

Research Questions

1. How can we create more realistic and adaptive non-player characters (NPCs) in video games?
2. How can the pursuit for autonomous, self-aware NPCs help to model, develop and test new and novel real-time learning methods and techniques?
3. Can modeling intelligent virtual entities in real-time computer games help to inform models for self-awareness in broader fields such as robotics or more general software applications?

Introduction

NPCs in games often follow predefined scripts for behavior, but research is working towards AI systems that can adapt to player behavior and react in realistic and contextually appropriate ways to make the characters feel more dynamic, responsive, and lifelike.

For example, research in Behavior-driven AI considers how to model NPC behavior using techniques such as Finite State Machines (FSMs), Behavior trees, and the use of Machine learning models and Goal-oriented action planning (GOAP) to model adaptive NPC behavior.

Research in Environmental Awareness focuses on techniques that create NPCs that can react dynamically to the changing world such as modeling sense and perception, while pathfinding and navigation techniques allow for reactive movement and adaptive routing choices.

Research in Emotional AI aims to enhance immersion and realism by modeling emotional responses through the development of personality frameworks and modeling relationship dynamics.

Research in Memory and Learning focuses on persistent memory in NPCs and learning from player's past behaviors, while Advanced Combat and Strategy focuses on developing adaptive combat tactics and strategic decision-making.

Lastly, the research in Social AI aims to model NPC group dynamics, social interactions, conflict and cooperation.

These areas of active research all aim to improve the adaptability and realism of NPCs in games and help to situate the proposed research.

Research Proposal

The proposed research is to undertake the creation of a virtual reality simulation framework for the research and development of NPC behavior based primarily on modeling and developing the real-time decision making faculties (or the 'brain') of these virtual entities based on their momentary perception of their circumstances using virtual sensory observation.

Specifically, the simulation framework will utilize computer game technology to initially develop real-time sensory virtual environments (worlds, realities) for developing adaptive NPC behavior based on perception, observation, and with a special focus on the exploration and characterization of the 'unknown' within these dynamic sensory environments.

In these environments, NPCs will be able to 'sense' and develop 'experience' through the interaction and sensation of other 'objects' (including other NPCs) thereby creating a real-time situational environment in which experiments can be undertaken to create and test new models (or simulate hybrid models) to contribute towards adaptive and realtime AI and NPC behavioral research.

Sensory virtual environments

The premise of this research is that through the simulation of an autonomous entity or entities (NPCs) that have been afforded the facilities to 'experience' and 'perceive' sensory information, and an environment that provides sensory information, that encountered situational experiences in the first instance can be identified and studied, codified and measured, and therefore provide situational data for analysis that incorporates learning, adaptive behavior and detection/classification of situations.

Observations might include the perception of unknown entities such as other characters, objects, heat, danger etc. within the virtual reality, and equally the sensation of multiple aspects of an observation that might define and identify the makeup of an on-going situation.

In this way, objects and the environment that the agent interacts with will need to possess characteristics and properties that can be sensed.

Modeling circumstantial perception in game characters based on the assimilation of surrounding sensory experience, specifically through empirical observation, is a key aspect of the simulations proposed in this research.

An important aspect of such a simulation is an environment that can be designed to emit dynamic circumstantial sensory information as part of the virtual reality, including variable, constantly evolving artificial entities or game objects like NPCs, behaviors and other environmental variations that the learning agents can perceive.

A possible model, for example, might be by sensing stimuli and response sequences that are encountered within the sensory environment, and observing their coordination and organization patterns. This might help build a hypothetical model that can describe what is initially unknown, but which is otherwise empirically ‘experienced’.

Research by Lewis *et al.* (2015), for example, produced a general architectural blueprint (Figure 1) for designing self-aware systems that use sensory input to develop, refine and enhance levels of awareness in software in order to, “... achieve sophisticated autonomous behavior by adapting themselves at runtime and through learning processes that enable ongoing self-change”. Recently this architecture has been specialized by Alemaw *et al.* (2025) to further model the interactions between autonomous agents.

