Proceedings, The Thirteenth AAAI Conference on
Artificial Intelligence and Interactive Digital Entertainment (AIIDE-17)

Reputation Systems for Non-Player Character Interactions Based on Player Actions

Joseph Alexander Brown†, Jooyoung Lee‡, Nikita Kraev‡
† Artificial Intelligence in Games Development Lab and ‡ Network and Information Science Lab
Innopolis University
1 University Str., Innopolis, Republic of Tatarstan, Russia 420500

Abstract

In digital games there is an emphasis on the idea of quest completion; by completing a quest the character wins resources from the giver of the quest, they also will gain a reputation among the Non-Player Characters (NPCs) for its completion. However, this reputation currently propagates across the game world in an unrealistic manner; many NPCs will know of a completion of a quest many townships over without a narrative rationale. In this paper, we examine a method for allowing NPC interactions to spread reputation in a game world from an initial witness point of a quest completion to all other NPCs. This model is examined in a series of connected graphs: size five models, small world graphs, and graphs developed from digital games. Tests show that propagation of the information is highly dependent upon easily established properties of interactions, such as the graph regularity, average degree, and diameter. Further, real game graphs demonstrate that information generated in high population hubs can be propagated faster than that generated in smaller quests from outlying areas.

Introduction

Persistent memories of Non-Player Characters (NPCs) in Role Playing Games (RPGs), primarily revolve about the events in a game. Table top methods, with a storyteller or game master (GM), have an idea of *in character knowledge*, what is known to the character themselves, and *out of character knowledge*, what is known by the player on the mechanical side. Character knowledge is able to be retained in terms of the relationships via the GM's own mind and notes. This ability to keep multiple character's and group knowledges about the interactions going on between NPC and characters has sadly not made the transition to the digital realm in a seamless fashion.

Many games implement an event occurring, such as a quest completion or killing of an important NPC, then with immediate effect all NPCs know of its occurrence. While this may happen with a narrative reasoning, such as in the case of a massive world event, commonly such events are very rare and even then the causal factor of the actions of the player will not be easily seen. Take for example stealing an item when undetected, the game may have other NPCs react to the theft without a narrative reasoning, or a mechanical reasoning. In fact the game mechanics tell the player that the action was made undetected.

Various games have attempted to produce some manner of explanation for the ability of NPCs in order to see the representational value of a player, such as a karmic balance (2K Games 2007; Bethesda Softworks 2008). Though this is rather heavy handed, as other characters can in a way read your soul, which makes it hard for a player to role play a good thief or an anti-hero caught up in the situation. Others have factional affinity in which a player by making actions towards each faction, e.g. (Bethesda Softworks 2010). This factional trope is perhaps a hold over from the table top games such as *Dungeons and Dragons* in which there is some form of parameterized morality on a spectrum between good to evil, chaos to order, see (J.Tweet et al. 2012). Yet, reputation and role play is far beyond this simplistic, and very often players will find themselves in situations where their perceived action is not the same as their nature. *World of Darkness*, e.g. *Vampire: The Masquerade* (Achilli et al. 2011), represents this model of nature, how they are, and demeanor, how they wish to be perceived, via their archetype models this disconnect more fluidly, giving the player rewards of bonus points to willpower based off acting their nature, while demeanors should be role played, which the GM is able to monitor.

The personal level of relationships has been examined in a limited manner in such games as *Fallout 4*'s followers, who each have specific likes and dislikes with a factional system for all other NPCs beyond the limited set of followers with their own reputation system (Bethesda Softworks 2015). Further, player's reputation only changes based on actions made in their presence when the liked or offending action is made or based on your alliances with one of the faction in key narrative sections in the game.

Memory of NPC agents has been extensively examined in the framework of decision making models. (Li, Balint, and Allbeck 2013) examines the use of a parameterized model of knowledge, which allows for memory and forgetting, in order to allow for human like behaviours based on memories in NPC agents. Mooney et al. (Mooney and Allbeck 2014) look at NPC reputations of players based on a Bayesian model; previous interactions with the player are monitored and expected future actions are predicted.

