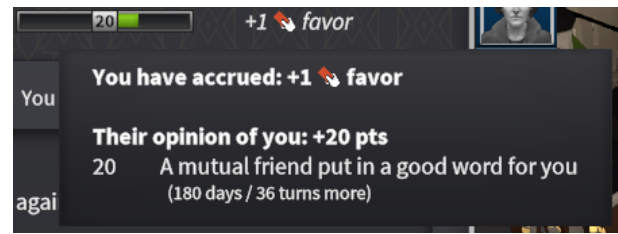

Figure 4: The effect of a boost from one NPC to another

Figure 5: The connections UI

are themselves connected to other agents, and a player can (and does) leverage these indirect connections as a means to an end during a play session.

This flexibility in addressing a given problem provides a degree of undoability. If the player makes a mistake or a miscalculation, they have several different routes to take in undoing that error. This reversibility is necessary in social simulation games with heavy systemic elements. If interactions were hand-authored, of course, a developer could make soft-locked situations impossible script-side and be done with it. We have limited such capacity in social economy games with heavy procedual elements, and must therefore be extra vigilant for limitations in player options.

Succinctness of Social Norms

The number of different social consequences that the player has to keep in mind needs to be limited. In a social economy video game, tracking a huge number of potential social variables is burdensome. *CoG* meets this need by choosing a tightly-focused set of social norms. Each norm is:

- Genre appropriate. We chose norms that make sense in a gangster game.
- Easy to describe. Each norm can be alluded to with a short phrase.
- Well-known in Western culture. Norms are not, for example, complex and subtle psychological patterns unknown to the average player.

Finally the norms are all related to one another through one of the game's fundamental design ideas: "You gotta

know a guy." The world of *CoG* is quid-pro-quo top to bottom, and the set of social norms we implemented matches this:

- "You scratch my back, I scratch yours." Help somebody, and they and their friends will think highly of you (and likely reward you for it).
- "An eye for an eye." Hurt somebody and they and their friends will think badly of you for it (and likely try to enact revenge).
- "The enemy of my enemy is my friend." Hurt somebody, and their *enemies* will think highly of you for it (and likely reward you for it).

The social norms chosen in *CoG* serves as important flavor elements in a dynamic social graph. When a player sees an NPC mention an action of theirs (see Figure 1) in conversation, they know that their every move is being watched. But what gets watched, and how it surfaces later, is a subtle reinforcement of the aesthetic choices of a social economy game's designers.

Fungibility of Individuals

The fiction of *CoG*—criminal syndicate operation in a world where the sale alcohol is illegal—afforded several guiding principles for the social norms that we chose to include in gameplay. The phrase that came up frequently during design was, "you gotta know a guy." An important idea related to black-market deals is that they are facilitated by existing connections; there is no place where an individual new to a given illicit trade can just walk in and set up shop, and wait for customers to arrive. Trust and credibility are worth a lot of social capital.

The social network and the various means of forming and strengthening relationships address the challenge of "knowing a guy." Of further concern for a player (and a designer) is what happens once a relationship has been formed. If a player has a relationship in a given state, how do they make efforts to change that relationship to some other state? We have discussed the important aspects of surfacing and user interface elements such that a player may know the state of a relationship. This provides them both with a means of sizing up a given relationship: how far off is it from where they would like it to be? On top of that, the explicit listing of relationship history elements provides them with a special kind of knowledge: a relationship in a given state can provide a template for getting a different relationship into that same state.