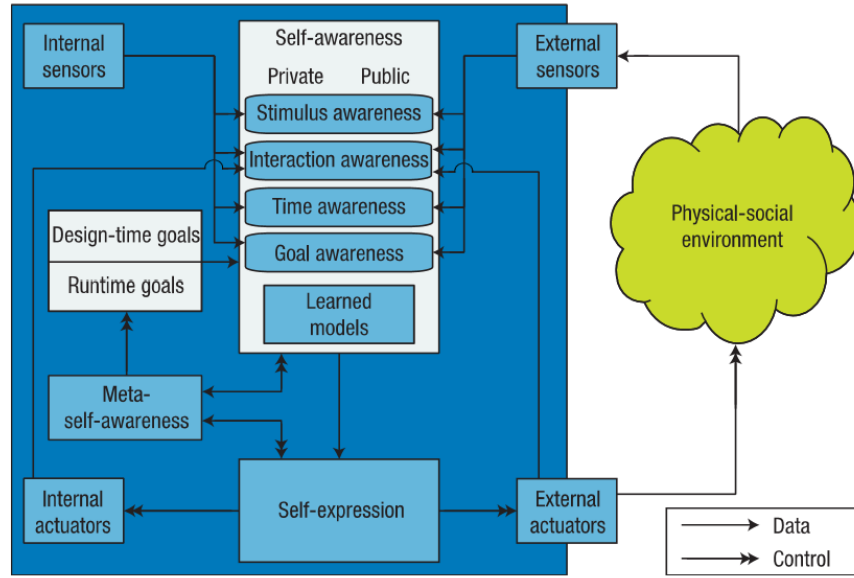


Figure 1: Reference architecture for self-aware and self-expressive computing systems (Lewis *et al.*, 2015)

Examples of possible sensory virtual environments (in which NPCs would be situated) are shown in figure 2.

Outcomes

Aim

Using computer games technologies to simulate real-time experiences of autonomous and self-aware NPCs for developing new and novel real-time models for characterizing the ‘experienced’ unknown, improving contextual adaption, behavior and situation detection in NPCs through exploration and learning within dynamic and sensory-based virtual environments.

Outcomes

1. Simulation framework for modeling approaches for creating NPC behavior based on sensory observation.
 - A) Methods and techniques for representing and delivering sensory stimulus in virtual environments to NPCs (observational protocol)
 - B) Methods and techniques for implanting self-awareness in NPCs using sensory observation from sensory-based virtual worlds.



Figure 2: Potential 2D and 3D sensory simulation environments (Mathews, 2022)

2. Models for simulating sensory-based online learning approaches such as:
 - A) Real-time situation detection using temporal networks
 - B) Real-time situation changes/boundaries using scene graphs
 - C) Dynamic behavior generation using dynamic behavior trees
3. Research findings and observations of simulated approaches

Design Approach

Game engines such as Crytek’s Target Tracks Perception System (TTPS) (Rabin, 2014a) and Epic’s Unreal’s AI Perception System (‘AI Perception’, no date) are two examples of perception systems that base their architecture on the notion of a stimuli being elicited by a source object in the world, and perceived by an agent. This essentially provides the agent the ability to subscribe to sense events and have sensation delivered to them when they are emitted by event sources in the world.

The perception system used in the game, *Mark of the Ninja* by Klei Entertainment, is described in (Rabin, 2014b), as a “...sense detection architecture”, whereby certain “interest” objects (interest sources) can be placed in the world, and can be sensed by agents.

These so-called ‘interests’ themselves determine if an agent (or agents) are able to be perceived by them at any moment in time (they are updated in real-time in the game loop), and could be good candidates in this research for modeling the implementation of virtual emitters and sensors within the virtual environment for stimuli-response data collection purposes.

For example, figure 3 illustrates a character, labeled as ‘Player’ who is registered as such an interest source that provides sensation events such as auditory stimuli to surrounding interested parties when they are within an established sensation proximity radius.