Further, there is a level of work in the creation of narrative using NPCs and reacting to memories. In *Talk of the Town*(Ryan and Wardrip-Fruin 2016) dialogs are created based off of inputs including the game state, these inputs could include player reputation as part of a dialog generator. This team has also worked on allowing for characters to be deceptive in their actions in a limited transfer of knowledge(Ryan et al. 2015).

However, what has been lacking is how the communications can be made via interactions and the role of the propagation of information on a large scale. Brown and Qu (Brown and Qu 2015) examine this problem of NPCs allowing for instant messaging about the game world in a variety of contexts, and give rationales for using elements of cascade method updates as a method of passing information.

This paper provides an early implementation of such a cascade of updates and examines the flow of information about a number of graphs. Three types are examined: a set of test graphs of small size in order to act as a proof for the method, a series of synthetic graphs with expectations in their properties (small world graphs), and graphs extracted from actual interactions made by NPCs in the digital games. The selected digital games graphs being *The Elder Scrolls 4: Oblivion* and *Fallout 4* (Bethesda Softworks 2015). It is our hypothesis, based on other implementations of cascades, that the underlaying graph of connections of characters will have dramatic effects upon the propagation of information, however, using basic knowledge of graph parameters, developers will be allowed for a level of control of this propagation and expected outcomes can be modeled.

Reputation on Networks

Reputation computation is well studied in many fields of study including multi-agent systems, social networks, autonomous systems to name a few. Reputation is used in decision problems where one needs to make a decision with insufficient information. In this section, we discuss how reputation propagation is modeled on real network applications.

Real life multi-agent environments can be turned into graphs with vertices as agents. Therefore, reputation propagation on graphs concern various types of applications such as vehicular networks (Hussain et al. 2016) and author networks (Lee et al. 2011). (Khairullina et al. 2015) studies how different types of information has different patterns of propagation behaviors on social networks. Authors shows that depending on contents, such as advertisements and social appeals, the spreadings of information differ. In this paper, we also show that how reputation propagates differently in different types of networks. In (Lee and Oh 2014), authors study the convergence of reputation based on *Bayesian game theory*. They proposed a reputation game where players are either honest or dishonest. Then based on the behaviors of players, they showed that playing honest is an evolutionary stable strategy. In our work, we model interactions among NPCs based on their preferences and test saturation of information on various graphs. In (Lee and Oh 2013), authors introduced a model for reputation propagations in social networks. We also show how our reputation propagates in various networks according to the proposed reputation system for NPCs.

Proposed Model

In this section, we formally introduce our reputation model for NPCs. In many game environments, information propagates instantly. For example, if a user kills the *dragon*, everyone in the game knows about the incident. Since the instance knowledge propagation is neither realistic nor ideal, we propose a realistic information transfer model based on reputation of NPCs.

Model Requirements

In order to improve treatment of interactions among NPCs, a set of requirements is defined for realistic information transfer system as follows:

- **Beliefs** NPCs have beliefs about the state of the world which may be correct or incorrect compared to the true state of the world.
- **Communications** NPCs may communicate their beliefs to other NPCs.
- **Updates** NPCs may update their beliefs based on new observations of the world or information that was obtained via *communications*.
- **Spacial and Temporal** NPCs are located in a game space and *communications* must pass through with respect to the space/time and *beliefs* will change via *updates* over time.
- **Reliability** NPCs may lie, misremember, and forget.
- **Persistence** NPCs are persistent objects in a game world and their memories are retained beyond any number of interactions.
- **Relationships** NPCs engage in relationships to various factions and this places a network on their *communications*.
- **Trust** NPCs trust or mistrust each other in the *relationships* and will only change their *beliefs* via *updates*.
- **Goals** NPCs have goals, ambitions, dreams, desires, and will work towards them, i.e. might lie or not make communications to serve their or a factions goals.

