LEVEL OF DETAIL EVENT GENERATION

Luis Francisco Flores Vazquez

Thesis of 60 ECTS credits
Master of Science (M.Sc.) in Computer Science

June 2017

Level Of Detail Event Generation

by

Luis Francisco Flores Vazquez

Thesis of 60 ECTS credits submitted to the School of Computer Science
at Reykjavík University in partial fulfillment
of the requirements for the degree of
Master of Science (M.Sc.) in Computer Science

June 2017

Supervisor:

David James Thue, Supervisor
Assistant Professor, Reykjavík University, Iceland

Examiner:

Stephan Schiffel, Examiner
Assistant Professor, Reykjavík University, Iceland

Kári Halldórsson, Examiner
Adjunct Professor, Reykjavík University, Iceland

Level Of Detail Event Generation

Luis Francisco Flores Vazquez

June 2017

Abstract

Level of detail is a method that involves optimizing the amount of detail that is simulated for some entity. We introduce an event generation method to optimize the level of detail of upcoming events in a simulation. Our method implements a cognitive model, which uses an estimate of the player's knowledge to estimate their interest in different aspects of the world. Our method predicts the salience of upcoming events, and uses this salience value to define the level of detail of potential new events. This dissertation provides a strong basis for future work, which will use our model to predict the salience of upcoming events in a simulation and optimize the level of detail with which they are generated.

Level Of Detail Event Generation

Luis Francisco Flores Vazquez

júní 2017

Útdráttur

Sundurliðun er aðferð sem felur í sér að besta magn smáaatriða sem er líkt eftir fyrir einhverja einingu. Við kynnum aðferð sem framkallar atburði sem hefur það að leiðarljósi að nýta sundurliðun komandi atburða í eftirlíkingu. Aðferð okkar framkvæmir hugrænt líkan sem áætlar bæði þekkingu og áhuga leikmannsins á mismunandi þáttum heimsins. Aðferðin spáir fyrir um áberanleika komandi atburða og notar gildi áberanleikans til að skilgreina sundurliðun nýrra mögulegra atburða. Þessi ritgerð veitir sterkan grunn fyrir framtíðarvinnu sem mun nýta líkan okkar til að spá fyrir um áberanleika komandi atburða í eftirlíkingu og besta sundurliðun á framköllun þeirra.

Level Of Detail Event Generation

Luis Francisco Flores Vazquez

Thesis of 60 ECTS credits submitted to the School of Computer Science
at Reykjavík University in partial fulfillment of
the requirements for the degree of
Master of Science (M.Sc.) in Computer Science

June 2017

Student:

....

Luis Francisco Flores Vazquez

Supervisor:

....

David James Thue

Examiner:

....

Stephan Schiffel

....

Kári Halldórsson

....
date

....
Luis Francisco Flores Vazquez
Master of Science

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Level of detail (LOD) is a method that involves reducing the complexity of a simulation when doing so would be transparent to one or more observers of the simulation. It can decrease the complexity of geometry, artificial intelligence, or physics computations. Since simulating everything at a high level of detail could be very expensive, LOD is very important in video games, as it allows a larger game world to be simulated while still maintaining believability for the players of the game.

The concept of level of detail was introduced in the earliest days of computer graphics. Computation resources were limited and rendering complex geometric objects was not practical. Reducing the complexity of geometric details allowed programmers to improve the rendering time and increase the number of geometric objects in each scene. LOD was applied mainly to manage geometric detail. Today, level of detail is used in many fields and for objects' properties other than their geometry. LOD can be used in image rendering [1], video games [2], 3D modelling [3], 3D graphics [4], simulations [5], etc.

A common LOD method is to use a distance-based algorithm, which measures the distance of each given object from the observer, or player. Each object in the scene is evaluated by proximity. The closer it is to the player, the more important it becomes to the player and more details are rendered. Different "levels" of detail are switched between at certain distance thresholds, and the number of details rendered is updated dynamically.

Applications of the level of detail method are used in computer games, not only in visualisation but in the artificial intelligence reasoning and physics engines. Cournoyer and Fortier [2] described how this method is used effectively in the Assassin's Creed Unity video game. Graphical LOD and Simulation LOD allows the video game to render a large population of agents with only a minimal quality loss.

Using levels of detail in a simulation optimises computing resources and allows a larger and more complex world to be rendered both by using cheaper agent AI reasoning and decreasing the geometric detail when possible. For example, the LOD method used by Cournoyer and Fortier [2] depends on the agent distance to the player. It decreases the agent reasoning and geometry detail when the player is not close to them. When the player is far away, the agents have a simple and inexpensive behaviour such as walk and turn around, and the geometric detail has very few polygons as the user cannot perceive more details at that distance. The agents are more interactive and have better image quality when the player is at a closer distance to them. The player at this point can appreciate facial expressions and other geometric details. The highest level of detail used is very complex; the agent reasoning is expensive and the geometry is very elaborate. This level is used when the player is at a very close range to the agents (e.g., close enough to interact). At this range the agents are the main focus of the player, so the video game simulates very detailed geometry and autonomous artificial intelligence agents.

Simulations can take advantage of this method by reducing the details of different aspects of the simulation that are presented to the player. However, the details should be relevant to the player and avoid omitting any essential properties. Keeping the essential properties is very important because the simulation must render relevant data without overwhelming the player. Simulating a high level of detail when it is not necessary would be wasteful; the details would be irrelevant to the player. The simulation should know what elements can be omitted during rendering, given the state of the world. Knowing which properties can be removed in a simulation is complicated, as there are many conditions and constraints that can influence the evaluation of the properties. For instance, a simulation that generates events in a world can omit different details if the player is not aware of an event's intended location, or if the player cannot access that area. The event should be generated with minimum details to avoid wasting resources, while still keeping relevant information to the player. If a new generated event is expected to be very relevant to the player, the simulation should display every detail of the event.

Measuring which events are believable can be hard during a simulation, as the player considers different actions to be convincing or not depending on the world context. We propose a method to optimize the level of detail with which events in a simulated world are generated, using an estimate of the players knowledge to estimate their interest in different aspects of the world. Doing so can help the method avoid creating unnecessary details, enabling larger simulations to be run.

In our method, we account for the player's knowledge during the event generation process. The player's knowledge is very important to creating upcoming events because we can avoid generating unnecessary details depending on how salient the event's details are to the player. For instance, we can avoid specific location details if the player doesn't know the location at which the event is generated. For example, suppose that an important character in the simulation is hosting an event in "city x". If the player has knowledge of "city x", they might want more details of the event such as the directions to the event. However, if the player doesn't have knowledge of that city, or the player cannot possibly attend the event, the simulation can avoid creating some details like the address of the event.

1.1 Background

Cardona-Rivera, Cassell, Ware, *et al.* [6] described a computational model of narrative which predicts the salience of previous events in the user's memory based on the current world state. The salience of an event suggests how remarkable the event is as the simulation runs. They used salience to determine the link between new events and previous events.

The model that we propose uses salience as a metric when generating upcoming events; a lack of salience for a potential new event means that we can reduce the amount of detail that gets generated for the properties of that event. Our model is based on Cardona-Rivera, Cassell, Ware, *et al.*'s [6] Indexter model, which we describe in the following section.

1.1.1 Indexter Model

Cardona-Rivera, Cassell, Ware, *et al.* [6] predicted the salience of narrative events by keeping track of the user's comprehension during the game play. The salience of an event indicates how relevant or memorable it is to the player. The absence of salience means the event is not relevant or memorable to the player.

The Indexter model contains two fundamental bases. The first is a story comprehension model by Zwaan, Langston, and Graesser [7] and Zwaan and Radvansky [8], called the Event-Indexing Situation Model (EISM), which is a cognitive model used in online narrative comprehension.

The second is an AI narrative model, which follows directly from IPOCL (Intentional, Partial Order, Causal Link) planning by Riedl and Young [9]. Indexter model is constructed by extending the IPOCL model with information from the Event-Indexing Situation Model. The IPOCL model structure already contained many features needed to represent an EISM model.

1.1.1.1 The Event-Indexing Situation Model

The event-indexing situation model (EISM) presented by Zwaan, Langston, and Graesser [7] is a cognitive model of online narrative comprehension. Specific events in the story world are represented by a situation model defined by psychologists. The EISM categorises a narrative into events with important details. Each event is categorised by different key factors or indices. EISM represents each event by indexing the following elements:

- **Time index**. It represents the interval of time where an event occurred. An interval of time is bounded to a list of constants. The constants are created by the author or automatically.

- **Space index**. It represents the location where an event occurred. A location is bounded to a list of constants that refer to locations in the given world. These constants are specified in the event's description, and are defined by the author.

- **Protagonist index**. It represents when the main character of the story is involved. The protagonist is a single character that represents the main role in the entire story.

- **Causation index**. It represents when two events have a causal relation. A relation exist if event $e1$ had not happened, then event $e2$ could not have possibly happened.

- **Intention index**. It represents whether the event furthers a character's plan to achieve a single goal, if any.

The EISM does not specify which indices are more remarkable than others.

1.1.1.2 Building the Indices

The EISM makes predictions about which indices are shared between two events. The criteria for computing those indices are the following:

- Time index is shared when two events happen during the same time frame.

- Space index is shared when two events take place in the same location.

- Protagonist index is shared when two events involve the main character of the story. EISM takes into account only a single main character.

- Causation index is shared when two events are related casually. Cardona-Rivera, Cassell, Ware, *et al.* [6] describe two different types of casual relation: *direct* and *indirect*. A *direct casual relation* from $e1$ to $e2$ is when both events satisfy a logical criteria: if event $e1$ had not occurred, then event $e2$ could not have possible happened. An *indirect casual relation* is when events are sufficient but not necessary to trigger other events.

- Intention index is shared when two events aim to achieve the same goal.

1.1.1.3 Calculating Salience

The Indexter model computes the salience of the properties of two events. Two events have a value of 1 for a given property when they both share the index with the same name. For example, when two events occur in the same location, the location index is shared and they have a value of 1 for space salience.

Cardona-Rivera, Cassell, Ware, *et al.* [6] attach a weight to each index such that the weighted sum of all of the indices is within the range of 0 and 1. Given this constraint, the equation to calculate the salience of the event is the following:

$$salience(e_i, e_n) = w_1 t_{e_n} + w_2 s_{e_n} + w_3 p_{e_n} + w_4 c_{e_n} + w_5 i_{e_n} \tag{1.1}$$

This formula represents the shared indices of events e_i and e_n, where t_{e_n} represents the time index, s_{e_n} represents the space index, p_{e_n} represents the protagonist index, c_{e_n} represents the causality index, and i_{e_n} represents the intention index. If two events share an index x then $x_{e_i} = x_{e_n} = 1$, otherwise $x_{e_i} = x_{e_n} = 0$.

The coefficient w_j is the weight of the salience of indices. The value of each index is within the range $[0, 1]$.

The authors of the EISM do not state the weight of the indices, and the authors of the Indexter model set the same weight to every index, that is $\forall j, w_j = 0.2$. The final equation is:

$$salience(e_i, e_n) = 0.2 t_{e_n} + 0.2 s_{e_n} + 0.2 p_{e_n} + 0.2 c_{e_n} + 0.2 i_{e_n} \tag{1.2}$$

For example, consider the following story. A troll breaks into a house $(e1)$, steals a treasure in the house $(e2)$, and runs off to a cave $(e3)$. Later, the owner takes a weapon from the house $(e4)$ to prepare for his quest. During the night, the owner travels to the troll cave $(e5)$, slays the troll $(e6)$, takes his belongings $(e7)$, and returns to the house $(e8)$. The *owner* is the protagonist in this story.

In order to calculate the event salience of $e1 =$ (Breaks into, Troll, House) at event $e7 =$ (Take, Owner, Treasure), the Indexter model checks which indices are shared.

Event $e1$ is not connected in space, protagonist, time or intention to $e7$; however, they are connected because $e1$ had to occur in order for $e7$ to occur. Then using Equation 1.2:

$$\begin{aligned} salience(e_1, e_7) &= (0.2 \times t_{e_7}) + (0.2 \times s_{e_7}) + (0.2 \times p_{e_7}) + (0.2 \times c_{e_7}) + (0.2 \times i_{e_7}) \\ &= (0.2 \times 0) + (0.2 \times 0) + (0.2 \times 0) + (0.2 \times 1) + (0.2 \times 0) = 0.2 \end{aligned} \tag{1.3}$$

The salience of event e_1 at event e_7 is thus 0.2.