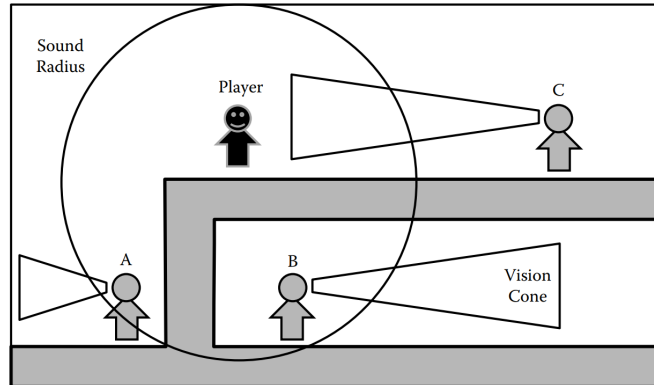


Figure 3: Player as interest source producing audible stimuli (Rabin, 2014a)

This research proposes to model sense detection using similar mechanisms to simulate the production of situational stimuli within the simulation environment.

The creation and propagation of sensory stimuli within the simulated environment(s) would likely be handled by the event management system within the simulation framework library, which is currently under development (along with other framework components that aim to simulate causality and the formation of expectations from sensory input), and would continue to evolve with this research.

Methods and Analysis

The general research approach is outlined as a series of phases as illustrated in figure 4

The first phase is concerned with modelling and simulating virtual environments to establish a situational learning environment for NPCs.

The next stage creates self-awareness models in NPCs that are capable of experiencing and sensing their virtual environments.

The penultimate phase implements autonomous exploration of NPCs within the virtual environments to dynamically collect real-time observations about the environment to inform the models used for online learning.

The final stage is to use the learnt models to produce adaptive and contextual behavior and decision making in the NPCs.

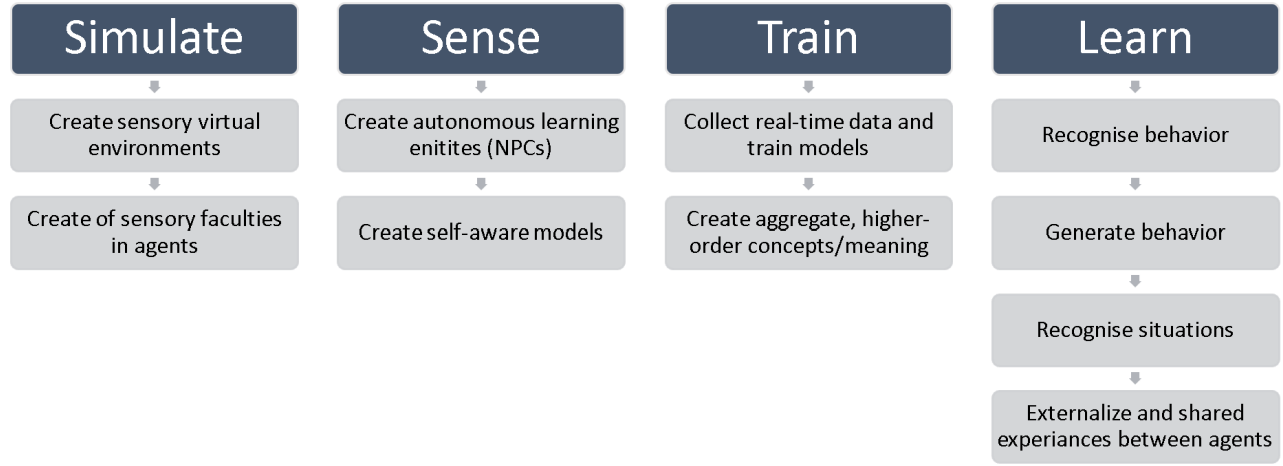


Figure 4: Research phases.

Tools and Techniques

A foundational cornerstone of this research is the use of computer game technology and AI to develop real-time simulations of NPCs and their environments to model learning, adaptability, and the implementation of self-awareness and creation of contextual behavior.

This requires translating abstractions that capture methods for implementing these abilities into software models which can be harnessed and utilized by the reasoning repertoire of any such agent in order to assist in carrying out sophisticated tasks that are normally otherwise expected of living things (remembering, deliberating, weighing options, feeling and deciding based on their own motivations, circumstances and requirements). This means a large portion of this research requires advanced programming for modeling ideas and implementing them as software models within computer gaming simulations.