Considering previously listed elements, we build our knowledge transfer system based on communications among NPCs which are influenced by reputation of NPCs.

Basic concepts

Now we introduce preliminary concepts which are used in our model to update reputation values.

- **Event** An event occurs when a player makes some actions in game. Events are usually tied to quests and this model can extend traditional RPG quest games easily.
- **Category** Events will have associated *categories* which briefly describe the parts of the game the events are related to. For example, when we introduce an event in traditional RPGs, an event for helping a poor woman will have the following associated categories: *old*, *woman*, *help*, etc.

- **Weight** We also introduce a special parameter to each event, its *weight*. The greater the weight of an event, the more it affects the game world. In other worlds, bigger weights can imply bigger payoffs or influence.
- **Reputation** A players reputation can be expressed as a number for each NPC independently, i.e. each NPC has its own perception for the player. A NPC computes players' reputation upon every event. A NPC evaluates the events according to its preferences about the world. Weighted aggregation of all evaluated events for a player is expressed as the NPC's reputation to the player.

Non-Player Characters

Here, we describe NPCs to support our model. Every NPC contains the following.

- *like* category: list of topics that the NPC likes
- *hate* category: list of topics that the NPC do not like
- *memory*: current knowledge about events of the NPC. Memory degrades with time.
- *unforgettable*: the events in this category will never be forgotten. If a NPC becomes knowledgeable about an *unforgettable* event, they will remember forever.

Obtaining Knowledge

The process of obtaining knowledge about events for NPCs is simply an addition of the new event to its memory. Associated time and location of the event is also remembered when the knowledge is obtained. The time and location are important information to compute the *value function* of NPCs.

NPCs get knowledge about events in two ways: (1) Being a witness of an event (Direct experience) and (2) Hearing about an event from other NPCs (Indirect experience). The first case is simple. NPCs add the event to his memory set. In the next section, we discuss the second case (indirect experience) in detail.

Indirect Interactions: Obtaining Knowledge Through Communications

When two NPCs meet, they share knowledge about events. We assume that the knowledge transfer occurs in one direction at a time. This process can be described with the following steps.

1. For every event in the memory, a NPC calculates the value of the event according to the *value function*.
2. An event with the top value is picked for the current communication.
3. The picked event is then shared among all NPCs who are participating to the current communication. If some of the NPCs already know about the event, they refresh the time of the event as if they have just learned. If the NPCs did not know about the event, they add it to their memory with associated information.

Value Function and Spreading

All events are spread with respect to the *value function*. Value function expresses the importance of an event to the NPC. We believe that *value function* can model realistic information spreading since humans tend to discuss or communicate about topics which have more importance to them in any given situations. We also introduce a time decaying function as shown in Equation (1) which captures the decay of importance of the event as time elapses.

$$f_{decay}(e_u, T) = 1 - \frac{(t - t^{e_u})}{T} \tag{1}$$

where $e_u \in Event_u$ is an event in the memory of a NPC $u \in N$, t is the current time, t^e is the time of e and T is a constant which defines the memory. After T time is passed the user forgets about the event.

In order to compute the *value function*, we need to count categories of the event that the NPC like or hate.

$$likes(e_u, u) = |C(l_u) \cap C(e_u)| \tag{2}$$

$$hates(e_u, u) = |C(h_u) \cap C(e_u)| \tag{3}$$

where $e_u \in Event$, $u \in N$, $C(l_u)$ is the set of categories that u likes, $C(h_u)$ is the set of categories that u hates, and $C(e_u)$ is the set of categories that are associated with e_u.

Finally we define the *value function* as following.

$$f_{value}(e_u, u) = f_{decay}(e_u, T) * [(likes(e_u, u) - hates(e_u, u)) * w^{e_u}] \tag{4}$$

where $u \in Event_u$, $u \in N$, w^{e_u} is the weight of e_u given by the world and we use $T = 1000$ in this paper.

