The Simulation of Needs

Connor Easterbrook
18015101

University of the West of England

March 22, 2023

Abstract

The simulation of skills, like programming, provides users with an immersive and interactive experience in the world of programming and entrepreneurship. This project utilises autonomous agents within a dynamic market to simulate the creation of a startup software development company. The simulation incorporates a complex set of variables, including agent skills, project complexity, and market competition, to simulate the creation and growth of a software development company. The proposed simulation model can be adapted to other domains beyond software development, providing a valuable tool for understanding the dynamics of entrepreneurship and market competition. This project is analysed as both commercial viability as a game and an educational simulation, with a concluding discussion on areas of improvement.

1 Introduction

1.1 Simulations

The field of computer science and entrepreneurship has seen a significant growth in recent years, with a heightened interest in developing software and creating successful startups. Alongside this, there has been a significant increase of interest in artificial intelligence and simulations, which allow us to explore ideas previously impossible (Sousa, Cravino, and Martins, 2023). Makridakis, 2017 suggests that artificial intelligence will "rewire" the development of future technologies. This report aims to introduce a simulation video game that provides players with an opportunity to explore these fields in a unique and interactive way.

The artificial intelligence system created in this project uses simple sorting algorithms to calculate what "need" the character has and how essential it is to get it done (Maes, 1993). A visible example of this can be seen in Figure 1. Barrera, 2018 details the creation of an AI system in Unity that is both optimised and able to scale effectively. This alleviated concerns of the class structure being unable to scale effectively. This project was conducted in the Unity Game Engine due to its NavMesh (He, Shi, and Li, 2016) tools, which helped immensely in the development of artificial intelligence.

1.2 Related work

Maxis, 2014 is a game that uses autonomous agents in a similar way, shown in Figure 2, while Games, 2012 simulates the development of video games. The closest relation, however, is Coredumping, 2015's game 'Software Inc.' which focuses less on the autonomous agents and more on the marketing and development.

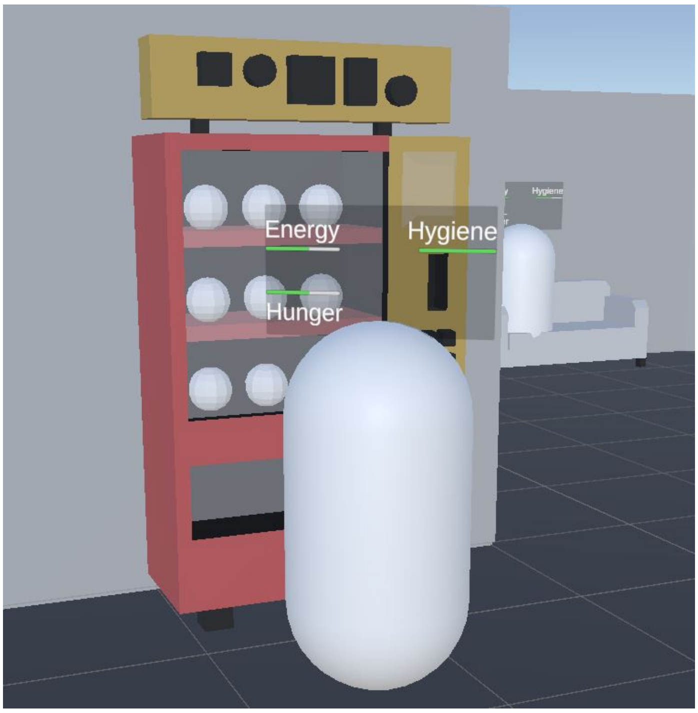

Figure 1: *Early example of autonomous agents with needs.*

1.3 Rationale

The potential applications for this project can effectively be categorized in two ways - the autonomous agent system, and the video game. The autonomous agents can easily be utilized in any simulation project that requires the agents to have their own specific needs and the ability to solve their issues themselves (Maes, 1993). The video game provides players with a realistic and immersive experience in the world of programming and entrepreneurship, which is both fun and educational (Coredumping, 2015).

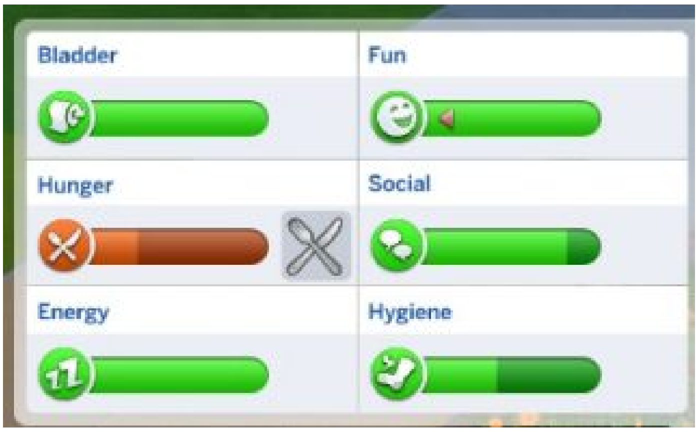

Figure 2: *Example of autonomous agents' needs in The Sims 4.*

2 Methodology

Due to the base requirement of this task being a simulation that could double as a game, it was believed that the best approach was using autonomous agents and allowing only limited control to the user. This user limitation allowed for the simulation to take its own course whilst keeping the user involved enough to be able to change the general flow of game play. As Aldrich, 2005 suggests - users learn best by doing and experiencing the results of their actions.