For example, simulations include implementing aspects of 2D/3D graphics, implementing and developing scripting and AI systems and engineering the design of the software models that model ideas such as the sensory and event management systems, the autonomy and exploratory behavior of NPCs and their reasoning capabilities etc. Other techniques often used in game development include the flexibility of data driven development which is capable of dynamically loading, storing and transferring models, behaviors, and other functionality. The formation of these models, and the techniques and approaches for implementing them are ultimately what will form the contribution of this research.

Technologies to simulate the experiments are expected to include C++, Lua, OpenGL/DirectX, and the use of game AI systems that can be used to implement autonomous exploration and store and trigger behavior such as FSMs, BTS and more flexible learning systems which include models for developing persistent memory and self-awareness, including learning systems such as machine learning.

An important aspect of implementing a framework for system is a systematic approach to its design and engineering. A strategic part of this research is to adopt an approach to engineering software models that are re-usable and adaptable to change.

To achieve this, this research will develop a software library (or libraries) along side the main agent simulations, and this will be integrated into these simulations. In this way, much of the programming and model implementation will be done in an abstract way such that the code, including the AI models, can be reused throughout further simulations. This will encourage the composition of more creative variations from the engineered models.

By creating a library (or libraries) that embody the different learning approaches and strategies used in the research,

they can then be used to highlight and externalize learnt approaches, as well as be used to produce isolated demonstrative examples. This will also help with an iterative progression of general understanding by incorporating them into newly developed models.

This component-based strategy is also likely to improve the rate at which productivity can proceed particularly in making future sensory-based systems as the artifacts and models can be more easily used by other participants who wish to incorporate the ideas into independent projects or participate in this research. This will enable them to be more reusable in more situations, and also can drive creativity through making experimental variation easier.

This will also allow the ability to abstract and externalize the complexity of systems and sub-components that are developed such that they can be practically studied independently.

As the research progresses, the library could develop into discernable areas which can be tested and reasoned about separately. These component-based libraries will also allow for the validation of software models that can be verified as correct through unit testing. This will not only encourage change and refactoring, but also likely help document the components while showing how they are used. Its strongly believed that verifiable software models that are testable make exploration and usage of those models more robust than if they were integrated tightly into the specific simulations.

Advancing knowledge

In many ways, a simulated virtual entity such as NPC and their advancements in dealing with circumstantial information can be considered as an abstraction for what could be utilized in other fields such robotics, intelligent assistants or even self-driving cars. This makes researching and advancing these capabilities pivotal in developing more intelligent and self-aware applications in the future.

This research aims to explore how using computer games technologies can be used to simulate and develop new and novel real-time models for characterizing the ‘experienced’ unknown, improving contextual adaption, behavior and situation detection in real-time situations.

Beyond the simulation and development of models for self-awareness and real-time learning in NPCs within sensory environments, there are some specific approaches that might be explored in more detail during this research.

Temporal Networks

Temporal networks are graph representations of relationships that occur over time between entities, often represented as ‘nodes’, with edges between them representing a relationship (figure 5).