1.1.1.4 Limitations of the Indexter Model

We believe that important details are lost by only using this salience metric when generating new events. For example, the space index would be 0 if the player was not at the event's location; therefore, the generator would avoid creating details for this index. However, we are interested in displaying some details of the event's location, even if the player only has knowledge of this location.

The Indexter model compares two different events during gameplay, but we propose a new approach to generate upcoming events by evaluating potential new events in terms of the player's knowledge and current state using the Indexter model. This method computes salience for each index within a range of $[0, 1]$. This way we can generate new events with different levels of detail according to the player's knowledge and the current world state.

Chapter 2

Problem Formulation

While playing computer games, players' experiences are shaped by their observations. Crucially, their game play experience is the same if unseen objects are not rendered at all.

Many aspects of a simulation can benefit from LOD techniques. Distance is a natural choice when applying LOD to objects, but does it work for every aspect of a simulation? There might be cases where the simulation needs a high LOD event when the player is far away. An example is an upcoming event that is critical to the player's pursuit of a given goal; the simulation should generate a high LOD for the event (in comparison to a side event) such that the player could still achieve their goals without attending it. We can use a LOD method to optimise the generation of events during the simulation.

2.1 Event Generator

In this dissertation we aim to create a method to optimize the generation of new events in a simulation using level of detail. Creating new events in a simulation can be computationally expensive, similarly to rendering geometry in the simulation. We can optimize the level of detail of events by understanding when they are and aren't important to players during the simulation. For example, creating a very detailed event that the player can't access would be a waste of computation, since the player would not fully appreciate those extra details. Instead, the generator should use a low level of detail.

When applying LOD to an event generator, a few questions arise regarding LOD methods and techniques: Is distance from the player still a feasible solution to determine LOD for events? Are there situations when we want to render a very detailed event when the player is far away? What event aspects can be reduced or omitted? In this work, we address those inquiries and present a novel level of detail event generator.

Applying level of detail to event descriptions can be challenging. When is a good moment to apply LOD to an event? After or before the creation of events? The method needs to know the event's properties before it can decide which of them are good candidates for applying LOD. We also need to determine if LOD is a good technique to produce an event description. It might seem like text summarisation methods can be applied to this problem effectively, but a few questions arise. A summarisation method can be applied after the event is generated, but this means that computational resources to generate this event were already used, and we might not need those details. We aim to solve this problem by not generating unnecessary details at all, thereby saving computational resources.

The generator should only create event properties that are salient for the player given the current state and player's knowledge. It should know how much information the player perceived and remembers about the current world. This way, the method could generate event properties that the player finds interesting. The generator should take into account important events in the world. The upcoming events should be created based on the player's knowledge. The generator should avoid creating unnecessary event details, and completely eliminate properties that are not of interest to the players. Events in a simulation can change the current state of the world, and some events might be more important in the world than others. The simulation should have a higher LOD in important events.

The generator should maintain believability when creating new events, because when players are immersed in the simulation, creating a new event out of context would likely be confusing for them, and such events might be ignored. Measuring what is unbelievable in a simulation is a difficult task, since it's hard to know what "unbelievable" means to the player given the world context. We need to identify what is reasonable or unrealistic in the simulation.

Another challenge is to model the player's knowledge. When an event is perceived, the model should identify which properties are easily remembered by the player, so that upcoming events can be optimized based on this information. An event can contain different properties such as location, time, description, agents involved, etc. Knowing which properties are more critical to the player is difficult. For instance, how can the model tell if an event's location is more or less important than its time or other property?

The model needs to distinguish between the properties of an event, to identify the most important properties to the player. Each event property has a salience, which identifies how remarkable the property is to the player. We need a metric to measure the salience of each property, and the overall salience in the event.

The model should describe what properties should be included or avoided for different levels of detail. A low level of detail event description contains the minimum information to formulate an event (e.g., title and location). A high level of detail event description is characterised by containing very specific and accurate properties of the event, such as directions, date, location, description, who is involved, the purpose of such event, etc. For example, if a player is currently in a different city from where a given event is held, the method can avoid generating irrelevant or unnecessary details about the event's location, like directions, specific time, etc. Such details could be both useless to the player and expensive to generate.

The weight factor on each event property should depend on the player's knowledge and world state. For example, for an event that is held in the same city as the player, the space salience should be higher than it would be for an event held in the next city. However, the event in the next city might be more relevant to the player if the other indices have higher salience. An important issue to tackle is to find a useful weight for every index; there might be a case where the time index is more important than the location index, or the other way around. The authors of EISM and Indexter model set an equal value in each index, that is $w = 0.2$.

We aim to create a flexible LOD event generation method by evaluating the player's knowledge and the current state of the simulation. The events generated must be as relevant as possible. The method should keep the simulation believable at every state given the context of the world. The generation of upcoming events must display enough properties to make the events believable.

2.2 Criteria for Success

While evaluating the generated events in a simulation, we need to keep in mind different factors.

- *Believability*. The event must contain properties to maintain credibility within the context of the simulation. The properties should describe the event well enough to keep the player interested in the upcoming events.
- *Level Of Detail*. The event must contain enough properties for the player to receive the notification of a new event that has been created. The expected salience for each property should determine how each property of the event gets generated, where a lower salience means that property will be discarded or will contain basic information, and a high salience means that property will contain detailed information.

When the players perceives a new event, they should get enough information about the event to keep the simulation believable. Events should have only essential details, since generating unnecessary details will waste resources. The description of the events should be relevant to the player at all times.

A good solution is when each event contains properties that the players think are relevant. We will assess players' perception of relevance by predicting how important different event properties are to them and comparing the results to ground truth values obtained via a survey. We will conclude that our method is successful if the predictions shows better results than what a uniform random predictor would produce.

Chapter 3

Related Work

Level of detail has been researched many times and applied to different aspects of a simulation. A level of detail method does not guarantee an optimal solution; there are cases where a LOD method is better than others depending on the structure of the world or simulation type. This chapter reviews relevant work on level of detail in simulations.

One popular approach to apply level of detail is by using a distance-based approach; a straightforward strategy is to check the object position with the player position [2]. The closer the object to the player, the higher level of detail is rendered. However, this paper does not use a distance-based method, instead, it takes the player's knowledge as a reference to decide the level of detail.

3.1 Level of Detail

Clark [10] introduced one of the first concepts of rendering geometry structures with different levels of details, depending on their size. He discussed how those methods helped to improve the rendered scene.

Senina, Rohrbach, Qiu, *et al.* [11] discussed existing approaches to autogenerate video descriptions. This approach produces three different level of detail descriptions for complex videos. The level of detail descriptions were listed as detailed, short, and one sentence description. They predicted a semantic representation and then created a natural language description. While their approach focused on images and videos, we aim to generate different levels of detail for descriptions for events.

Lin and Pan [12] combined different LOD methods on real-time human locomotion generation. They also developed a locomotion generation system by integrating this LOD method in a game engine. LOD is applied to geometry and animations, but not to events.

3.2 Simulation Level of Detail

Chenney, Arikan, and Forsyth [13] presented an example of proxy simulation to reduce computational resources of a simulation in a large world. A proxy swaps the place of objects that are unseen. The proxy makes sure that the visible objects enter the scene at an optimal time and agents behave properly when they are seen. This method used two levels of detail. The quality of the proxy is measured by how properly the method maintains believability given the context. This LOD method is applied to potential visible objects and it is applied to the agents' interactions. This is a distance-based approach, while we propose a knowledge-based approach.

Hodgins and Carlson [14] implemented a game that uses a dynamic simulation, where the simulation applies techniques of level of detail to the graphics. The game consisted of a one-legged robot. The creatures in the simulation switch between different levels of detail, allowing the game to achieve a good performance for a larger crowd of creatures. They defined a set of conditions for different LOD when a creature changed, based on the importance of the creature and the capabilities of the creatures given the current state. They assessed the performance by using different levels of detail and measuring the effect that decreasing the quality of the simulation had on the game.

Sunshine-Hill [15] presented the algorithm "LOD Trader", a method for optimising the perceptual quality of a simulation. Their approach focuses on maximising the perceived reality of the scene by estimating every agent's realism rendered within the scene. Their work lacks or ignores the influence of important events and the current player's knowledge of the given world, which should affect the agent's level of detail. Our approach takes into account the state of the world and the player's knowledge during the simulation.

Cournoyer and Fortier [2] describe a level of detail method used in the video game Assassin's Creed Unity. This method uses a distance-based approach where the AI and geometric detail depend on the player's distance from the agents. This method uses a "bulk" system, where each bulk has its own attributes and geometric detail, and each bulk controls the agent's AI and the collision system. This method focused on a distance approximation, whereas our research focuses on the player's knowledge and aims to generate events.

Brom, Serý, and Poch [16] developed and implemented different LOD simulation algorithms to optimize the space and AI of virtual humans during a storytelling game. It specifies four different levels of detail, and it uses a simplified distance metric to compute the level of detail. Level of detail is applied to navigation, interaction with objects and other agents. The method that we proposed is focused on events.

Osborne, Dickinson, and Patrick [17] presented an optimisation method based on level of detail which simplifies agent behaviours and allows a bigger population in an expansive game world. The LOD is applied to navigation, flocking, and group decisions.

Kistler, Wisner, and André [18] describes an approach to handle LOD and how it is used to reduce the quality of different aspects of an agent during a simulation. LOD is determined based on distance and visibility. It uses 10 different levels of detail, and it is applied to movements, collision avoidance, navigation, and actions.

Riedl and Young [9] introduced Intentional Partial Order Causal Link (IPOCL), a search planning algorithm that creates plot progressions, and reasons about character goals. The IPOCL planner offers a way to represent stories that have relationships between them. An IPOCL plan is a sequence that describes how the world transitioned from its beginning until the goal state. While the IPOCL planner is used in the Indexter model [6], we did not use a planning algorithm in our approach.

3.2.1 Physics Level of Detail

Level of detail in physical simulations is focused on adapting the type of physical model used for different objects by understanding the importance of the object and the efficiency of different models at reacting to the current simulation state.

Redon, Galoppo, and Lin [19] presented an adaptive algorithm to compute the dynamics of articulated bodies in response to the camera distance. It can simplify the dynamics based on the desired number of degrees of freedom, the active joint forces, and the location of external forces.

Debunne, Desbrun, Cani, *et al.* [20] presented an adaptive method for animating dynamic deformable objects, using an automatic space and time level of detail technique along with a strain tensor formulation. The object resolution is determined by a value that indicates when the resolution is too rough. It performed deformations of bodies dynamically. As the object keeps moving and deforms, the test is redesigned to intensify the computational load in the most deformed regions.

Paris, Gerdelan, and OSullivan [21] proposed a collision avoidance method that allows a bigger crowd to be rendered within a simulation. It proposes two collision avoidance models, one focused on performance and the second focused to obtain the best realism. LOD is applied to navigation and collision avoidance. The method that we propose doesn't take into account any collision system.

3.2.2 Behavioural Level of Detail

O'Sullivan, Cassell, Vilhjálmsson, *et al.* [22] described a framework for adaptative level of detail for human interaction (ALOHA), focused on human simulation. It incorporates level of detail in the geometry, motion, complexity gradient for natural behaviour, collision avoidance, and facial expressions using pre-defined scripts.

The ALOHA framework applies LOD to physical, rendering, and behavioural elements. It defined three conversational levels of detail for agents:

- Non-rhetorical behaviour linked to prerecorded dialogue such as gestures.
- Randomly generated non-rhetorical behaviour following basic conversational rules.
- Randomly generated non-rhetorical behaviour that does not conform to communication.

This framework focused on human interaction models and LOD was based on distance, whereas we aim to generate larger scale events.

MacNamee, Dobbyn, Cunningham, *et al.* [23] extended the adaptative level of detail for human interaction (ALOHA) framework by integrating an intelligent agent based on a role-passing technique, allowing the creation of dynamic scenes. This technique adds a behaviour to the agents in order to accomplish a certain role. While this approach does not use LOD precisely, the authors reduce computational resources by setting a specific set of behaviours to roles in agents.

Musse, Kallmann, and Thalmann [24] proposed a method for virtual human agents with different levels of autonomy. They enumerated three different levels of autonomy on a human agent.

- *Guided* indicates the agent must receive specific information about the action is to be performed.
- *Programmed* indicates the agent is able to determine a set of actions and how they can occur.
- *Autonomous* indicates the agent can determine actions and goals for itself.