Reputation Updates

NPCs($u \in N$) have reputation values about players($p \in P$), $reputation_u(p)$. Reputation values are updated when u learns about new events ($e \in Events$). Reputation is a normalized sum of values of the events which a NPC has in memory as defined in Equation (5).

The *value function* is applied to all the events that each NPC has and then reputation is updated with the normalized sum of the *value functions*.

$$reputation_u(p) = normalize\left(\sum_{e_u \in Events_u} f_{value}(e_u, u)\right) \tag{5}$$

Different normalization methods can be used depending on the game environments. For example, *The Elder Scrolls 4: Oblivion* and *Fallout 4* use normalized reputation scale of [0,100]. The reputation levels from NPCs towards players are useful in many game environments to adjust reactions of NPCs to the players.

Trust among NPCs

In the current proposed model there is no trust levels among NPCs (there is reputation from NPCs to players). Therefore when NPCs communicate they are assumed to trust each other and the knowledge is transfered from one to another. To make interactions more realistic we introduce a transmission probability of knowledge between NPCs which determines the probability that the knowledge will be transfered. In other words, the more two NPCs trust each other the higher the chances are to successfully transfer the knowledge.

We define the level of trust among NPCs based on the common *likes* and *hates* categories.

$$trust(a,b) = m * \left(\frac{|C(l_a) \cap C(l_b)| + |C(h_a) \cap C(h_b)|}{|C(l_a) \cup C(h_a)| + |C(l_b) \cup C(h_b)|} + \epsilon \right) \tag{6}$$

where $a, b \in N$, ϵ is a minimum trust level (that may be imposed by the system) and m is *mood* parameter. $trust(a, b)$ computes the probability of knowledge transfer between two NPCs, a and b. We further introduce a *mood* parameter which affects probability of information transfer. *Mood* is a value between 0 and 1 and the greater the *mood* of the NPC, the more likely the NPC shares knowledge with other NPCs.

An Example

This section presents a use case of the proposed model. Let us introduce Alice who is a NPC. Alice likes *cats, dogs, animals* and her *job*. Alice hates *stupidity* and *spiders* because she has *arachnophobia*. Alice is represented as follows by our model.

$$\begin{aligned} Alice :\{ & l = \{cats, dogs, animals, job\}, \\ & h = \{stupidity, spiders, arachnophobia\}, \\ & memory = \{\}, \\ & unforgettable = \{\} | Alice \in N\} \end{aligned}$$

There is also another NPC, Bob. He likes *spiders* and tries to *help* them if possible.

$$\begin{aligned} Bob :\{ & l = \{spiders, help\}, \\ & h = \{\}, \\ & memory = \{\}, \\ & unforgettable = \{\} | Bob \in N\} \end{aligned}$$

Assume that a player finished a quest with $id = 1$ which is related to spiders. For example, the quest is to save spiders in a cave from extinction. Then an event is generated.

$$e : (id = 1, categories = \{spiders, saving, help\}, weight = 25)$$

Imagine that Alice have observed the player (p) completing the quest. Then she updates her memory with the new event and adjusts the reputation of the player.

$$\begin{aligned} & f_{value}(e_{id=1}, Alice) = 1 \times (0 - 1) \times 25 = -25 \\ & reputation_{Alice}(p) = f_{value}(e_{id=1}, Alice) = -25 \end{aligned}$$

The reputation of the player for Alice is the same as the *value* of this event since this is the first event in her memory.

Now Alice wants to share her experience with Bob. We assume Alice has the maximum *mood*, i.e. 1.0.

$$\begin{aligned} & |C(l_{Alice}) \cap C(l_{Bob})| = 0, |C(h_{Alice}) \cap C(h_{Bob})| = 0, \\ & |C(l_{Alice}) \cup C(h_{Alice})| = 7, |C(l_{Bob}) \cup C(h_{Bob})| = 2, \\ & trust(Alice, Bob) = 1 \times ((0 + 0)/(2 + 7) + 0.5) = 0.5 \end{aligned}$$

Thus, the information is shared with Bob with the probability of 0.5. Assume that the communication was successful and Bob has learned about the event from Alice. Now Bob adjusts reputation of the player.

$$\begin{aligned} & f_{value}(e_{id=1}, Bob) = 1 \times (2 - 0) \times 25 = 50 \\ & reputation_{Bob}(p) = f_{value}(e_{id=1}, Bob) = 50 \end{aligned}$$

In the example we assumed that no time has passed between the actions and thus f_{decay} is 1.