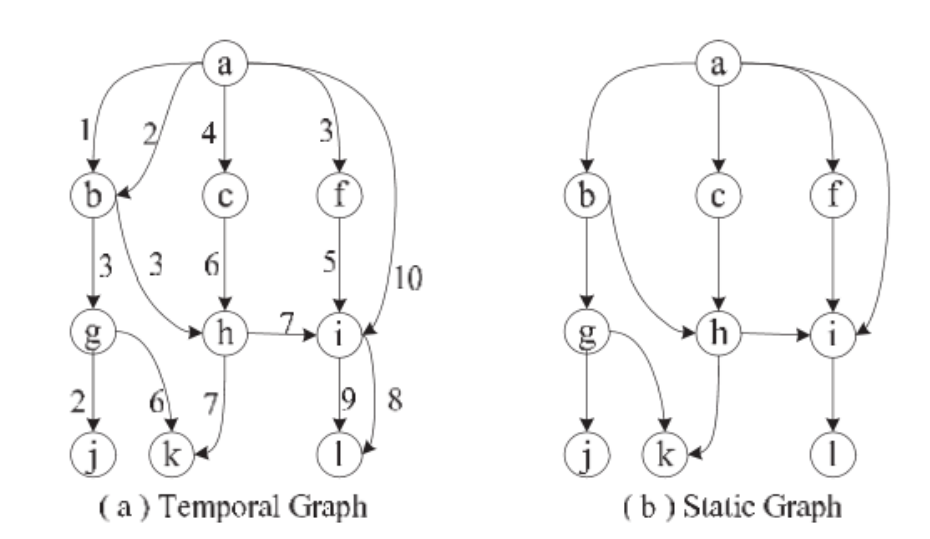


Figure 5: Temporal vs Static Graphs (Wu *et al.*, 2014)

An approach that is compelling is to stream the generation of temporal networks from the real-time experiences of causality based on stimuli and responses within virtual worlds.

This offers an means to study how these networks evolve over time, including how similar or dissimilar they are in certain situations, and what they can tell us about the nature of those interactions and the effect that environmental stimuli have.

Work by (Masuda and Holme, 2019) investigated how general system states could be detected when representing states as sequences of changing directed graphs over time (temporal networks), as shown in figure 6.

This insight provides the opportunity to model the formation of real-time situations as the formation of temporal networks.

These could be used to model the relationships that define and identify situations in real time. This also provides the potential to model situational changes, in response to experienced stimuli within the virtual environment, as evolving temporal networks.

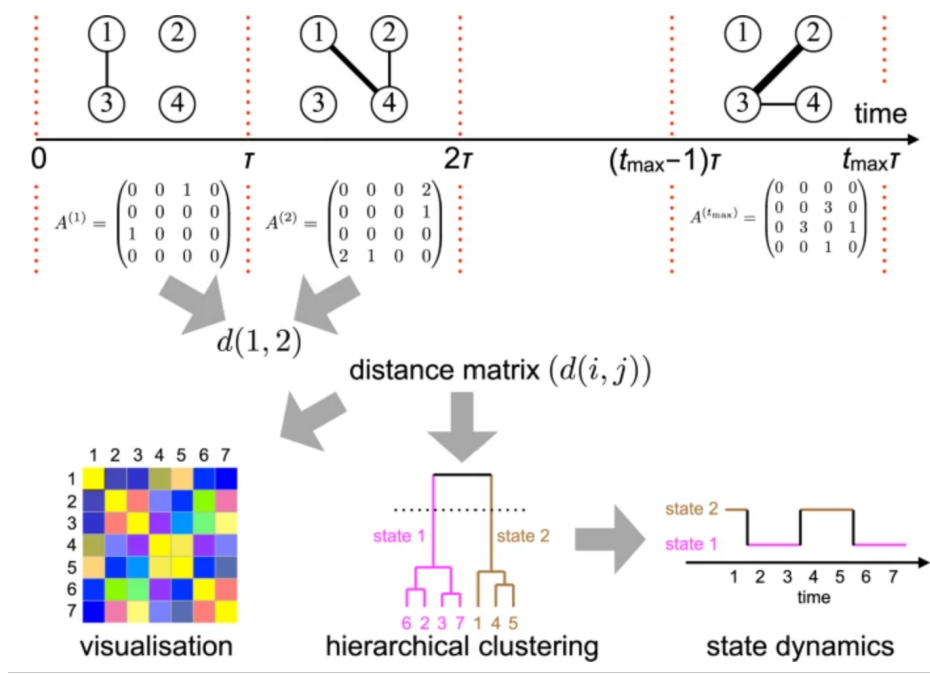


Figure 6: Representing system states over time (State Dynamics) (Masuda and Holme, 2019)

In this proposed research, we might not only be concerned with detecting system states (situations) from sensory events, but also allowing agents to identify the events that cause those states to happen and also predicting the behavior/situation that is likely occur as a result of the event.