Although the authors didn't state these levels as conventional levels of detail but "levels of autonomy", they discussed the situations in which different levels of autonomy would be sufficient to execute a human simulation.

Niederberger and Gross [25] present a solution for real-time simulations using a behavioural model in agents. A scheduling algorithm distributes the available computation time rendered across agents in the scene; agents that are closer to the player get more time than invisible or remote agents. The time available per agent affects the behaviour, allowing smooth transitions in the simulation. The LOD is based on proximity and visibility, and it is applied to collision paths, group decisions and path finding. While the approach of this method combines distance and visibility of the agents, we propose an approach where the LOD of events are affected by more and different factors.

3.2.3 Optimizing Simulations

Narain, Golas, Curtis, *et al.* [26] presented a scalable approach for rendering crowds in a simulation. The authors modeled a large scale crowd's behaviour by introducing a variational constraint named unilateral incompressibility. It accelerates agent collision avoidance in dense scenarios. However, a level of detail method was not used during this simulation.

3.3 Summarization Methods

Text summarisation is the process of creating a summary or abstract by choosing only important information from a text. A text is used as an input to the computer and the computer returns only relevant data, and removes unnecessary details from the text. This is known as automatic text summarisation. It usually consists of two approaches: extraction and abstraction. The extraction approach is about selecting each sentence, and assigning it a score, and then picking the ones with better scores. The abstraction approach is about removing unnecessary sentences or words in the text.

The goal of generating event descriptions that are relevant to the player has similarities to some summarisation text techniques. The goal of summarisation is to gather the main ideas using as little text as possible, that is, to gather relevant data into the smallest possible format.

The approach of many summarisation techniques can be remodelled to use any data structure. We can take advantage of it by decoupling the algorithm and input from the desired data.

Kruengkrai and Jaruskulchai [27] proposed an approach for gathering the important sentences from a document to create a summary. This approach uses local and global properties of sentences. Global properties are used a relations between all sentences in the document. Local properties are treated as cluster of relevant words in each sentence. The properties are later combined to create a sentence showing a summary of the given text. The authors claim the results are similar to a commercial text summariser software.

Sarkar [28] presented a method for text summarization which extracts important sentences from a Bengali document. To compare summaries, the authors adopted an automatic summary evaluation metric.

Suanmali, Salim, and Binwahlan [29] presented an extraction-based approach to summarise text. The authors preprocessed the documents in the experiment using: tokenization, sentence segmentation, word stemming, and remove stop word. The method used fuzzy logic to improve the quality of the summary. The results were compared to Microsoft Word 2007's summarizer. The authors claim that the results are more favourable using the fuzzy method.

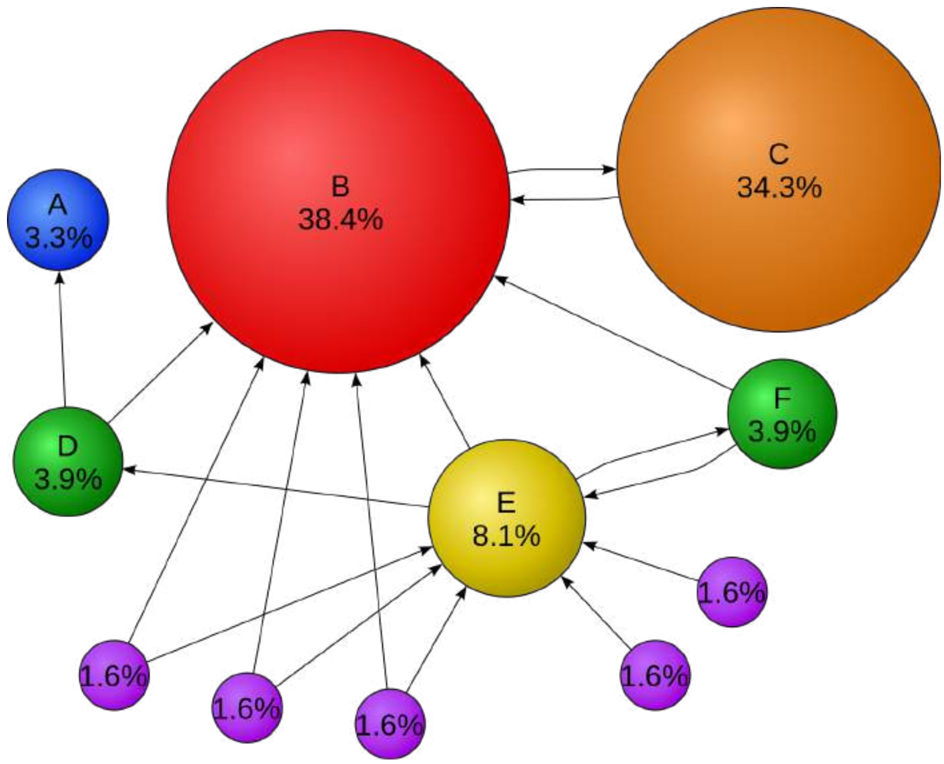

Figure 3.1: Page rank for a simple network. Licensed under Public Domain [34].

Mani [30] developed two different models to extract important features in text. One model is a ridge regressor, and the second a multi-later perceptron. The first step of the method sorted the sentenced by importance, then removed the sentences that had high similarities.

For our work, the central issue with using a summarisation method is that we need to generate a detailed description and then use a summarization technique. For creating upcoming events, the method that we propose aims to avoid generating unnecessary details at all. When the event description is generated, the event should not contain unneeded content.

3.4 Graph-based Ranking Algorithm

A graph-based ranking algorithm decides the importance of each node in a graph by considering the global information of the graph. The algorithm checks the number and the quality of the links connected through the graph to determine how important each node is. Graph-based ranking algorithms have been widely used to retrieve information, and they have proven to be successful in several fields such as web page ranking[31], social networks [32], and text summarisation[33].

There are many approaches to solve this problem. One interesting approach is using a graph-based ranking algorithm such as Google's PageRank original algorithm. Page, Brin, Motwani, *et al.* [35] developed a method to rank websites, later this method was used to develop the popular search engine "Google".

Mihalcea and Tarau [31] introduced "TextRank", a graph-based ranking model for text processing. This method proposes two unsupervised methods to extract keywords and sentences. This method relies on text. The authors investigated different graph-based ranking algorithms, and they evaluated their how well the task of summarisation text was done. They demonstrated that the results obtained with their graph-based ranking method are equally good compared with previously developed summarization methods.

Xu, Bu, Chen, *et al.* [36] propose a scalable graph-based ranking algorithm called Efficient Manifold Ranking (EMR). The authors build an anchor graph instead of a shortest node graph, and designed a new way to represent an adjacency matrix to boost the performance. Results on a large database of images showed that the method was promising to use in retrieval applications.

Erkan and Radev [33] introduced a graph-based approach for computing importance of textual units called "LexRank". The authors defined the importance in word or terms by using a salience value. A connectivity matrix was used based on the words similarities. Results show that a degree-based methods such as LexRank, perform better than other summarisation methods, and that LexRank is very insensitive to the noise in the text.

The properties of events in a simulation, such as location, social, and time, can be represented using graphs. For instance, to represent locations in an event, we can model nodes as places, and edges as travel connections. We describe our approach to doing so in Chapter 4.

Chapter 4

Proposed Approach

This dissertation proposes a method to optimize the generation of upcoming events in a simulation using a level of detail approach. By using the player's knowledge of the simulation and the current world state, this method allows us to estimate the salience of different event properties during generation, which can affect the level of detail of the generated events.

Specifically, our generator creates events according to the current state of the simulation and the player's knowledge. In general, events with many salient properties are created with a high level of detail, meaning that each events properties are fully detailed. Events with few salient properties are created with a low level of detail, meaning that their properties are lightly detailed or omitted completely.

The generator's goal is to generate event descriptions at an appropriate level of detail, while avoiding generating irrelevant event details. This method avoids generating unnecessary information about the world, creating just the information that the player might be interested in. Thus we can reduce the computational requirements of the generator.

Figure 4.1 shows how our event generator can be integrated with a simulation.

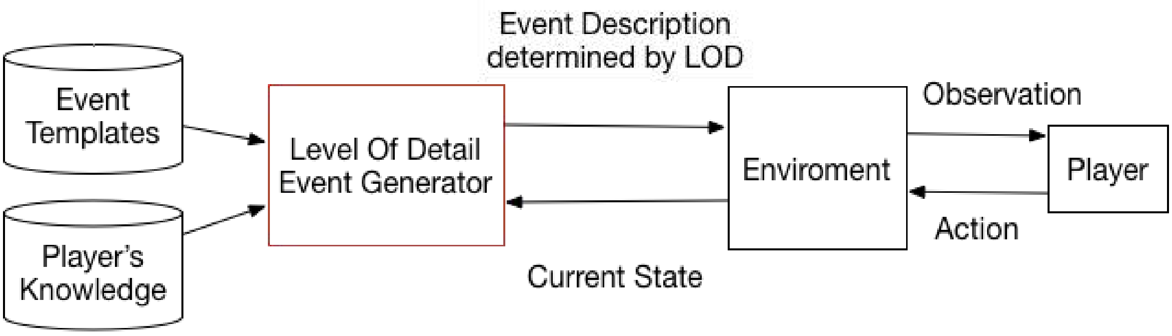

Figure 4.1: Simulation with event generator.

Our generator makes use of event templates, which constrain the values that each event's properties can take on during the generation process. Given an event template, a level of detail evaluation process determines which event properties should have more detail, given the current context. It takes into account the current state, the event properties, and the knowledge of the player. It is used to generate an event description as shown in Figure 4.2.

Our generator uses a model to estimate the salience of each given event template's properties; a separate salience value is estimated for each of the properties (Salience Model). Given these values, our generator decides a level of detail independently for each property (LOD Evaluator). Our generator then creates a description of the event based on these values (Event Factory).

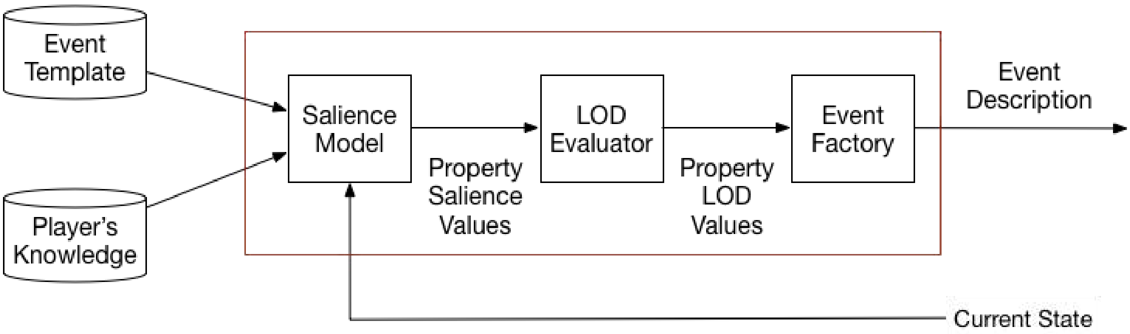

Figure 4.2: Generation process.

For example, if the space property of a template had low salience, it would be assigned a low level of detail and a light description of the location (such as only the title) would be generated. In contrast, if the same property had high salience, it would be assigned a high level of detail and a detailed description would be generated.

4.1 Salience Model

Our method uses a model to estimate the salience of potential upcoming events with respect to the current world state and player's knowledge.

We extend the Event Indexing Situation Model (EISM) proposed by Zwaan, Langston, and Graesser [7] to define our model of the salience of an event. EISM describes a cognitive model of narrative comprehension. It defines a model as a mental representation of a story at a certain point during its consumption. EISM assumes that a narrative is perceived in terms of events or chunks of information, which can be organized under five types of indices: time, protagonist, space, causation, and intention. Our method uses the same indices, but we compute them differently.

Cardona-Rivera, Cassell, Ware, *et al.* [6] estimated the salience of an event in comparison with another by adding the shared indices of the two events. If two events shared an index, the value was set to 1, otherwise it was set to 0. The coefficient of the indices in the sum all had the same weight, that is $w = 0.2$. The equation of their Indexter model is given by:

$$salience(e_i, e_n) = 0.2t_{en} + 0.2s_{en} + 0.2p_{en} + 0.2c_{en} + 0.2i_{en}$$

Where e_i is the event i, e_n is the event n, t_{en} is the time index, s_{en} is the social, p_{en} is the space index, c_{en} is the causation index, and i_{en} is the intention index.