Table 1: Graph Properties for Five Nodes

Graph Name	Short	Diameter	k-regularity
Complete Graph	K5	1	4
Linear Graph	L5	4	1
Cycle Graph	C5	4	2
Star Graph	S5	2	1

Experimental Settings

A *Simple Graph* $G(V, E)$ is a non-empty set V of vertexes or nodes and a set E of unordered pairs of elements of V, called edges. Two distinct nodes, v_1 and v_2, are said to be *neighbours* if $(v_1, v_2) \in E$. The number of edges in E which contain a vertex is called the *degree* of the vertex. A graph is *connected* if there is a pathway via the edges from any vertex to any other vertex. If all the vertices in a graph have the same degree, then the graph is said to be *regular*. A weaker evaluation of regularity, a graph is said to be $k - regular$ for any k if there exists a degree of at least k on very node. The *diameter* of a graph is the largest number of edges in any shortest path between any two of the vertices.

Graphs of Size Five

In order to act as a verification of the method meeting with developer expectations we examine a series of graphs of five nodes. They are the fully connected graph (K5), a linear graph (L5), a cyclic graph (C5), and a star graph (S5). The properties of their connections are in Table 1. In order to further test the propagation based on connection the starting node is changed in the Linear and Star graphs. In the linear graph we examine propagations happening on three starting points, the first, second, and third node in the line. For the star graph we examine one of the extremal nodes, and the centre node. All other nodes in these graphs should have similar properties due to isomorphisms. Though it should be the case that information starting from extremal nodes will take longer to propagate.

Small World Graphs

We also test our knowledge transfer model on random graphs. We generated three different sized small world graphs using Watts-Strogatz model.

The first graph (small world graph 1 as shown in Figure 1) contains 50 nodes and has an average degree of 4. Vertex 1 is chosen as a starting node for propagation. The diameter of small world graph 1 is 6 and so is the required number of steps to fully propagate, i.e. every node becomes aware of the knowledge which was started from the vertex 1 (when transmission probability is 1), as shown in Table 3. As the transmission probability decreases, we observe that the number of steps needed to transfer information to every node increases.

Small world graph 2 has 100 nodes and an average degree of 5. We started propagating information from node 8 and the required number of steps for the full propagation was 5.0 on average.

Small world graph 3 has 150 nodes with average degree of 7. It is very dense graph and its diameter is 4. We start information transmission from vertex number 8. In Table 3, we show summarized results with different transmission probabilities. As expected, the denser the graph is the shorter the required steps are to fully transfer knowledge on the graph.

Graphs from Games

We created two graphs from real games. These games are from Bethesda Softworks - *The Elder Scrolls 4: Oblivion* and *Fallout 4*. We created the same set of NPCs again, now connecting them into a big game world. You can see that graphs in Figures 4 and 5.

Experimental Setups

In the following sections, we show our experimental results on various types of graphs. In order to produce non-biased and comparable results on different graphs, we create a set of NPCs (according to the size of the graphs) who *like* and *hate* exactly the same categories. Then the probability of transferring ($trust$ between two NPCs as defined in Equation (6)) the information is 1. Therefore we can adjust mood parameter, m, to assign exact transmission probabilities for the experiments. Lastly, we connect NPCs to form graphs and run each simulation at least thirty times to observe the spread of information throughout the graphs.