Research by (Wu *et al.*, 2014) shows that ‘minimum temporal paths’ statistics (earliest arrival path, latest departure path, fastest path, shortest path) can be used to analyze temporal networks, and this has been used in related research by (Ceria and Wang, 2023), (Zhan *et al.*, 2021), to show that it is possible to characterize and generalize the temporal and spatial properties of evolving networks to distinguish differences or similarities between them over time.

The implication of this is that such mechanisms, and improved tooling for calculating temporal metrics such as those provided by (Oettershagen and Mutzel, 2022), could be used by NPCs to compare properties of encountered situations to detect similar situations in real-time within sensory environments.

To represent sensory and environmental changes over time as temporal networks, this research proposes to model the formation of such temporal networks in real-time from the sensory events that occur in response to the experience the agent(s) encounters within the sensory world. Such time-based graph sequences could be used to form signatures to to detect situations, and derive richer causality relations within the world.

An important consideration is to be able to accurately distinguish discrete states and their contextual characteristics in real-time amidst complex, noisy and high-event rate virtual reality simulations. While (Masuda and Holme, 2019) uses graph distance, and (Zhan *et al.*, 2021) the fastest arrival distance (FAD) to identify and generalize network graphs into discrete states, this might be too coarse a generalization, and might require more features of the

network/environment to be considered in representing situational state, particularly in complex environments that can produce large volumes of data such as those seen in real-time environments (like games).

Another interesting avenue of simulation is using real-time (online) sensory experience obtained within the virtual environment to train and test NPC behavioral and self-awareness models using machine learning.

For example, techniques such as Bayesian networks, Hidden Markov models and SVM could be used to incrementally describe sensory events that are experienced by the agent.

In conjunction with the above methods, research by IJsselmuiden *et al.* (2014) showed how the use of Situation Graph Trees (SGTs) can be used for modeling and detecting dynamic situations. These are in many ways like behavior trees (BTs) that detect situations by traversing a graph of conditions that match situation signatures.

While SGTs, like BTs, are often predefined constructions used to respectively detect well-known situations (see figure 7) or trigger well-known behavior (in the case of BTs), an interesting pursuit is the dynamic creation of SGTs and BTs, based on the real-time experiences of agents encountered within virtual environments.

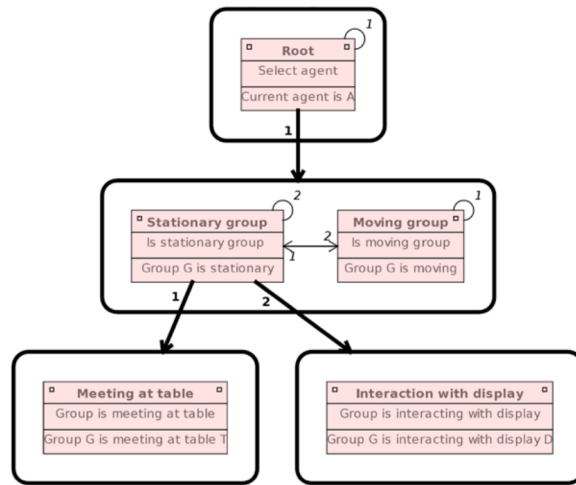


Figure 7: Situation graph tree (IJsselmuiden *et al.*, 2014) used to detect group behavior

Observing and modeling behavior

Typically behaviors in games are not generated from experience and instead are pre-defined behavioral logic or scripts that define how the agent should behave in predefined circumstances. This would be pre-defined actions/behavior to predefined circumstances, meaning behavior is not created, but that pre-existing behavior is triggered mostly using finite state machines (FSMs) and BTs.

Through this research there is scope though an experiential learning model to dynamically generate these behavioral scripts in real-time as behavior trees (see figure ??).

For example, an approach developed by (Robertson and Watson, 2015) was to dynamically generate behavior trees (BTs) based on the supply of recurring textual action sequences. Their research showed that a resultant "... behavior tree was able to represent and summarise a large amount of information from the expert behavior examples". These examples could instead come from the actions of NPCs and other dynamic objects that 'live' and behave within real-time virtual environments.