We believe many details are lost by using this approach, as the indices must be shared by both events to have salience of 1. Our method modifies Cardona-Rivera, Cassell, Ware, *et al.*'s [6] approach by computing the salience of each property using a different algorithm for each property.

During the computation of an event's salience, our generator evaluates the salience of each event property independently. The event generator evaluates potential upcoming events and selects the most salient. Properties are evaluated differently depending on how they affect the player or the current state of the world.

The evaluated properties are space, location, protagonist, causation and intention, and each property corresponds to the index with the same name in our salience model.

We believe that event properties that affect the player directly or sooner should be more salient. For instance, an event involving the player should be more salient than an event that does not. Or, an event taking place in the same location as the player should be more salient than another taking place far away. When an event does not affect the player directly, our model prioritizes properties that are somehow known to the player. For example, an event happening in a known place will have more salience than another taking place in an unknown place, regardless if the unknown place is closer to the player.

In the following sections, we describe how the salience for each property is computed.

4.1.1 Social Salience

The social index is a modification of the EISM protagonist index. The EISM protagonist index represents whether the player is involved in the event. We extend this definition with the addition of it also representing how socially close the player is to the agents involved in such event. It captures the relationship between agents and the player.

The social salience of an event depends on how closely the agents in the event are related to the player in a social sense. Events can involve the player of the story; in such cases, the salience for the social index should be the highest possible. Therefore, when the player is involved in the event, a salience of 1 is set. Otherwise, we create a social graph to represent relations between agents, and determine how well the player knows the agents in the event. The social graph contains information about the player's knowledge of the agents and the connections between agents. Our social graph represents agents with nodes and relationships are represented with edges, as shown in Figure 4.3.

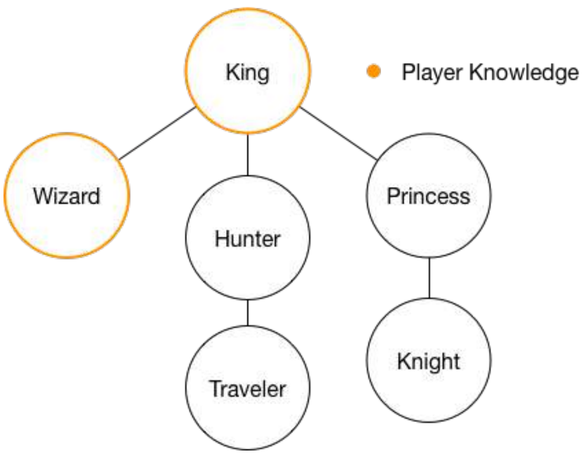

Figure 4.3: Example of a social graph where the agents are represented by nodes and relationships are represented by edges. The orange outlines shows which agents the player currently knows.

The value of social salience for an event is computed using the social graph by checking the relationships between the agents involved and the agents known to the player. A higher salience means the agents in the event are socially close to the player. If the player does not have knowledge of any agent, the social salience is 0.

To compute the social salience we use the following equation:

$$social\_salience(e) = \begin{cases} 1 & \text{if player is involved} \\ 1 - \frac{\frac{1}{social\_distance_e+1}}{n} & \text{otherwise} \end{cases} \tag{4.1}$$

where n is the total number of agents and $social\_distance_e$ represents the minimum number of edges between any agent that the player knows and any agent that is involved in the event e.

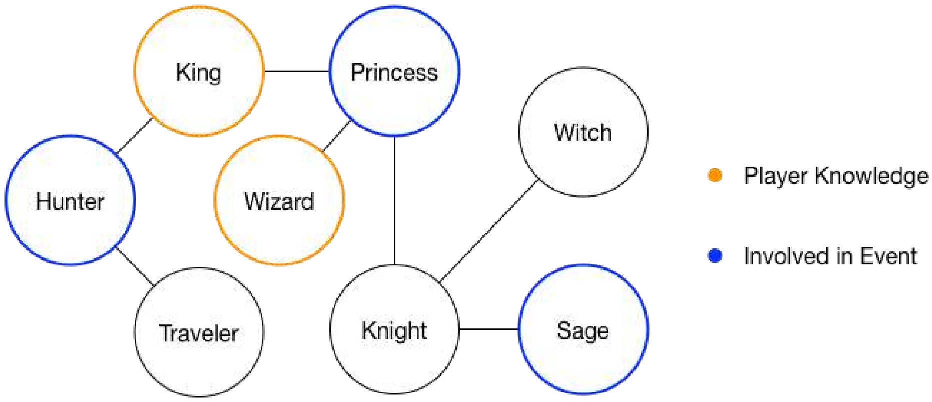

Figure 4.4: Example of a social graph showing the player's knowledge (orange outlines) and the agents involved in a candidate event.

For example, consider the social graph in Figure 4.4, which shows both the player's knowledge of agents (they know the king and the wizard) and the agents that are involved in a candidate event (hunter, princess, and sage). If the player knew an agent that is involved the event, the salience would be 0.875, the maximum salience value this current state can achieve. To get a salience of 1, the player must be involved in the event.

Given the player's knowledge shown in Figure 4.4, the value of $social\_distance_e$ is 1, and thus the social salience of the event is 0.75.

4.1.2 Space Salience

The value of space salience is calculated by checking how close the event location is to the player. This salience is also influenced by the player's knowledge of locations; higher saliences means the player knows either the location of the event or a location nearby. A salience of 1 occurs when the event is held in the same location as the player. Otherwise, a location graph is constructed where nodes are locations and edges represent traversable paths between locations, as shown in Figure 4.5. The figure also shows the player's knowledge of locations in the world, as well as the location of a candidate event. To compute the space salience for this event, we use the following formula.

$$social\_salience(e) = \begin{cases} 1 & \text{if player is involved} \\ 1 - \frac{space\_distance_e + 1}{n} & \text{otherwise} \end{cases} \tag{4.2}$$

Where $space\_distance_e$ is the minimum distance from event e's location to any location that the player knows and n is the total number of locations. Thus, the space salience for an event that is taking place in the *valley* is given by:

$$space\_salience(valley\_event) = 1 - \frac{1+1}{5} = 0.6$$

4.1.3 Time Salience

The value of time salience is related to how relevant the event's time is to the player. Time salience depends on both the event time and the propagation rate. Event time is the moment

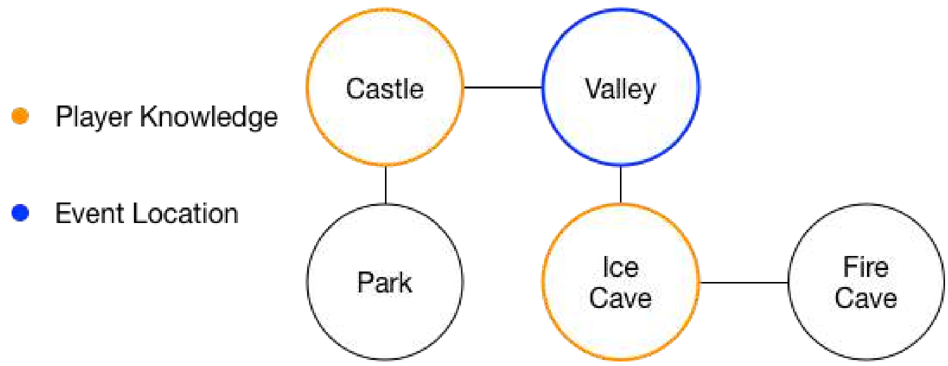

Figure 4.5: Example of a spatial graph for computing space salience. Orange outlines show locations known to the player. The blue outline shows the location of a candidate event.

when the event will take place, and propagation rate is how fast the event's effects spread across the world. A low salience means the event time is not relevant to the player, or the event's effects take too much time to reach the player. This model prefers events that reach the player sooner. For instance, two volcanos erupting at the same time might influence the player differently, especially if one of them is located where the player currently is. The event that reaches the player sooner will be more salient. A salience of 0 means that the player cannot reach the event on time, or the player won't be affected by the event.

To compute time salience, the following formula is applied:

$$reach\_time(e) = \frac{player\_distance_e}{propagation_e}$$

$$time\_salience(e) = \begin{cases} 0 & \text{if } reach\_time(e) > limit \\ \frac{limit - reach\_time(e)}{limit} & \text{otherwise} \end{cases}$$

Where $reach\_time(e)$ represents the time that the effects of the event e will take to reach the player, $player\_distance_e$ represents the distance between the player and event e's location, $propagation_e$ represents the propagation rate of event e, and $limit$ represents the longest time that the player will stay interested in an event. If an event cannot reach the player by this given limit, its time salience is set to 0. As shown in Figure 4.6, assuming both events happened at the same time, event B is more salient to the player as the propagation rate is higher and it will reach the player sooner.

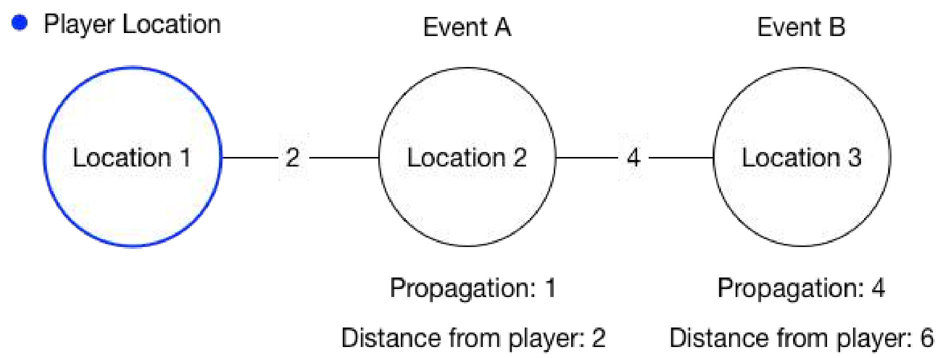

Figure 4.6: Example of time salience when both events happen at the same time, but have different propagation rates.

In this example, we defined $limit = 10$. This means that the player does not care about events after 10 units of time; this can represent days, months, years, etc. Thus, to compute the time salience for event A, we use the formulas:

$$reach\_time(A) = \frac{2}{1} = 2$$
$$time\_salience(A) = \frac{10-2}{10} = 0.8$$

Computing the time salience for event B would give:

$$reach\_time(B) = \frac{6}{4} = 1.5$$
$$time\_salience(B) = \frac{10-1.5}{10} = 0.85$$

The time salience of event B is higher, even though event A occurs closer to the player than event B, because event B's effects reach the player sooner; therefore, event B is more salient to the player.

4.1.4 Causation Salience

The causation index is defined by the EISM. It represents when whether or not two actions are related. Since our model compares the player to new events, the causation index was adapted to our model to indicate whether the player can potentially acquire a new goal by attending this event.

Causation salience indicates when the event motivates the player to achieve a new goal. We define causation salience $c_e = 1$ when the event e offers a new goal to the player and $c_e = 0$ otherwise.

4.1.5 Intention Salience

The intention index is defined by the EISM. It indicates whether or not two actions in two events are part of the same plan. Since our model compares the player to new events, we adapted this index to indicate if the player can achieve a current goal by attending this event.

Intention salience indicates when the event's goal is the same as one of the player's goals. We define intention salience $i_e = 1$ when the event e shares the same goal as the player, and $i_e = 0$ otherwise.

4.1.6 Global Salience

Once it has computed the salience of every property, the generator calculates the total salience for a given event. We set a coefficient value for each index with the same weight, that is $w = 0.2$. Then we can use the following formula:

$$event\_salience(e,p) = 0.2s_e + 0.2t_e + 0.2p_e + 0.2c_e + 0.2i_e \qquad (4.3)$$

Where p is the player, e is the event, and s_e, p_e, t_e, c_e, and i_e are the property salience values for social, space, time, causation, and intention respectively.

The global salience is used to select the most salient event when all the event templates have been considered. Our method then proceeds to evaluate the level of detail of each property in the selected event.

4.2 The Level of Detail Evaluator

The level of detail evaluator converts each event property salience to a high, medium, or low level of detail.

4.2.1 Social Property

There are three levels of detail for the social property:

- **High.** This classification is used when the player is involved. The event factory creates a full detail description of the agents involved.
- **Medium**. This classification is used when a known agent is involved, or the agent involved is very close to a known agent. The event factory creates details of the known agents involved.
- **Low**. This classification is used when the agents involved are unknown. The event factory avoids generating any details.

Example: The knight is fighting the dragon at midnight. The social property in this example is *the knight* and the level of detail is *medium*.