Figure 6 shows the list of buffs applying to two relationships. Relationship A contains two buffs (one for generosity on the part of the player, and one for having given the player a loan) and has a resulting relationship level of +10. Relationship B contains one buff (one for generosity)—only one of the two that A has. Fungibility of individuals implies that the player can use Relationship A as a guide for getting Relationship B to a particular state. They can do this by taking the action implied by the buff (generosity in business dealings) listed in Relationship A but not listed on Relationship B. They do this knowing that this action will consequently apply the same buff to Relationship B.
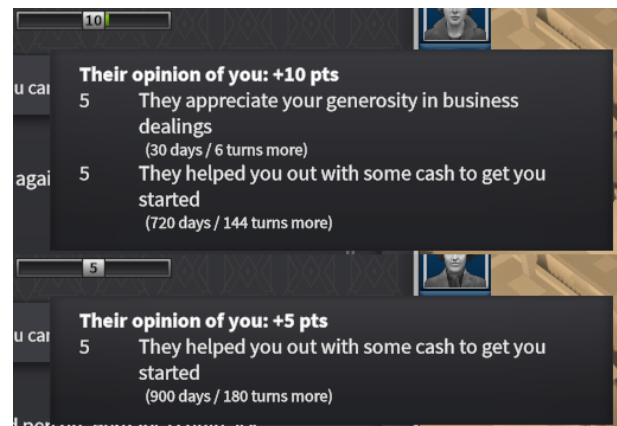

Figure 6: Examples of two relationships with similar sets of buffs. The player can use the information from one relationship as a guide to improving another relationship.

This explanation may seem straightfoward—after all, if you do the same thing to two different people, they are likely to respond in similar ways—but it bears emphasizing that this is an important design principle to follow in a social economy game. The alternative option of making each relationship unique, and respond uniquely to particular actions, dramatically increases the burden for players. To put a finer point on it, keeping individuals fungible counter-balances the sheer number agents in the simulation. Players have a finite capacity for tracking different aspects of the game, and reducing complexity by providing uniformity makes a social economy game mentally tractable to play.

Discussion

Social economy games are distinguished by their scale and by inclusion of social norms as the cornerstone of their design. In addition to the design lessons learned in creating the AI for *CoG*, we arrived at several conclusions for consideration by would-be social economy game designers.

First, that quality over quantity is crucial for social norms. A plethora of rules is confusing, for one. A handful of rules, deliberately crafted, obsessively surfaced, and thoughtfully explained, offers a wide range of design depth. Though we offer no hard limit, in the current world of social simulation, having more than twenty rules strikes us as too many. We emphasize the richness we were able to achieve with only a handful of rules. Perhaps a second generation of social economy games may be more approachable to players, and consequently may include dozens of rules. Perhaps we may see social economy games, analogous to hyper-complex military simulation games, involving hundreds of social norms and thousands of agents.

Second, social propagation should be fairly local to the actor and the object of their action. If A performs an action to B, then the relationships of B to C and C to A should be considered. The relationships from C to D, and from D to A should not be (assuming no relationship from B to D). In other words, social effects should be "second-order," but

they should not traverse the transitive closure of relationships from the object of a social action onward. The player is going to have a rich enough challenge managing their social network as is (without worrying about all 1200 agents at once).

Third, the nodes at either end of a relationship graph edge should be fungible, or mostly fungible. Players should be able to regard a relationship between A and B and its history elements, and consider the relationship between A and C and its history elements, and make plans based off of the difference between those sets of history elements. They should not have to consider the *kind* of relationship between A and B and A and C. There are exceptions to this rule, and indeed *CoG* makes such exceptions for certain social history elements, but only in terms of the intensity of change in relationship, not in terms of the broad strokes.

Future Work

City of Gangsters is a demonstration of social simulation at scale. Of particular interest to the authors are separate genres and fictions, unrelated to the Prohibition-era United States, and involving social entities other than a web of criminal co-conspirators. SomaSim is a full-time indie games studio working towards realizing such projects.

However, SomaSim represents only one entity in the pursuit of broader industry/academic collaborations seeking to implement experimental game AI techniques (see [Redacted] for a technical discussion of such techniques). We hope that the design lessons, presented here at length, will encourage the use of social simulation in experimental academic games. Furthermore, we hope that the commercial nature of SomaSim will ipso facto suggest, to other commercial entities especially, the feasibility of using such AI architectural elements as more than mere add-ons, but rather as fundamental and foundational design considerations for future industry works.