This would allow agents to dynamically create rules for observed behavior based on their experience and observations of situations and the behavior of others. The agent could use this as part of its behavioral knowledge base and carry out those behaviors in the future. Equally these learnt behaviors (trees) might be transferrable to other agents for them to use when situations call for it, in a similar way that transfer learning is used in machine learning.

Scene Graphs

An approach to modeling situation detection that is also compelling is to compose sensory virtual world objects into a hierarchical dependency scene graph and to model sensory events as influences upon those objects.

This may provide opportunities to identify the effects of causality and the influence it has on multiple related objects such as the propagation of effects to their dependencies.

For example, pushing a box with a pen in it, will also move the pen relative to the box. In this way, when a stimuli affects an object, the resulting change/response can be observed in its dependencies, allowing for a better generalization of the effect of the stimuli.

A hierarchical representation could serve as a means to aggregate, group and generalize the effects that multiple changes in the environment.

For example, this could help analyze situational consistency and help derive situational boundaries by forming level of detail change calculations, where a lower level of detail might ignore smaller, more frequent or consistent changes in the scene graph and focus on bigger changes in order to generalize a situation boundary at that level.

Figure 8 illustrates an example of organizing virtual objects into a scene graph.

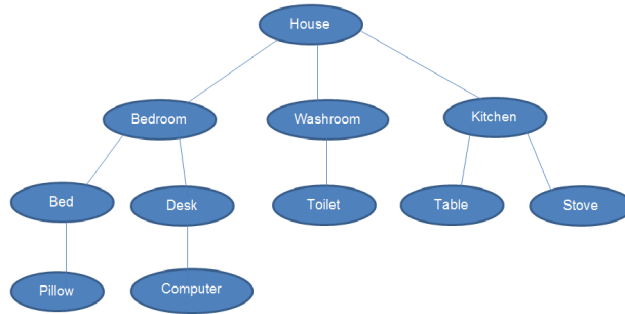


Figure 8: Organizing objects into a scene (Unknown, 2013)

This approach should also make it possible to identify which objects should be actively observed for changes (as a result of stimuli in the environment) without checking every object in the scene, e.g., pushing the box means its dependents might not need to be actively checked, as the majority of their changes are likely derived from the box's changes, therefore saving computation time. While this is useful, there is likely still a need to be able to determine when objects that are still important, i.e make up a situation, but have not actually changed in response to the stimuli (universals).

The basic premise is that the scene graph is checked for differences caused by stimuli and these changes can be used to detect situational boundaries. An illustration of this approach is outlined in figure 9.

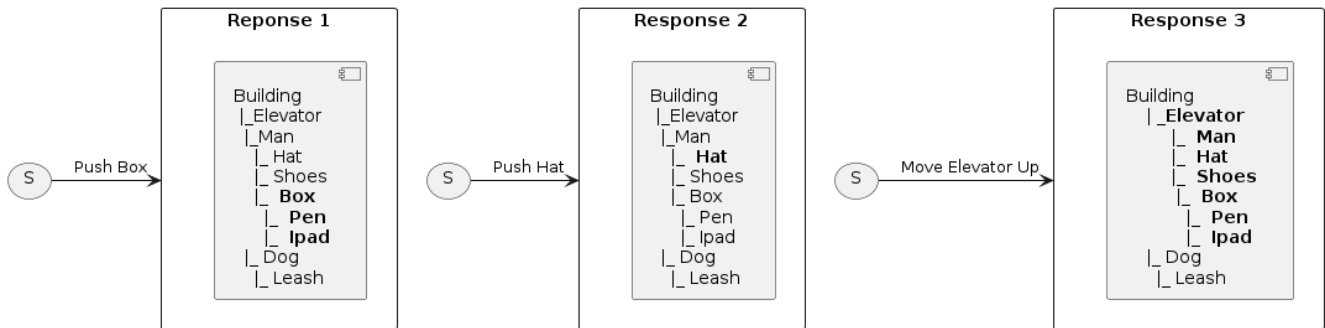


Figure 9: Detecting changes in the environment due to environmental stimuli using a scene graph

About me

I have been actively engaged in researching computer game technologies while building out a game engine that incorporates many of my interests in game development including graphics, multiplayer networking and behavioral AI.