4.2.2 Space Property

There are three levels of detail classifications for the space property:

- **High.** This classification is used when the event is happening in the same location as the player. The event factory creates a full description of the event location.
- **Medium**. This classification is used when the event location is known to the player. The event factory only creates the location name.
- **Low**. This classification is used when the location is not known to the player. The event factory avoids generating any details about the event's location.

Example: The knight is fighting a dragon in the fire cave. The space property in the given example is *fire cave* and the level of detail is *medium*.

4.2.3 Time Property

There are three levels of detail classifications for the time property:

- **High.** This classification is used when event's effects will reach the player quickly. The event factory creates full details of the event's time.
- **Medium**. This classification is used when the event's effects reach the player late, but it is still relevant to the player based on the value of limit (recall section 4.1.3). The event factory creates the event date without the exact time frame.
- **Low**. This classification is used when the event's effects reach the player late, and it is not relevant to the player. The event factory avoids generating any details.

Example: The knight is fighting the dragon at midnight. The time property in the given example is *at midnight* and the level of detail is *medium*.

4.2.4 Causation Property

There are two levels of detail classifications for causation index:

- **High.** This classification is used when the event allows the player to acquire a new goal. The event factory creates a description of the new goal that could be acquired.
- **Low.** This classification is used when the event does not allow the player to acquire a new goal. The event factory avoids generating any details.

Example: A troll has kidnapped the princess, and the king asks for your help to rescue the princess. The causation property in the given example is *Rescue the princess*, since attending this event will allow the player will take on this new goal. The level of detail is *high*.

4.2.5 Intention Property

There are two levels of detail classifications for the intention property:

- **High.** This classification is used when the event satisfies one of the player's goals. The event factory creates the description of the event goal.
- **Low.** This classification is used when the event does not satisfy one of the player's goals. The event factory avoids generating any details.

For example, suppose that one of the player goals is to *fight a dragon*. If an event shares the same goal as the player, the event factory would create a description like: There is a pack of wild creatures around the valley, get ready to fight a dragon. The intention property in this example is *fight a dragon*. The level of detail is *high*.

4.3 The Event Factory

Once we have the level of detail for each property in a given event template, the event factory creates an event description according to the given levels of detail.

The event factory makes use of a tool called "Tracery" to create event descriptions [37]. Tracery is a story-grammar generation library and uses grammar rules to create generative stories. The event factory specifies one grammar rule per index, and each grammar rule is more detailed depending on the level of detail. Each grammar rule can be composed of different grammar objects. For example, simple grammar example would be:

```
{
    "name": ["Arjun","Yuuma","Darcy"],
    "animal": ["unicorn","raven","sparrow"],
    "story": ["#name# traveled with her pet #animal#"]
}
```

A possible output for this grammar is: *Darcy traveled with her pet unicorn*.

The grammar used in our event factory is:

```
{
    "location": generateLocationGrammar(locationLOD),
    "time": generateTimeGrammar(timeLOD),
    "social": generateSocialGrammar(socialLOD),
    "intention": generateIntentionGrammar(intentionLOD),
    "causation": generateCausationGrammar(causationLOD),
    "event": ["${event.title} #location# #time# #social# #intention# #causation#"]
}
```

4.3.1 Social Grammar

The grammar for the social property is:

- **High.**

```
{
    "social": ["You are involved in the event, ${agents} are attending as well.",
      "You are taking part in this, the following ${agents} are attending too.",
      "The ${agents} are attending, you have been called to attend it too."]
}
```

- **Medium.**

```
{
    "social": ["The ${known_agents} are attending."]
}
```

- **Low.** The event factory avoids creating any details at this LOD.

4.3.2 Space Grammar

The grammar for the space property is:

- **High.**

```
{
    "space": ["It is taking place at ${location.name}, ${location.description}."]
}
```

- **Medium.**

```
{
    "space": ["It will take place at ${location.name}."]
}
```

- **Low.** The event factory avoids creating any details at this LOD.

4.3.3 Time Grammar

The grammar for the time property is:

- **High.**

```
{
    "timeFiller": ["noon", "dawn", "dusk", "sunset", "midnight", "sunrise"],
    "time": ["It will take place in ${time} days from now at #timeFiller#.",
    "It will take place today at #timeFiller#."]
}
```

- **Medium.**

```
{
    "time": ["It is ${time} days from now.", "It is happening now."]
}
```

- **Low.** The event factory avoids creating any details at this LOD.

4.3.4 Causation Grammar

The grammar for the causation property is:

- **High.**

```
{
    "causation": ['You will be offered to ${new_goal} if you attend it.']
}
```

- **Low.** The event factory avoids creating any details at this LOD.

4.3.5 Intention Grammar

The grammar for the intention property is:

- **High.**

```
{
    "intention": ["You might be able to ${goal}.", "Get ready to ${goal}."]
}
```

- **Low.** The event factory avoids creating any details at this LOD.

4.4 Event Generator in Action

An execution of the method involves instantiating an event template, which are specified as a set of high level event properties. Each template is evaluated and the most salient to the player is selected and formatted to be displayed in the simulation. For example, consider the event templates shown in Table 4.1. For the sake of simplicity, assume that all the events would occur at the same time. We can compute the salience of each potential events and create an event from the most salient event template. The player's location is the *castle*. The player's goals are to *Talk to the king* and *Recover the treasure*. The player has knowledge of the following locations: *castle*, *dungeon*, and *valley*. The only agent the player knows is the *king*. To compute the time salience, we set the *limit* time to 3 time units.

The locations and neighbours are shown in Figure 4.7. The agents and social connections are shown in Figure 4.8.

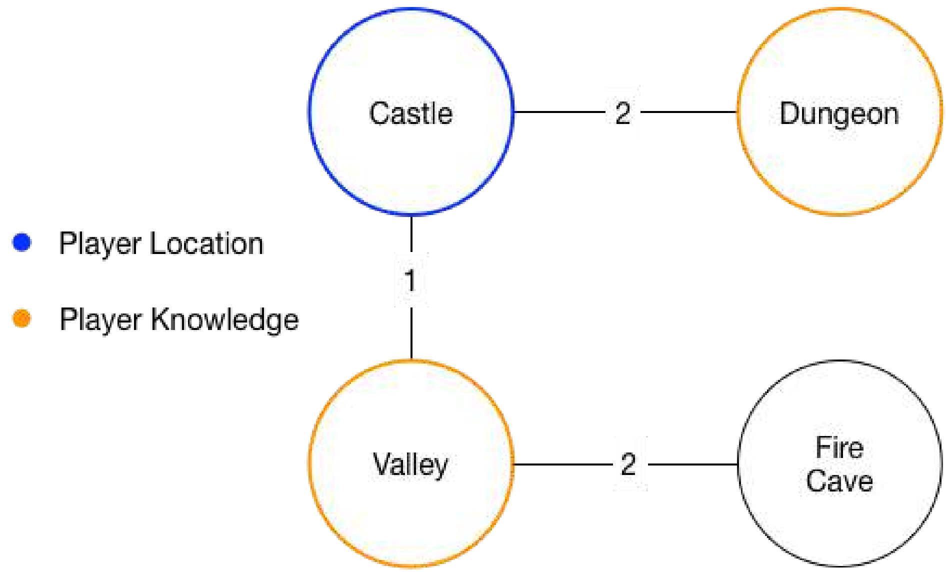

Figure 4.7: Locations graph. The player's knowledge is shown in orange, and the player's location in blue. Edges are marked with the distances between connected nodes.

	Event 1	**Event 2**	**Event 3**
Title	The orcs are gathering	A wild dragon appeared	A troll kidnapped the princess
Player	No	Yes	No
Agents	Traveler	Hunter	King, Traveler
Locations	Valley	Fire Cave	Castle
Propagation	1	2	4
Goal	Spy on the orcs	Kill the dragon	Talk to the king
Cause	Talk to the king	Recover the treasure	Rescue the princess

Table 4.1: Event Templates. The row labeled Player indicates if the player is involved in the events. The row labeled location indicates the events' location. The row labeled Propagation indicates the propagation time of the events. The row labeled Goal indicate the events' goal. The row labeled Cause indicate the causation property of the events.

The saliences of the properties have been computed given the player's knowledge and the world state. These and the events' global salience can be seen in Table 4.2.

The event factory would create the following event descriptions for the levels of detail given in Table 4.2. Annotations appear as emphasized text.

Event 1: The orcs are gathering *(event title)*. It will take place at the valley *(space, medium LOD)*. It is happening today *(time, medium LOD)*.

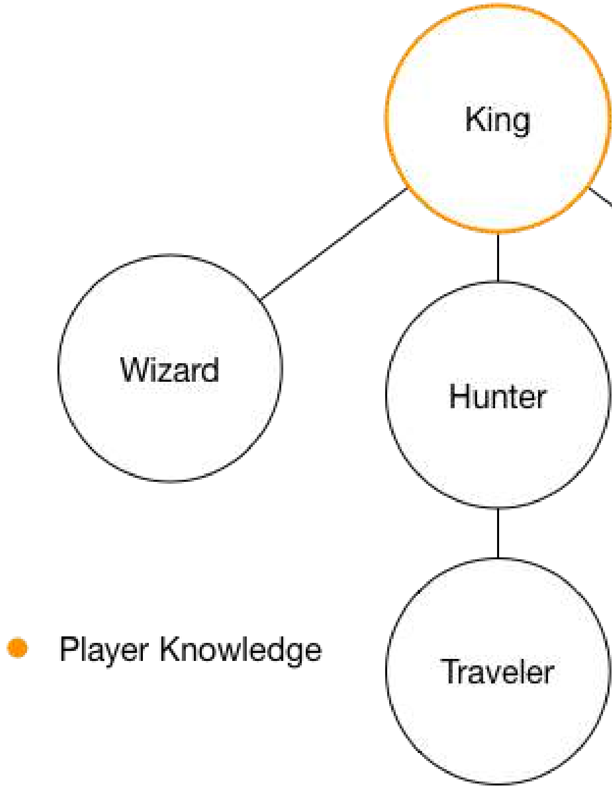

Figure 4.8: Social graph. The player's knowledge is shown in orange.

Property	Event 1		Event 2		Event 3	
	Salience	LOD	Salience	LOD	Salience	LOD
Social	0.25	L	1.0	H	0.875	M
Space	0.75	M	0.5	L	1.0	H
Time	0.66	M	0.5	L	1.0	H
Intention	0.0	L	0.0	L	1.0	H
Causation	0.0	L	1.0	H	1.0	H
Global	0.33	N/A	0.6	N/A	0.975	N/A

Table 4.2: Events salience and level of detail (L=Low, M=Medium, H=High).

Event 2: A wild dragon appeared *(event title)*. You are involved into this, the hunter is attending as well *(social, high LOD)*. You will be offered to recover the treasure if you attend it *(causation, high LOD)*.

Event 3: A troll kidnapped the princess *(event title)*. It is taking place at castle, it is happening outside the gates *(space, high LOD)*. The king is attending *(social, medium LOD)*. It will take place today at noon *(time, high LOD)*. Get ready to talk to the King *(intention, high LOD)*. You will be offered to rescue the princess if you attend it *(causation, high LOD)*.

The user study website is described in the appendix A.

Chapter 5

Evaluation

Given an event template and the player's knowledge, for each salience index, can we identify the "right" level of detail to use when generating the event for the current player? Since our generator aims to use higher levels of detail only for the more salient properties of an event, finding the "right" levels of detail amounts to correctly predicting the salience of each of the event's properties.

To evaluate our approach, we conducted a pilot study to assess which properties of potential future events would be more or less interesting to players, given their knowledge about the game's world. We created two simple game worlds as testbeds, defining the world state and the player's knowledge of the agents, locations, and goals as shown Table 5.1. The player's knowledge and world state were listed in a sentence during the study, and participants were not able to see the social or space graphs.

	World 1	**World 2**
Player Location	Castle	Husavik
Player's Goal	Talk to the king	Capture an elf
Known Locations	Castle, Dungeon, and Valley	Husavik, Keflavik, and Borganes.
Known Agents	King	President

Table 5.1: Game worlds used to test our approach.

The social graphs for both worlds are shown in Figure 5.2. The location graphs for both worlds are shown in Figure 5.1.