The usage of artificial intelligence techniques were to create a complex and dynamic system of autonomous agents that interact with each other within a simulated market economy (Winston, 1984). The project also includes a grid-based free building system that allows users to construct and modify their environment. Echoing the sentiments of Wai et al., 2021, this involved a combination of programming, algorithm development, and user testing to refine the system and ensure its effectiveness. The breakdown of the AI, market, and building systems will be split into their own respective sub-sections.

2.1 Artificial intelligence

Autonomous agents based on needs are software entities that act independently to satisfy a set of predefined needs or goals, much like Maxis, 2014's game The Sims 4. These agents are programmed to monitor the status of their needs and take actions to satisfy them. A soft limit was introduced to ensure that the agents, or "performers" as referred to in the project, work until one of their needs reach it. Every variable regarding needs is altered based on several characteristics given to make each agent unique. An example of this would be the traits serialized from a .JSON file in order to change the needs threshold. This approach allows agents to prioritize their actions and allocate resources efficiently, and create a more adaptive autonomous system (Maes, 1993). Wai et al., 2021 was referenced when managing the collisions between agents when in a crowd.

$$N(t) = N_0 - rt + g(t)u(t-T) \tag{1}$$

In this equation, $N(t)$ represents the current value of the need at time t, N_0 represents the initial need value, r represents the rate of decrease in the need value over time, and $g(t)$ represents the rate at which the need value increases when the user interacts with the object that satisfies the need. Finally, $u(t-T)$ is a step function that becomes 1 when t is greater than or equal to T, the threshold value below which the user seeks the object that satisfies the need.

While the agent does work, there are also calculations surrounding its work efficiency.

$$P(t) = s \cdot M(t) \cdot e^{-aE(t)} \cdot e^{-bM(t)} \tag{2}$$

In this equation, $P(t)$ represents the current rate of work progress at time t, s represents the skill level of the character, $M(t)$ represents the motivation level of the character at time t, and $E(t)$ represents the energy level of the character at time t. The constants a and b are parameters that control the effect of energy and motivation levels on the work progress rate. As energy and motivation levels decrease, the corresponding exponential terms increase, leading to a decreased work progress rate.

A performer's skill level can accumulate over time spent working. This means that the user should aim to maximize their work allocation while also keeping in mind that performer salary will increase with each level.

2.2 Dynamic market

The mechanic that the simulation is based around is the dynamic market. Similar to Monahan and Sobel, 1994's work, the dynamic market system in this project generates companies, products they produce, and their net worth based on certain parameters. It also keeps track of the software products the player has created and the statistics regarding its sales. There is a breakdown of salary costs each month, reflecting the financial reality of running a business. This system aims to create a more realistic and immersive experience, enabling players to make informed decisions based on their company's financial health and performance in the market.

A new company or product is introduced via the Time Manager. It is a system that tracks when a day finishes and calculates a percentage chance for a new item to appear in the market. This creates a sense of competition that could cause trouble to the user's simulated company, as if two recent products are too similar, the demand for the second product will be lower. The initial calculation for the demand of a product at the end of the workday is showcased in the example below.

```
foreach (Product p in completeProducts)
{
    if (p.Popularity > 0)
    {
        DaySales(p);
        p.Popularity -= (Random.Range(0, 5) + p.
    Price) * p.Age;
    }

    p.Age += 1;
}
```

2.3 Building system

Similar to Maxis, 2014's The Sims video games, this project uses a free building system. This building system uses a room-scanning flood fill algorithm that is triggered every time a corner is updated, enabling the system to detect and adjust to any changes made in the building layout. Featuring object-specific placement checks, the system ensures that different objects can only be placed in locations based on their properties. For instance, indoor furniture can only be placed inside rooms, while doors are limited to walls. These checks guarantee that the building is functional and follows logical rules for a more realistic and enjoyable user experience. This system can be seen in Figure 3.

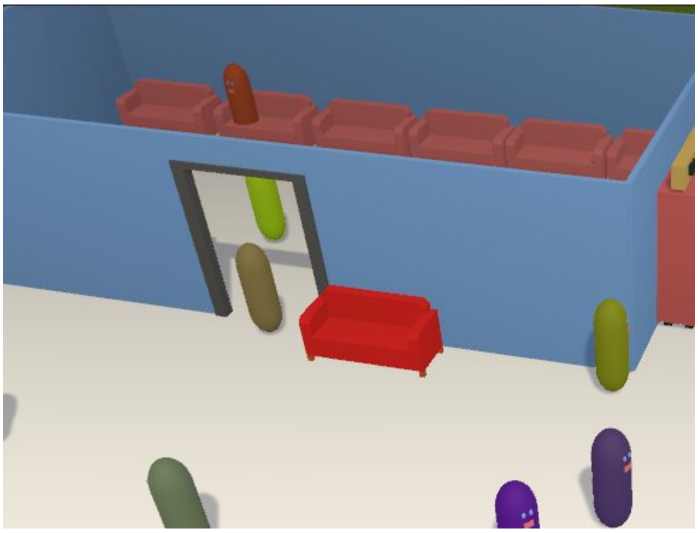

Figure 3: *Example of furniture placement in Simulation. The strong colour indicates the building object.*

3 Evaluation

Overall, this project showcases the development of a simulation video game featuring autonomous agents in a company setting. From an academic perspective, the game provides an opportunity for users to develop skills in strategic thinking, decision-making, and problem-solving in a simulated business environment. The usability and performance of the project may require careful consideration and optimization, depending on the specific application and intended user audience.