Currently I work full time (remotely) in Uxbridge as a software developer for Citrix Systems creating a variety of hybrid systems (on-premise and cloud-based). Prior to this I worked in a similar capacity for VMWare. I have been developing systems for software companies for the last 15 years, most of which while studying.

I enjoy the process of learning and in that time I have completed the following study:

- MSc Software Engineering
- MSc Computer Games Technology
- MSc Cyber Security (thesis candidate)
- BSc (Hons) Open

My last three theses have revolved around researching computer games, namely:

- Evaluating the design requirements for a secure, low latency, multiplayer network protocol.
- Applying and evaluating functional programming paradigms and techniques in developing games.
- Applying pattern oriented designs in complex software (Games)

I'd like to continue learning by doing a formal research degree by focusing on what I already do in my spare time, which is studying computer games development.

This research project is intended to be self-funded, part-time and based in primarily in Uxbridge. This research appears to align very well with an existing self-funded topic offered by Brunel, entitled 'Multi-Agent Learning on Sensory data for Autonomous Systems'.

Stuart Mathews

References

- 'AI Perception' (no date). Available at: <https://docs.unrealengine.com/5.1/en-US/ai-perception-in-unreal-engine/> (Accessed: 5 January 2023).
- Alemaw, A. S. *et al.* (2025) 'Modeling Interactions between Autonomous Agents in a Multi-Agent Self-Awareness Architecture', *IEEE Transactions on Multimedia*, pp. 1–16. doi: 10.1109/TMM.2025.3543110.
- Ceria, A. and Wang, H. (2023) 'Temporal-topological properties of higher-order evolving networks', *Scientific Reports*, 13(1), p. 5885. doi: 10.1038/s41598-023-32253-9.
- IJsselmuiden, J. *et al.* (2014) 'Automatic understanding of group behavior using fuzzy temporal logic', *Journal of Ambient Intelligence and Smart Environments*, 6(6), pp. 623–649. doi: 10.3233/AIS-140290.
- Lewis, P. R. *et al.* (2015) 'Architectural Aspects of Self-Aware and Self-Expressive Computing Systems: From Psychology to Engineering', *Computer*, 48(8), pp. 62–70. doi: 10.1109/MC.2015.235.
- Masuda, N. and Holme, P. (2019) 'Detecting sequences of system states in temporal networks', *Scientific Reports*, 9(1), p. 795. doi: 10.1038/s41598-018-37534-2.
- Mathews, S. (2022) 'Cppgamelib'. Available at: <https://github.com/stumathews/cppgamelib> (Accessed: 21 January 2023).
- Oettershagen, L. and Mutzel, P. (2022) 'TGLib: An Open-Source Library for Temporal Graph Analysis'. arXiv. doi: 10.48550/arXiv.2209.12587.
- Rabin, S. (2014a) 'Crytek's Target Tracks Perception System', in *Game AI Pro*. A K Peters/CRC Press, pp. 432–441. doi: 10.1201/b16725-37.
- Rabin, S. (2014b) 'How to Catch a Ninja: NPC Awareness in a 2D Stealth Platformer', in *Game AI Pro*. A K Peters/CRC Press, pp. 442–451. doi: 10.1201/b16725-38.
- Robertson, G. and Watson, I. (2015) 'Building behavior trees from observations in real-time strategy games', in. IEEE, pp. 1–7. doi: 10.1109/INISTA.2015.7276774.
- Unknown (2013) 'Game Engine Design Blog: Scene Graphs', *Game Engine Design Blog*. Available at: <https://aldwinchogameengine.blogspot.com/2013/11/scene-graphs.html> (Accessed: 16 March 2025).
- Wu, H. *et al.* (2014) 'Path problems in temporal graphs', *Proceedings of the VLDB Endowment*, 7(9), pp. 721–732. doi: 10.14778/2732939.2732945.
- Zhan, X.-X. *et al.* (2021) 'Measuring and utilizing temporal network dissimilarity', *arXiv.org*. Available at: <https://arxiv.org/abs/2111.01334v1> (Accessed: 1 January 2024).