For each event property and for each level of detail that our method would consider for that property, we proposed an event to each participant such that our generator would use the given level of detail for the given property. Then, we asked the players to rate how interesting the event seemed with respect to that event property. We asked one such question per property's level of detail, for each of our testbed world states. We asked questions related to the current world state and player's knowledge such as: *will you be interested in a new event if it is located at the castle?*, *will you be interested in the new event if the king is involved?*, and *will you be interested in the new event if it is happening today?*, etc. Participants could respond to each question by selecting "Very much interested", "Somewhat interested", and "Not at all interested". The worlds have the same structure but different context. More specifically, building the location and social graphs in both worlds will result the same graph structure, but the characters and locations are different. The reason of doing so is to help us reveal potential biases caused by different content.

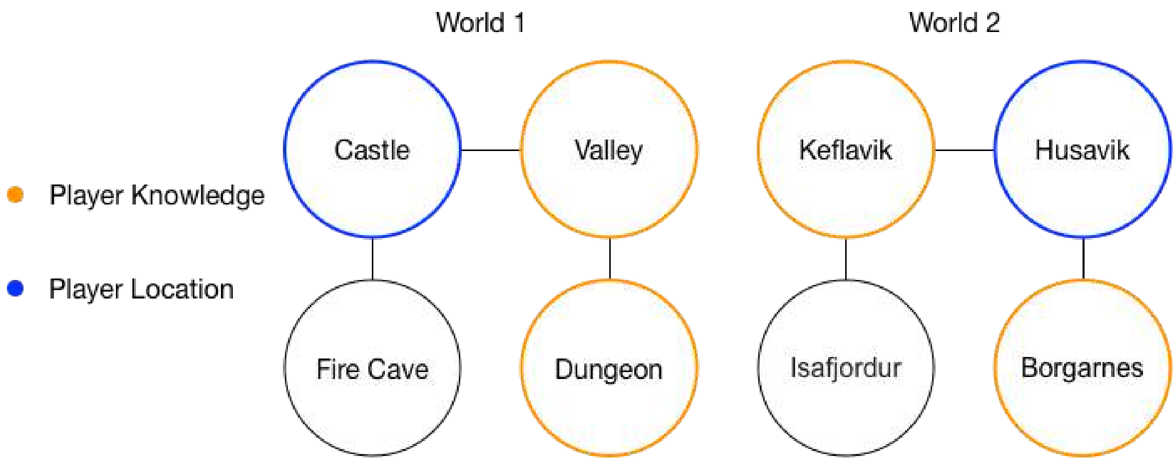

Figure 5.1: Location graphs for both worlds.

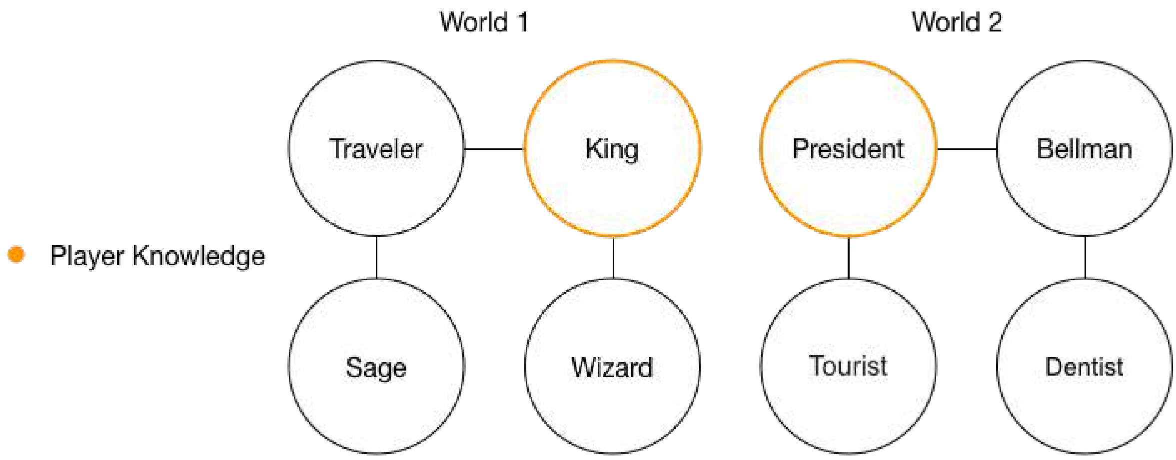

Figure 5.2: Social graphs for both worlds.

We recruited players by sharing a weblink to our survey using social media. Most of them had a computer science background. A total of 48 players evaluated our method. The players were mainly from Iceland and Mexico. The text of the survey can be found in Appendix A.

5.1 Data and Results

We collected data for each event property at each of the levels of detail that our method considers for that property. Each event property can contain a different number of levels of detail, as we described in section 4.2.

5.1.1 Social Salience

We asked the players how interested they would be in three new events, where each event involved one of the three different agents that would cause our method to use a high, medium, or low level of detail, respectively.

The medium level of detail is used when the social salience is 0.75, meaning the agent involved is known to the player. A low level of detail is used when the social salience is less than 0.75, meaning the agent involved is unknown.

For example, our method would use a medium level of detail in a new event involving the king (but not the player) in World 1, because the player knows that agent.

- **High**. This level of detail is used when the player is involved. The survey showed the results given in Table 5.2.

Interest	World 1	World 2	Average
Very much	77.1%	77.1%	77.1%
Somewhat	18.7%	22.9%	20.8%
Not at all	4.2%	0.0%	2.1%

Table 5.2: Results when the player is involved in the event (High LOD).

- **Medium**. This level of detail is used when an agent is known to the player, or an agent involved is very close to a known agent. Table 5.3 displays the results.

Interest	World 1	World 2	Average
Very much	81.2%	64.5%	72.85%
Somewhat	18.8%	25.0%	21.9%
Not at all	0.0%	10.4%	5.2%

Table 5.3: Results when a known agent is involved in the event (Medium LOD).

- **Low**. The lowest detail is used when the agents involved are unknown to the player. The survey showed a lack of interest in the event if an unknown agent is involved. Table 5.4 displays the results.

Interest	World 1	World 2	Average
Very much	18.8%	6.2%	12.5%
Somewhat	37.4%	31.3%	34,3%
Not at all	43.8%	62.5%	53,1%

Table 5.4: Results when a unknown agent is involved in the event (Low LOD).

Table 5.2 shows that the interest is high when the player is involved. However, Table 5.3 shows a possible bias, it seems that the involvement of important agents or agents with a high rank in the event can catch the interest of players. In this case, *the king* caused more interest to the players than their own direct involvement. Table 5.4 shows the interest for the events reduced when the player was not involved.

5.1.2 Space Salience

There are three levels of detail for this property. We asked the players how interested they would be in a new event for each of the three levels of detail for the space property.

The medium level of detail is used when the space salience is 0.75, meaning the event location is known to the player. A low level of detail is used when the space salience is less than 0.75, meaning the event location is unknown.

Interest	World 1	World 2	Average
Very much	66.7%	56.2%	61.4%
Somewhat	31.2%	31.2%	31.2%
Not at all	2.1%	12.5%	7.3%

Table 5.5: Results when the event is held where the player currently is (High LOD).

- **High**. This level of detail is used when the event is held in the same location as the player. The survey showed the results given in Table 5.5.
- **Medium**. This level of detail is used when the event is held in a location that the player has knowledge of. The survey showed the results given in Table 5.6.

Interest	World 1	World 2	Average
Very much	45.8 %	29.1%	37.4 %
Somewhat	35.4%	54.2%	44.8%
Not at all	18.8%	16.7%	17.7%

Table 5.6: Results when the event is held in a location known to the player (Medium LOD).

- **Low**. This level of detail is used when the event is held in a location that is unknown to the player. The survey showed the results give in Table 5.7.

Interest	World 1	World 2	Average
Very much	31.3%	14.6%	22.9%
Somewhat	41.6%	45.8%	43.7%
Not at all	27.1%	39.6%	32,8%

Table 5.7: Results when the event is held in an location unknown to the player (Low LOD).

Table 5.5 shows that the interest increased when the event takes place in the same location as the player. Table 5.6 shows that players are somewhat interested in the event when it is located in a known location, however, it seems there is a possible bias when the event is taking place in an interesting sounding place, like the fire cave. Table 5.7 shows that the interest in the events reduced when the location was not known to the player in both worlds, however, it seems like the players are always somewhat interested in the event location.

5.1.3 Time Salience

There are three levels of detail for this property. We asked the participants how interested they would be in new events involving three different levels of detail: when the event is about to happen, when the event is happening in a few days, and when the event will only happen after a long time.

- **High**. This level of detail is used when the event is happening right away. The survey showed the results given in Table 5.8.
- **Medium**. This level of detail is used when the event is happening in a few days. The survey showed the results given in Table 5.9.

Interest	World 1	World 2	Average
Very much	60.4%	62.5%	61,4%
Somewhat	35.4%	31.3%	33,3%
Not at all	4.2%	6.2%	5,2%

Table 5.8: Results when the event is happening right away (High LOD).

Interest	World 1	World 2	Average
Very much	12.5%	8.3%	10.4%
Somewhat	64.6%	54.1%	59.3%
Not at all	22.9%	37.6%	30.2%

Table 5.9: Results when the event is happening in a few days (Medium LOD).

- **Low**. This level of detail is used when the event is happening after a long time. The survey showed the results given in Table 5.10.

Interest	World 1	World 2	Average
Very much	8.3%	12.5%	10.4%
Somewhat	60.4%	56.2%	58.3%
Not at all	31.3%	31.3%	31.3%

Table 5.10: Results when the event will only happen a long time from now (Low LOD).

Table 5.8 shows a high interest when the event is happening right away. Table 5.9 shows that players are somewhat interested when the event is happening in a few days. Table 5.10 shows that the interest reduced when the event will only happen a long time from now. Even so, the low and medium level of detail results are similar; it seems like players are always somewhat interested in the event time.

5.1.4 Causation Salience

There are two levels of detail for this property. We asked the participants how interested they would be in the new event if it would motivate them to achieve new goals by attending the event.

- **High**. This level of detail is used when the new event motivate players to achieve new goals by attending the event. The survey showed the results given in Table 5.11.

Interest	World 1	World 2	Average
Very much	79.2%	72.9%	76.0%
Somewhat	16.7%	22.9%	19.8%
Not at all	4.2%	4.2%	4.2%

Table 5.11: Results when the event motivates the player to achieve new goals (High LOD).

- **Low**. This level of detail is used when the new event does not motivate players to achieve new goals by attending the event. The survey showed the results given in Table 5.12.

Interest	World 1	World 2	Average
Very much	4.2%	4.3%	4.25%
Somewhat	43.7%	52.2%	47.9%
Not at all	52.1%	43.5%	47.8%

Table 5.12: Results when the event does not motivate players to achieve new goals (Low LOD).

Table 5.11 shows a very high interest when the event motivates the player to achieve new goals. Table 5.12 shows that the interest reduced when the event does not motivate players to achieve new goals. However, it seems like half the players are somewhat interested in the event, regardless of whether the event does or does not motivate players to achieve a new goal.

5.1.5 Intention Salience

There are two levels of detail for the intention property. We asked the participants how interested they would be in a new event if they could achieve their current goals by attending the event.

- **High**. This level of detail is used when players can achieve a current goal by attending the event. The survey showed the results given in Table 5.13.

Interest	World 1	World 2	Average
Very much	79.2%	88.6%	83.9%
Somewhat	18.8%	6.8%	12.8%
Not at all	2.1%	4.5%	3.3%

Table 5.13: Results when the player can achieve their goals by attending the event (High LOD).

- **Low**. This level of detail is used when players cannot achieve a current goal by attending the event. The survey showed the results given in Table 5.14.

Interest	World 1	World 2	Average
Very much	23.0%	12.5%	17.7%
Somewhat	35.4%	39.5%	37.4%
Not at all	41.6%	48.0%	44.8%

Table 5.14: Results when the player cannot achieve any goals by attending the event (Low LOD).

Table 5.13 shows a very high interest when the player can achieve his goals by attending the event. Table 5.14 shows that the interest reduced when the player cannot achieve any goals by attending the event.

5.2 Accuracy and Precision

Our model is trying to solve a prediction problem: given a particular event property, we predicted how much interest a property can produce in the players. Our system effectively classifies each property as *not interesting*, *somewhat interesting*, or *very interesting* and then maps each of those values to the appropriate level of detail (*low*, *medium*, or *high*, respectively). We computed accuracy and precision for each of the 5 event properties (Social, Time, Space, Intention, Causation), for each world set. We used confusion matrices to compute the accuracy and the precision of our model. We are interested in the values that our model maps to each interest classification, which correspond to the diagonal on each confusion table and are marked in bold. In our pilot study, we gathered a total of 48 responses from participants. Since each participant answered questions about two or three levels of detail for each type of salience, the total number of data points in the confusion matrices is a multiple of 48.