Conclusion

With *City of Gangsters*, we demonstrate the possibility of large-scale social simulation games filled with interesting relationships among its many agents in which the player is able to make deliberate tactical and strategic social decisions vis-a-vis their various in-game relationships. With the description of the design desiderata for the social simulation elements of *CoG*, we highlight the portability and versatility of such social simulation to other game design contexts, bearing in mind the specific implications that large-scale social simulation has for various aspects of a game's design.

Future social economy games will undoubtedly require the same attention towards user interface fidelity brought to bear in developing *CoG*. Similarly, future games will require work to track and surface player actions such that they are legible. However, we see no theoretical upper limit on the number of agents involved in the simulation, nor of the potential complexity and number of of social norms simulated. *CoG* has 1200 agents and a handful of social norms, but it is interesting to consider social economy games with tens of thousands of agents and dozens (or hundreds) of distinct social norms. These norms could conceivably apply differently depending on the number of agents in an interaction or on their familiarity with one another. While we look forward to the possibility of a complex and rich set of social norms, we emphasize the need for genre relevance over realism. We also emphasize that social economy games are a relatively unexplored genre space, and that future games may need to accept players' need to be gently introduced to more and more complex social simulations. After all, the modern first-person shooter comes from a long lineage of progressively more complex designs. Social economy games will likely follow a similar progression.

City of Gangsters is a first step towards more involved social simulations at increasing scale. The richness of its gameplay and the many delightful complications throughout highlight just how much intricacy can result from even a small handful of social inference rules, when one designs the entire game with those rules and systems in mind.

Author Bios

Ethan Robison is a software engineer at Naughty Dog specializing in friendly AI behavior. He earned his master's in computer science from Northwestern University, where he researched experimental AI for video games. Prior to Naughty Dog, Ethan worked at the Chicago indie SomaSim, where he developed AI for social simulation.

Dr. Robert Zubek is a game designer and developer, and co-founder at SomaSim, an independent game studio in Chicago specializing in simulation and strategy games. Before SomaSim he built large-scale online social games at Zynga, MMO game and analytics infrastructure at Three Rings Design, and console games at Electronic Arts/Maxis. Prior to joining the industry, he specialized in artificial intelligence and robotics research. Robert holds a PhD in computer science from Northwestern University.

Matthew Vigiione is the co-founder and lead designer at SomaSim, a Chicago-based independent game studio that focuses on strategy simulation games. He is currently working on the forthcoming game City of Gangsters, a Prohibition-era crime simulation game. Previous games include the skyscraper simulation Project Highrise and the Gold Rush town builder 1849. Previously he was a communications professional and graphic designer.

Ian Horswill teaches Computer Science at Northwestern University, where he is Charles Deering McCormick Professor of Teaching Excellence. His research interests include interactive entertainment technologies, particularly interactive narrative and constraint-based procedural content generation. He received his PhD in Computer Science from the Massachusetts Institute of Technology in 1993.

References

Adams, T.; and Adams, Z. 2006. Slaves to Armok: God of Blood Chapter II: Dwarf Fortress. *PC Game). Bay* 12.

Evans, R.; and Short, E. 2013. Versu—a simulationist story-telling system. *IEEE Transactions on Computational Intelligence and AI in Games* 6(2): 113–130.

Mateas, M.; and Stern, A. 2003. Façade: An experiment in building a fully-realized interactive drama. In *Game developers conference*, volume 2, 4–8.

McCoy, J.; Treanor, M.; Samuel, B.; Reed, A. A.; Wardrip-Fruin, N.; and Mateas, M. 2012. Prom week. In *Proceedings of the International Conference on the Foundations of Digital Games*, 235–237.

McCoy, J.; Treanor, M.; Samuel, B.; Wardrip-Fruin, N.; and Mateas, M. 2011. Comme il faut: A system for authoring playable social models. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 6.

Samuel, B.; Ryan, J.; Summerville, A. J.; Mateas, M.; and Wardrip-Fruin, N. 2016. Bad News: An experiment in computationally assisted performance. In *International Conference on Interactive Digital Storytelling*, 108–120. Springer.