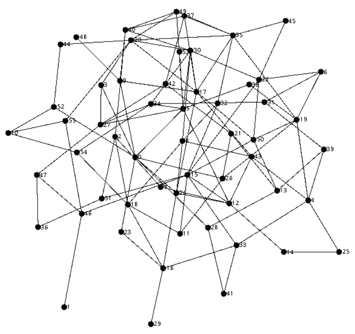
Figure 1: Small world graph 1

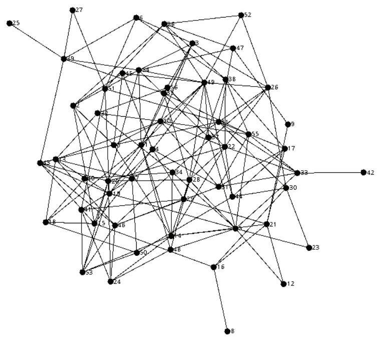
Figure 2: Small world graph 2

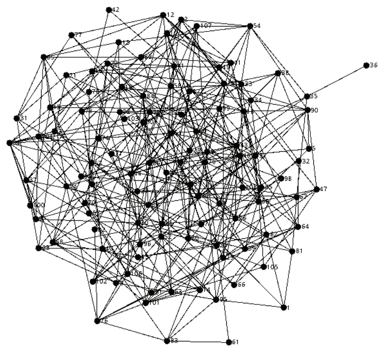
Figure 3: Small world graph 3

Results

Graphs of Size Five

The graphs of size five gives us a good estimate of local interactions which will be seen in larger graphs. As expected when the transmission probability is unity, a fully reliable transfer of information on the graph, the number of interactions in order to saturate the knowledge is related to transfer is bounded by the diameter of the graph. Though in C5 or middle position of L5 and S5 the propagation is made much faster. Further, starting information transfer from the extremal nodes in a graph extends the transmission time as seen in L5 and S5.

As the transmission channel becomes less reliable the time till a full transfer to everyone in the graph increases. K5 shows little effect of this change, it is also the most regular graph, meaning there are more opportunities for a successful transmission action.

This gives proof of the expectation that the diameter and regularity and node degree of graphs is able to determine the maximum length of the transmission time and the affect of reducing transmission probabilities before information is saturated in the graph. A developer can therefore use these graph parameters as controls.

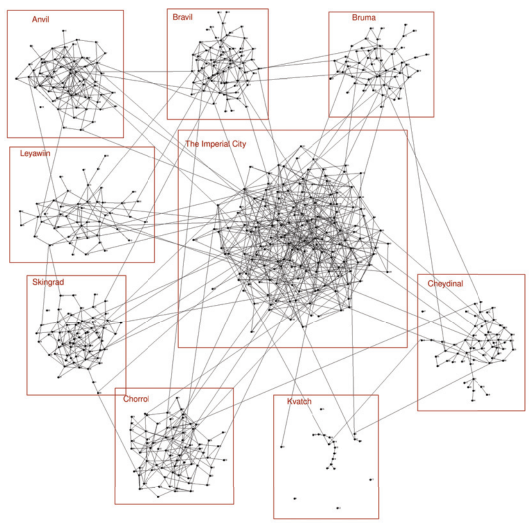

Figure 4: Graph from *The Elder Scrolls 4: Oblivion*

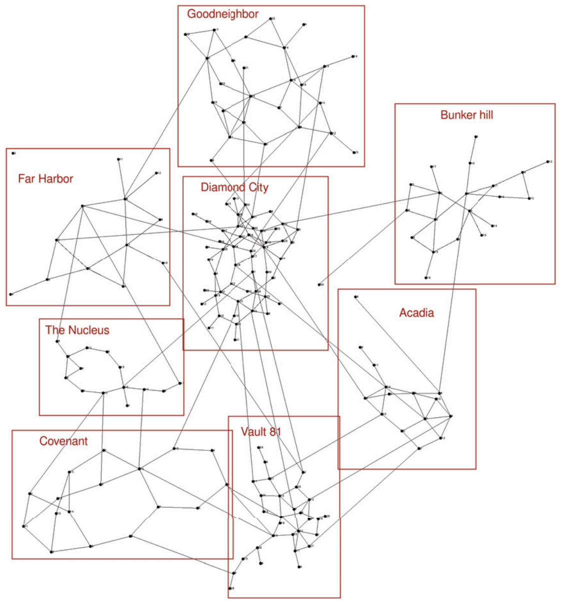

Figure 5: Graph from *Fallout 4*

Abstract Small World Graphs

We created the same set of NPCs. The only change is that we connected NPCs according to the graph generated.