We are interested in knowing the accuracy and precision. The accuracy will tell us how often our classifier is correct overall. The precision will tell us how often our model predicts a level of detail correctly.

5.2.1 Social Salience

The confusion matrix for social salience in World 1 is displayed in Table 5.15.

$n = 144$	**High**	**Medium**	**Low**	
Very interesting	**37**	39	9	76
Somewhat interesting	9	**9**	18	36
Not interesting	2	0	**21**	23
	48	48	48	

Table 5.15: Confusion matrix for social salience in World 1.

The confusion matrix for World 2 is displayed in Table 5.16.

$n = 144$	**High**	**Medium**	**Low**	
Very interesting	**37**	31	3	71
Somewhat interesting	11	**12**	15	38
Not interesting	0	5	**30**	35
	48	48	48	

Table 5.16: Confusion matrix for social salience in World 2.

To compute the accuracy in both worlds, we use the following equations:

$$accuracy_1 = \frac{37 + 9 + 21}{144} = 46.5\% \tag{5.1}$$

$$accuracy_2 = \frac{37 + 12 + 30}{144} = 54.8\% \tag{5.2}$$

The accuracy for the social property in World 1 is 46.5%. The accuracy for world 2 is 54.8%. The precision of our method for predicting social salience for both worlds can be seen in Table 5.17.

	High	**Medium**	**Low**
World 1	$\frac{37}{48} = 77.0\%$	$\frac{9}{48} = 18.7\%$	$\frac{21}{48} = 43.7\%$
World 2	$\frac{37}{48} = 77.0\%$	$\frac{12}{48} = 25.0\%$	$\frac{30}{48} = 62.5\%$

Table 5.17: Precision for social salience for both worlds.

5.2.2 Space Salience

The confusion matrix for space salience in World 1 is displayed in Table 5.18.

$n = 144$	**High**	**Medium**	**Low**	
Very interesting	**32**	22	15	69
Somewhat interesting	15	**17**	20	52
Not interesting	1	9	**13**	23
	48	48	48	

Table 5.18: Confusion matrix for space salience in World 1.

The confusion matrix for World 2 is displayed in Table 5.19.

$n = 144$	**High**	**Medium**	**Low**	
Very interesting	**27**	14	7	53
Somewhat interesting	15	**26**	22	63
Not interesting	6	8	**19**	28
	48	48	48	

Table 5.19: Confusion matrix for space salience in World 2.

To compute our method's accuracy for space salience in each world, we use the following equations:

$$accuracy_1 = \frac{32 + 17 + 13}{144} = 43.1\% \tag{5.3}$$

$$accuracy_2 = \frac{27 + 26 + 19}{144} = 50.0\% \tag{5.4}$$

The accuracy for space salience using our model in World 1 is 43.1%. The accuracy for World 2 is 50.0%. The precision of our method for predicting space salience in both worlds can be seen in Table 5.20.

	High	**Medium**	**Low**
World 1	$\frac{32}{48} = 66.7\%$	$\frac{17}{48} = 35.4\%$	$\frac{13}{48} = 27.1\%$
World 2	$\frac{27}{48} = 56.3\%$	$\frac{26}{48} = 54.2\%$	$\frac{19}{48} = 40.0\%$

Table 5.20: Precision for space salience for both worlds.

$n = 144$	**High**	**Medium**	**Low**	
Very interesting	**29**	6	4	39
Somewhat interesting	17	**31**	29	77
Not interesting	2	11	**15**	28
	48	48	48	

Table 5.21: Confusion matrix for time salience in World 1.

$n = 144$	**High**	**Medium**	**Low**	
Very interesting	**30**	4	6	40
Somewhat interesting	15	**26**	27	68
Not interesting	3	18	**15**	36
	48	48	48	

Table 5.22: Confusion matrix for time salience in World 2.

5.2.3 Time Salience

The confusion matrix for time salience in World 1 is displayed in Table 5.21.

The confusion matrix for World 2 is displayed in Table 5.22.

To compute accuracy in World 1, we use the following equations:

$$accuracy_1 = \frac{29 + 31 + 15}{144} = 52.1\% \tag{5.5}$$

$$accuracy_2 = \frac{30 + 26 + 15}{144} = 49.3\% \tag{5.6}$$

The accuracy for time salience using our model in World 1 is 52.1%. The accuracy for World 2 is 49.3%:

The precision of our method for time salience for both worlds be seen in Table 5.23.

	High	**Medium**	**Low**
World 1	$\frac{29}{48} = 60.4\%$	$\frac{31}{48} = 64.5\%$	$\frac{15}{48} = 31.2\%$
World 2	$\frac{30}{48} = 62.5\%$	$\frac{26}{48} = 54.1\%$	$\frac{15}{48} = 31.2\%$

Table 5.23: Precision for time salience for both worlds.

5.2.4 Causation Salience

The confusion matrix for causation salience in World 1 is displayed in Table 5.24.

$n = 96$	**High**	**Low**	
Very interesting	**38**	2	40
Somewhat interesting	8	21	29
Not interesting	2	**25**	27
	48	48	

Table 5.24: Confusion matrix for the causation property in World 1.

$n = 96$	**High**	**Low**	
Very interesting	**35**	2	37
Somewhat interesting	11	25	36
Not interesting	2	**21**	23
	48	48	

Table 5.25: Confusion matrix for the causation property in World 2.

The confusion matrix for World 2 is displayed in Table 5.25.
To compute accuracy in World 1, we use the following equations:

$$accuracy_1 = \frac{38 + 25}{96} = 65.6\% \tag{5.7}$$

The accuracy for causation salience in World 1 is 65.6%.

$$accuracy_2 = \frac{35 + 21}{96} = 58.3\% \tag{5.8}$$

The accuracy for World 2 is 58.3%.
The precision of our method for both worlds can be seen in Table 5.26.

	High	**Low**
World 1	$\frac{38}{48} = 79.1\%$	$\frac{25}{48} = 52.1\%$
World 2	$\frac{35}{48} = 72.9\%$	$\frac{21}{48} = 43.7\%$

Table 5.26: Precision for causation salience for both worlds.

5.2.5 Intention Salience

The confusion matrix for intention salience in World 1 is displayed in Table 5.27.

$n = 96$	**High**	**Low**	
Very interesting	**38**	11	49
Somewhat interesting	9	17	26
Not interesting	1	**20**	21
	48	48	

Table 5.27: Confusion matrix for intention salience in World 1.

The confusion matrix for World 2 is displayed in Table 5.28.

$n = 96$	**High**	**Low**	
Very interesting	**43**	6	49
Somewhat interesting	3	19	22
Not interesting	2	**23**	25
	48	48	

Table 5.28: Confusion matrix for intention salience in World 2.

To compute accuracy in World 1, we use the following equations:

$$accuracy_1 = \frac{38 + 20}{96} = 60.4\% \tag{5.9}$$

The accuracy for the intention property used by our model in World 1 is 60.4%.

$$accuracy_2 = \frac{43 + 23}{96} = 68.7\% \tag{5.10}$$

The accuracy in World 2 is 68.7%:

The precision of our method for intention salience for both worlds can be seen in Table 5.29.

	High	**Low**
World 1	$\frac{38}{48} = 79.1\%$	$\frac{20}{48} = 41.6\%$
World 2	$\frac{43}{48} = 89.5\%$	$\frac{23}{48} = 47.9\%$

Table 5.29: Precision for intention salience for both worlds.

Chapter 6

Discussion

The model we have presented in this dissertation focuses on generating upcoming events in a world driven by a simulation. We avoided generating different details of an event according to an estimate of the player's interest in each event property. To evaluate our work, we measured how accurate our model is at estimating the player's interest.

The results for social salience showed accuracies of 46.5% for World 1 and 54.8% for World 2. The largest numbers of players showed showed very much interest in the events where our method would have used a high level of detail (37 in World 1 and 37 in World 2), and not much interest when our method would have used a low level of detail (21 in World 1 and 30 in World 2). These are both positive results. Our results for the medium level of detail are weaker, but they may have been biased by the agent names we used. Specifically, having an important agent or someone with a high authority (e.g., king or president) may have raised levels of interest among players, even though they did not know those agents.

The results for space salience showed accuracies of 43.1% for World 1 and 50.0% for World 2. The results showed that the largest number of players had high interest in new events where our method would have used a high level of detail (32 and 27 for World 1 and World 2 respectively); this is a positive result. However, we suspect that places with inherently interesting names can also influence the player's interests. We used a compelling name ("fire cave") in World 1 and a random name in World 2, and players were more interested in an event taking place with the most interesting name.

The results for time salience showed an accuracy of 52.1% for World 1 and 49.3% for World 2. The evaluation showed positive precision results for the high (29 and 30 players in World 1 and World 2 respectively) and medium (30 and 26 players in World 1 and World 2 respectively) levels of detail. Results are very similar in both worlds, they showed the same time-interest pattern. We are interested in adjusting the model to put more weight on this type of salience. The low level of detail used in our method needs improvement, however, since most of the players were somewhat interested in the event even when it would only happen after a long time.

The results for intention salience showed accuracies of 60.4% for World 1 and 68.7% for World 2. The largest amount of players had a high interest for this property when our method used a high level of detail (38 and 43 in World 1 and World 2 respectively). The low level of detail used in our method needs improvement, however, since some players found events to be interesting even when they did not share the player's goal. Anecdotally, it seemed as though player attention increased when the goal of the event sounded more fun or challenging.

The results for causation salience showed an accuracy of 65.6% for World 1 and 58.3% for World 2. The largest numbers of players showed very much interest in the events where our method would have used a high level of detail (38 and 25 in World 1 and World 2 respectively), and less interest when our method would have used a low level of detail (25 in World 1 and 21 in World 2). However, some players are still interested in events for which our method would have used a low level of detail. We suspect that this index has less impact on the player's interest than other indices.

The results for the five saliences values showed accuracies much better than what a uniform random predictor would produce (33.3% and 50.0% when using as two and three levels of detail, respectively). However, the social and space results are not good if we compare them to a random predictor that considers the bias observed in the data. For example, a predictor that always predicts high detail would obtain accuracies near 50% for social salience, using the data from our study.

The results showed that players seemed somewhat interested even when our method predicted a low level of detail; it seems like we should always display something at this LOD. In our model, we omit details using a low level of detail. Given these results, we are interested in finding an approach in which details are generated on each level of detail.

The event factory used in our approach generates a simple event description text, which is not computationally expensive. We thus do not gain much computational savings with our simple event factory, but a more complicated one could gain a lot. For example, a complex simulation that creates new events might involve rendering 3D objects, geometric instances, locations, agents, etc. In such a case, our method has the potential to reduce the computation load significantly allowing a larger game world to be simulated.

6.1 Limitations

A few limitations arise when using our event generator's model. This model focused on a single player, and so we used the protagonist as the main player of the simulation. However, simulations can involve other important agents such the antagonist of the story, or video games can be multiplayer which can heavily influence the importance of the social aspects of different events.

The social property can be biased by the apparent importance of agents during the simulation, and our model does not take into account important agents that can influence the interest of players. We are interested in learning how important agents can change the interest of players during the simulation.

The space property can be very complex. For the sake of simplicity, our model only considers each location as a single entity. More complex locations could contain locations within locations such as rooms in buildings, or places within cities, etc. We want to improve our method by allowing these more complex representations in our model.

The levels of detail used by our method are determined by the salience of each event property. The model needs to know a numeric range of each level of detail in advance. An author must define these thresholds for each level of detail in each index.

6.2 Future Work

We want to improve the overall accuracy of our model and improve the precision on each level of detail used in our model. This challenge can be approached in several ways.

Simulations can contain one or more players. We would like to determine if extending this model is needed to handle more players. Important agents in the simulations can influence the interest of players. We want to know the qualities of agents that makes them more interesting to the players.

Interesting name locations can affect the player's interest, and future work could determine how location names affect space-related salience. Also, we want to improve our method by allowing complex location representations in our model.

Space and social saliences can be computed in a better way. For instance, we use the total number of nodes in the graph when creating the space or social graphs. This might be overkill if the simulation has a big number of locations and agents. A possible way to improve it is to use the width of the graph (the "longest shortest distance") instead of the total number of nodes.