3.1 Usability

One significant advantage of this simulation video game is its usability. The game's user interface is intuitive and easy to navigate, providing a seamless gaming experience. Though, it would be better to provide users with clear instructions and feedback, ensuring that they understand the objectives and their progress towards achieving them. The autonomous agent code base of the project can be dissected and used in any other AI-based piece of work, as it is built to be modular, though it may be restricted to more advanced users. Overall, the game's usability is a strength that contributes to its effectiveness as an educational tool.

3.2 Performance

In terms of performance, the video game would likely require modern hardware to run to calculate each autonomous agent's need and work calculations. Multi-threading these calculations could offer an increase in performance, while certain code could certainly be optimized. The AI system created in this project runs well when dissected from the video game aspects and solely focused on controlling the autonomous agents. Further optimization and testing will probably be necessary to increase performance to the point of this project's performance hit to a software system be unnoticeable to the end user. The current performance can be viewed in Figure 4. The spike showed represents the initial allocation of tasks and then the resource usage of reallocation tasks has next to no impact on the performance. The section before the performance spike showcases the generation of the world and registration of all currently placed objects.

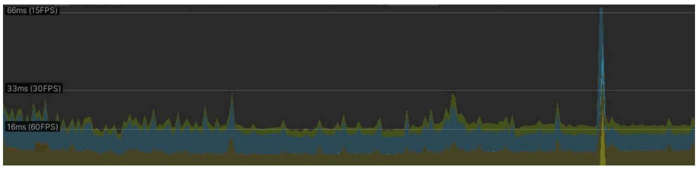

Figure 4: *Showcase of the start of the simulation. The spike represents work allocation.*

4 Conclusion

The system provided yields a good solution to the task specifications. An autonomous agent system achieves the simulation aspect of the task, while the video game systems add more depth to the gameplay. The free building and dynamic market systems function as smaller simulations, simulating the building of structures and a market, respectively. Each system would benefit from multi-threading so that it functions without being overly resource intensive. The end product of this project forms a video game and an autonomous agent library that allows for implementation in other Unity Game Engine projects. Further development could be conducted by adding more depth to the time management and market systems, allowing for a fully simulated experience. An example of this would be implementing individual scheduling for workers and teams.

In conclusion, it is believed that this project achieves the intention of being a simulation system appropriately, but there is room for improvement.

Bibliography

Aldrich, Clark (2005). *Learning by doing: A comprehensive guide to simulations, computer games, and pedagogy in e-learning and other educational experiences*. John Wiley & Sons.

Barrera, Raymundo (2018). *Unity 2017 Game AI Programming-: Leverage the power of Artificial Intelligence to program smart entities for your games*. Packt Publishing Ltd.

Coredumping (2015). *Software Inc.* Steam.

Games, Greenheart (2012). *Game Dev Tycoon*. Steam.

He, Zhang, Minyong Shi, and Chunfang Li (2016). "Research and application of path-finding algorithm based on unity 3D". In: *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*. IEEE, pp. 1–4.

Maes, Pattie (1993). "Modeling adaptive autonomous agents". In: *Artificial life* 1.1_2, pp. 135–162.

Makridakis, Spyros (2017). "The forthcoming Artificial Intelligence (AI) revolution: Its impact on society and firms". In: *Futures* 90, pp. 46–60.

Maxis (2014). *The Sims 4*. Origin.

Monahan, George E and Matthew J Sobel (1994). "Stochastic dynamic market share attraction games". In: *Games and Economic Behavior* 6.1, pp. 130–149.

Sousa, Sonia, José Cravino, and Paulo Martins (2023). "Challenges and Trends in User Trust Discourse in AI Popularity". In: *Multimodal Technologies and Interaction* 7.2, p. 13.

Wai, Sim Keng et al. (2021). "Autonomous Agents in 3D Crowd Simulation Through BDI Architecture". In: *JOIV: International Journal on Informatics Visualization* 5.1, pp. 1–7.

Winston, Patrick Henry (1984). *Artificial intelligence*. Addison-Wesley Longman Publishing Co., Inc.

Appendix A

Week 1 - `https://youtu.be/ZFccGZZfaqU`
Week 2 - `https://youtu.be/G1JS7MAk6iM`
Week 3 - `https://youtu.be/eSMpIF7s0xY`
Week 4 - `https://youtu.be/d6R0cTrSK7g`
Week 5 - `https://youtu.be/f2wfAcgsVq0`