The event started to propagate from different nodes. You can see the results of the runs in Table 3. In these cases there is a statistically significant increase, using a t-test between means, in all three graphs between all transmission probabilities. The denser the graph the faster the saturation, as seen with Graph 3 as apposed to Graph 1.

Graphs from Games

The results for the graphs from *Oblivion* and *Fallout 4* are in Table 4. For both the stating location of the information has a significant effect on the propagation rates. However, As transmission probabilities decrease however the difference begins to decline, for example there is no statistically detectable difference at the transmission probability of .25 between information spread from Skingrad than from Kvatch, see Figure 4 for the city locations in the graph of NPC connections. This could be due to the ability of the smaller cities to press their information into the core, if the trader is not making the transmission between the smaller cities (Skingrad and Kvatch) to the central hub of Imperial City, then this causes a delay. Note that information for example that begins in Imperial City with even traders loosing half of the messages, is still able to be propagated across the network as quickly as information started in these small towns with perfect transmission.

This transmission property continues in *Fallout 4*, see Figure 5 for the city defined graph, where information starting in Arcadia takes significantly longer than that started in Diamond City, the largest city in the game. In terms of game narrative, the major cities would become information hubs as much as they are trading and quest central hubs. Players would be rewarded in the monitoring of their reputation in the central cities as a method of seeing how they are perceived throughout the game world.

Conclusions

Though these examination on some graphs with known properties of size five, it was shown that this cascade method of reputation can be controlled via the construction of the graph. Graphs with low k-regularity show messages passed slower than those with high values. This provides a control for the designer and developer on how information is

Table 2: Mean and standard deviation of the number of iterations of communications until the information is fully propagated across the graphs of size five with N=1000 runs.

Graph	Transmission Probability			
p	1	0.75	0.5	0.25
K5	1.0	1.716 (0.50)	2.264 (0.64)	4.016 (1.55)
L5(Vertex 1)	4.0	5.308 (1.28)	8.068 (2.85)	15.845 (6.78)
L5(Vertex 2)	3.0	4.064 (1.19)	6.046 (2.42)	12.191 (5.77)
L5(Vertex 3)	2.0	3.08 (1.11)	5.07 (2.23)	10.638 (5.34)
C5	2.0	2.887 (0.78)	4.254 (1.44)	8.099 (3.34)
S5(edge)	2.0	3.08 (1.04)	5.155 (2.14)	10.752 (5.31)
S5(centre)	1.0	2.011 (0.92)	3.456 (1.67)	7.87 (4.52)

Table 3: Mean and standard deviation of the number of iterations of communications until the information is fully propagated across the small world graphs with N=30 runs.

Graph	Transmission Probability			
p	1	0.75	0.5	0.25
Small world 1	6.0	7.3 (0.702)	11.1 (1.999)	22.3 (6.36)
Small world 2	5.0	6.733 (0.944)	9.9 (2.04)	17.2 (5.51)
Small world 3	4.0	4.5 (0.57)	6.43 (1.278)	12.666 (4.197)

Table 4: Mean and standard deviation of the number of iterations of communications until the information is fully propagated across the graph for the graphs developed from digital games with N=30 runs.