The time index uses a time limit variable that determines the maximum time that an event will be relevant to the player. After this time limit, our method will use a low level of detail, and this time limit is set by the author. We want to find a better way to determine the time limit used by the low level of detail, or a way to avoid using it entirely.

Our current method uses two levels of detail for the intention and causation properties. They are modelled as binary values, similarly to prior work [6]. The player and the upcoming event are either connected by these indices or not. This method uses a salience of value 1 or 0, which then is mapped to a high or low level of detail, respectively, to represent this binary relation. We are interested in finding a way to compute the salience using values inside the range of $[0, 1]$. We also want to determine if more levels of detail can be used for both kinds of indices.

We are interested in finding a dynamic way to compute each threshold for each level of detail used by each property. Thresholds are currently set by the author.

Our model uses the same weight coefficient for every index when computing an event's overall salience. The Indexter model [6] and EISM [8] did not specify which indices should have more influence than others. We want to study which indices are more dominant than others, and determine a more realistic weight coefficient for each index in our model.

Chapter 7

Conclusion

Level of detail is a method that involves reducing the amount of detail that is generated for some entity. Using level of detail in a simulation has the potential to help improve performance, reducing computational load. Creating new events in a simulation can be resource intensive. We aim to optimize the level of detail of generated events by understanding when their properties are more or less important to players during the simulation.

We proposed an event generator that aims to optimize the level of detail with which events in the world are generated by using an estimate of the players knowledge to estimate their interest in different aspects of the world. Doing so can avoid generating unnecessary details, enabling larger simulations to be run. Our generator uses a model to predict the salience of events based on an estimate of the player's knowledge during the simulation and the current world state.

Our model is based on the Indexter model presented by Cardona-Rivera, Cassell, Ware, *et al.* [6]. The Indexter model predicts the salience between two events, whereas this work predicts the salience between the player and a potential upcoming event. Our generator relies on a cognitive model of narrative comprehension called the Event-Indexing Situation Model (EISM) developed by Zwaan, Langston, and Graesser [7]. EISM defines a cognitive representation at a certain point during the simulation. Each event is represented by the indices: time, protagonist, space, causation, and intention. Our model extends and adapts these indices to generate new events according to the world state and the player's knowledge.

Our generator computes the salience for each property event (one per index), and depending on this value, chooses a level of detail for the event property. There are three levels of detail for the space, location, and time indices ("high", "medium", and "low"), and two levels for the intention and causation index ("high" and "low").

We conducted a pilot experiment to evaluate how well our model predicted the ideal levels of detail. The results showed that the average accuracies of our model for the social, space, intention, causation properties were 50.6%, 46.5%, 50.6%, 64.5%, and 61.9% respectively. We discovered the accuracy in some indices can be biased easily by different factors such as interesting names in agents or locations.

In the future, we aim to improve our model's accuracy and precision by addressing its current limitations.

Bibliography

[1] L. Bergman, H. Fuchs, E. Grant, and S. Spach, *Image rendering by adaptive refinement*. ACM, Aug. 1986, vol. 20.

[2] F. Cournoyer and A. Fortier, *Building the Massive Crowds of Assassin's Creed Unity*, 2015. [Online]. Available: `https://youtu.be/AACPbPI955g`.

[3] J. Döllner and H. Buchholz, "Continuous level-of-detail modeling of buildings in 3D city models", in *Proceedings of the 13th annual ACM international workshop on Geographic information systems*, ACM, 2005, pp. 173–181.

[4] M. Weiler, R. Westermann, C. Hansen, K. Zimmermann, and T. Ertl, "Level-of-detail volume rendering via 3D textures", in *Proceedings of the 2000 IEEE symposium on Volume visualization*, ACM, 2000, pp. 7–13.

[5] B. Sunshine-Hill, "Managing Simulation Level-of-Detail with the LOD Trader", *MIG*, pp. 13–18, 2013.

[6] R. E. Cardona-Rivera, B. A. Cassell, S. G. Ware, and R. M. Young, "Indexter: a computational model of the Event-Indexing Situation Model for characterizing narratives", in *Proceedings of the 3rd Workshop on Computational Models of Narrative*, 2012, pp. 34–43.

[7] R. A. Zwaan, M. C. Langston, and A. C. Graesser, "The construction of situation models in narrative comprehension: An event-indexing model", *Psychological science*, 1995.

[8] R. A. Zwaan and G. A. Radvansky, "Situation models in language comprehension and memory.", *Psychological bulletin*, 1998.

[9] M. O. Riedl and R. M. Young, "Narrative planning: balancing plot and character", *Journal of Artificial Intelligence Research*, 2010.

[10] J. H. Clark, "Hierarchical geometric models for visible surface algorithms", *Communications of the ACM*, 1976.

[11] A. Senina, M. Rohrbach, W. Qiu, A. Friedrich, S. Amin, M. Andriluka, M. Pinkal, and B. Schiele, "Coherent Multi-Sentence Video Description with Variable Level of Detail", *arXiv.org*, Mar. 2014. arXiv: `1403.6173v1 [cs.CV]`.

[12] Z. Lin and Z. Pan, "LoD-based locomotion engine for game characters", *Technologies for E-Learning and Digital Entertainment*, pp. 214–224, 2007.

[13] S. Chenney, O. Arikan, and D. Forsyth, "Proxy simulations for efficient dynamics", in *Proceedings of EUROGRAPHICS*, vol. 2001, 2001.

[14] J. K. Hodgins and D. A. Carlson, "Simulation levels of detail for real-time animation", Georgia Institute of Technology, Tech. Rep., 1996.

[15] B. Sunshine-Hill, "Perceptually driven simulation", PhD thesis, University of Pennsylvania, 2011.

[16] C. Brom, O. Serý, and T. Poch, "Simulation level of detail for virtual humans", in *Intelligent Virtual Agents*, Springer, 2007, pp. 1–14.

[17] D. Osborne, Dickinson, and Patrick, "Improving games AI performance using grouped hierarchical level of detail", 2010.

[18] F. Kistler, M. Wisner, and E. André, "Level of Detail Based Behavior Control for Virtual Characters", in *Intelligent Virtual Agents*, Berlin, Heidelberg: Springer Berlin Heidelberg, Sep. 2010, pp. 118–124.

[19] S. Redon, N. Galoppo, and M. C. Lin, "Adaptive dynamics of articulated bodies", *ACM Transactions on Graphics (TOG)*, 2005.

[20] G. Debunne, M. Desbrun, M.-P. Cani, and A. H. Barr, "Dynamic real-time deformations using space & time adaptive sampling", in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ACM, 2001, pp. 31–36.

[21] S. Paris, A. Gerdelan, and C. OSullivan, "Ca-lod: Collision avoidance level of detail for scalable, controllable crowds", in *International Workshop on Motion in Games*, Springer, 2009, pp. 13–28.

[22] C. O'Sullivan, J. Cassell, H. Vilhjálmsson, J. Dingliana, S. Dobbyn, B. McNamee, C. Peters, and T. Giang, "Levels of Detail for Crowds and Groups", *Computer Graphics Forum*, vol. 21, no. 4, pp. 733–741, 2002, ISSN: 1467-8659. DOI: 10.1111/1467-8659.00631.

[23] B. MacNamee, S. Dobbyn, P. Cunningham, and C. OSullivan, "Men behaving appropriately: Integrating the role passing technique into the aloha system", in *Proceedings of the AISB symposium: Animating Expressive Characters for Social Interactions (short paper)*, 2002, pp. 59–62.

[24] S. R. Musse, M. Kallmann, and D. Thalmann, "Level of Autonomy for Virtual Human Agents", in *Advances in Artificial Life: 5th European Conference, ECAL'99 Lausanne, Switzerland, September 13–17, 1999 Proceedings*, D. Floreano, J.-D. Nicoud, and F. Mondada, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 345–349, ISBN: 978-3-540-48304-5.

[25] C. Niederberger and M. Gross, "Level-of-detail for cognitive real-time characters", *The Visual Computer*, vol. 21, no. 3, pp. 188–202, 2005.

[26] R. Narain, A. Golas, S. Curtis, and M. C. Lin, "Aggregate dynamics for dense crowd simulation", in *ACM Transactions on Graphics (TOG)*, ACM, vol. 28, 2009, p. 122.

[27] C. Kruengkrai and C. Jaruskulchai, "Generic text summarization using local and global properties of sentences", in *IEEE/WIC International Conference on Web Intelligence (WI 2003)*, IEEE Comput. Soc, 2003, pp. 201–206.

[28] K. Sarkar, "Bengali text summarization by sentence extraction", *arXiv.org*, Jan. 2012. arXiv: 1201.2240v1 [cs.IR].

[29] L. Suanmali, N. Salim, and M. S. Binwahlan, "Fuzzy logic based method for improving text summarization", *arXiv preprint arXiv:0906.4690*, 2009.

[30] K. B. Mani, "Text Summarization using Deep Learning and Ridge Regression", *arXiv preprint arXiv:1612.08333*, 2016.

[31] R. Mihalcea and P. Tarau, "TextRank: Bringing order into texts", Association for Computational Linguistics, 2004.

[32] M. Roth, A. Ben-David, D. Deutscher, G. Flysher, I. Horn, A. Leichtberg, N. Leiser, Y. Matias, and R. Merom, "Suggesting friends using the implicit social graph", in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2010, pp. 233–242.

[33] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization", *Journal of Artificial Intelligence Research*, 2004.

[34] Wikipedia. (). Page rank for a simple network, [Online]. Available: `https://en.wikipedia.org/wiki/PageRank/`.

[35] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the web.", Stanford InfoLab, Tech. Rep., 1999.

[36] B. Xu, J. Bu, C. Chen, C. Wang, D. Cai, and X. He, "EMR: A scalable graph-based ranking model for content-based image retrieval", *IEEE Transactions on knowledge and data engineering*, vol. 27, no. 1, pp. 102–114, 2015.

[37] K. Compton and M. Mateas, "Casual creators", in *Proceedings of the International Conference on Computational Creativity*, 2015.

Appendix A

User Study

Imagine that youre playing a video game that can automatically create new events while you play. With this survey, we hope to learn which event properties would be more or less interesting to players, given what they know about the game's world.

Suppose that you know the following facts about the game world:

- You are located at the *castle*.
- Your goal is to *talk to the king*.
- You know the following locations: *castle*, *dungeon*, and *valley*.
- You know the following agents: *king*.

Given this knowledge, please answer the following questions:

1. Will you be interested in a new event if it is located at the *castle*?
2. Will you be interested in the new event if it is located at the *fire cave*?
3. Will you be interested in the new event if it is located at the *dungeon*?
4. Will you be interested in the new event if you are directly involved?
5. Will you be interested in the new event if the *king* is involved?
6. Will you be interested in the new event if the *traveler* is involved?
7. Will you be interested in the new event if it is happening today?
8. Will you be interested in the new event if it is happening in a few days?
9. Will you be interested in the new event if it will only happen a long time from now?
10. Will you be interested in the new event if it allows you to *talk to the king*?
11. Will you be interested in the new event if it does *not* allow you to *talk to the king*?
12. Will you be interested in the new event if it will motivate you to achieve other goals?
13. Will you be interested in the new event if it does *not* motivate you to achieve other goals?

Suppose that you know the following facts about the game world:

- You are located at the *Husavik*.
- Your goal is to *capture an elf*.
- You know the following locations: *Husavik*, *Keflavik*, and *Borganes*.
- You know the following agents: *President*.

Given this knowledge, please answer the following questions:

1. Will you be interested in a new event if it is located at *Husavik*?
2. Will you be interested in a new event if it is located at *Keflavik*?
3. Will you be interested in the new event if it is located at *Olafsvik*?
4. Will you be interested in the new event if you are directly involved?
5. Will you be interested in the new event if *the bellman* is involved?
6. Will you be interested in the new event if *the president* is involved?
7. Will you be interested in the new event if it is happening today?
8. Will you be interested in the new event if it is happening in a few days?
9. Will you be interested in the new event if it will only happen a long time from now?
10. Will you be interested in the new event if it allows you to *capture an elf*?
11. Will you be interested in the new event if it does *not* allow you to *capture an elf*??
12. Will you be interested in the new event if it will motivate you to achieve other goals?
13. Will you be interested in the new event if it does *not* motivate you to achieve other goals?

School of Computer Science
Reykjavík University
Menntavegur 1
101 Reykjavík, Iceland
Tel. +354 599 6200
Fax +354 599 6201
www.ru.is