Graph (City:Initial Node)	Transmission Probability			
$p =$	1	0.75	0.5	0.25
Oblivion (Skingrad:27)	9	10.1 (0.89)	13.0 (1.89)	19.867 (3.12)
Oblivion (Imperial city:116)	5	6.367 (0.56)	9.3 (1.622)	16.6 (2.80)
Oblivion (Kvatch:4)	11	12.8 (0.97)	15.167 (1.60)	21.633 (3.46)
Fallout 4 (Acadia:12)	10	12.667 (1.06)	15.567 (2.596)	29.07 (5.57)
Fallout 4 (Diamond city:26)	7	9.63 (0.89)	13.13 (1.87)	22.97 (2.97)

passes within the game space. Graphs developed from current games demonstrate a larger levels of connection inside of cities and villages, with few connections between these nodes mostly via traveling traders.The datasets for the graphs, including those for the real games, are made publicly available at: <*https://github.com/nikitakraev/innothesis*>.

In our current study we did not examine the trust model, in future work we aim to examine the role of trust within the graph in order to determine the effect. Our hypothesis is that trust models will slow down propagations and if based on factional cities can even block information from being passed into entire towns. The Key aspects being the linkages between traders.

The model tests have not demonstrated the decay functionality to represent the interactions as agent based events - and our analysis is not taking to account the movement in the game space of the agents before a transaction is made. This is particularly important given that the game has a physical representation of each NPC agent moving about in the space. Traders do not *teleport* between cities - they move in the game world between cells of the game world over time in predefined trading routes. Meaning that while our analysis may show that there is a connection between Arcdia and Diamond city, it will not be as heavily utilized. This relates back to the special/temporal issues of our idealized model. This could be expanded upon by placing our NPC model into the game space and observing the interactions in the world based on time, though this analysis will allow for developers to examine how the graphs chosen effect transfers of information on connections of one time step between all actors.

References

2K Games. 2007. Bioshock.

Achilli, J.; Bailey, R.; McFarland, M.; and Webb, E. 2011. *Vampire: The Masquerade 20th Anniversary Edition*. White Wolf.

Bethesda Softworks. 2008. Fallout 3.

Bethesda Softworks. 2010. Fallout: New Vegas.

Bethesda Softworks. 2015. Fallout 4.

Brown, J. A., and Qu, Q. 2015. Systems for player reputation with NPC agents. In *IEEE Conference on Computational Intelligence in Games 2015 (CIG)*, 546–547.

Hussain, R.; Nawaz, W.; Lee, J.; Son, J.; and Seo, J. T. 2016. A hybrid trust management framework for vehicular social networks. In *International Conference on Computational Social Networks*, 214–225. Springer International Publishing.

J.Tweet et al. 2012. *Dungeons & Dragons Player's Handbook*. Wizards of the Coast.

Khairullina, A.; Lee, J.; Jang, G.; and Myaeng, S.-H. 2015. Observing behaviors of information diffusion models for diverse topics of posts on vk. In *Data Mining Workshop (ICDMW), 2015 IEEE International Conference on*, 1098–1102. IEEE.

Lee, J. Y., and Oh, J. C. 2013. A model for recursive propagations of reputations in social networks. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 666–670. ACM.

Lee, J., and Oh, J. C. 2014. Convergence of true cooperations in bayesian reputation game. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2014 IEEE 13th International Conference on*, 487–494. IEEE.

Lee, J.; Duan, Y.; Oh, J. C.; Du, W.; Blair, H.; Wang, L.; and Jin, X. 2011. Automatic reputation computation through document analysis: A social network approach. In *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*, 559–560. IEEE.

Li, W. P.; Balint, T.; and Allbeck, J. M. 2013. *Using a Parameterized Memory Model to Modulate NPC AI.* Berlin, Heidelberg: Springer Berlin Heidelberg. 1–14.

Mooney, J., and Allbeck, J. M. 2014. Rethinking npc intelligence: a new reputation system. In *MIG*.

Ryan, J., and Wardrip-Fruin, M. M. N. 2016. Characters who speak their minds: Dialogue generation in talk of the town. In *Proceedings of The Twelfth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE-16)*, 204–210.

Ryan, J. O.; Summerville, A.; Mateas, M.; and Wardrip-Fruin, N. 2015. Toward characters who observe, tell, misremember, and lie. In *Proceedings of the 2nd Workshop on Experimental AI in Games*.