**Project Proposal**

“Replacing traditional AI and Human playtesters using Machine Learning Agents”

Danielis Golubovskis

Student Number: 170798389

MComp in Computer Science (Games Engineering) (I610)

School of Computing, Newcastle University, Newcastle Upon Tyne, UK

**Motivation and rationale**

Artificial Intelligence (AI), and especially Machine Learning are one of the hottest tech trends nowadays, however the technology of AI is not at all new. The reason for their recent popularity is the advancements and utilization of the Graphical Processing Unit (GPU) for the training of the neural network architecture in modern AI (year 2011-present) [3]. Game playing was an area of research in AI from its inception.In fact, the first rudimentary AI has been used in video games since the early 80‘s with the implementation of a Finite State Machine model that is still used in modern games [4]. The FSM is a model of computation based on a hypothetical machine made of one or more states which are defined and programmed by the developer. While FSM are hard-coded, Machine Learning is the ability of a system to learn and improve from experience, without being explicitly programmed. Until now, the kind of self-learning AI – namely the deep learning subset of Machine Learning – that‘s led to advances in self-driving cars, computer vision, and natural language processing hasn‘t really bled over into commercial game development. The fact that Machine Learning has not yet dominated the video game industry made me very curious to find out what are the challenges that Machine Learning in video games faces.

**The problem:**  
Machine learning algorithms can offload a lot of the work that a game developer currently needs to perform. Currently, video game studios spend hundreds of man hours scripting NPCs alone. NPC character control and other things like the generation of unique environments could be automated if reliable algorithms are developed.   
Playtesting is also a crucial part of the game development cycle. For big titles the process of playtesting can take for thousands of hours which requires playtesters and funds. The truth is, for companies/developers with limited resources it‘s not easy to find someone to playtest a game for them. On the other hand, AI can execute hundreds of test cases in a matter of seconds and perform human-like tasks.

**Similar existing projects – OpenAI Five and Candy Crush Saga**OpenAI has succeeded to train a neural network to play Dota 2 and beat professional players in a 5v5 match [6]. Similar to my project, they used a Proximal Policy Optimization reinforcement algorithm to train the neural network, however on a much larger scale with 256 GPUs and 128,000 CPUs. My project is instead focused on not beating human players, but to replace them and to use the trained agents to playtest newly created game levels. More so, the success of my approach would suggest that replacing traditional AI with trained agents is achievable with much smaller systems.  
Another example is Candy Crush which uses trained neural networks to mimic human behavior to predict the difficulty of levels [5]. While the idea is near identical to that of mine, there are a few key differences. Candy Crush uses a supervised learning method and requires player data to teach the network, which means there is still a need for human player presence. My approach, however, is based on reinforcement learning. This means that my approach can be used on games that are still in development that do not have any player data.   
This project also goes beyond existing work by attempting to directly compare the performance of traditional FSM against ML-trained agents and ML agents against human players on which is something I have not yet seen in published works.

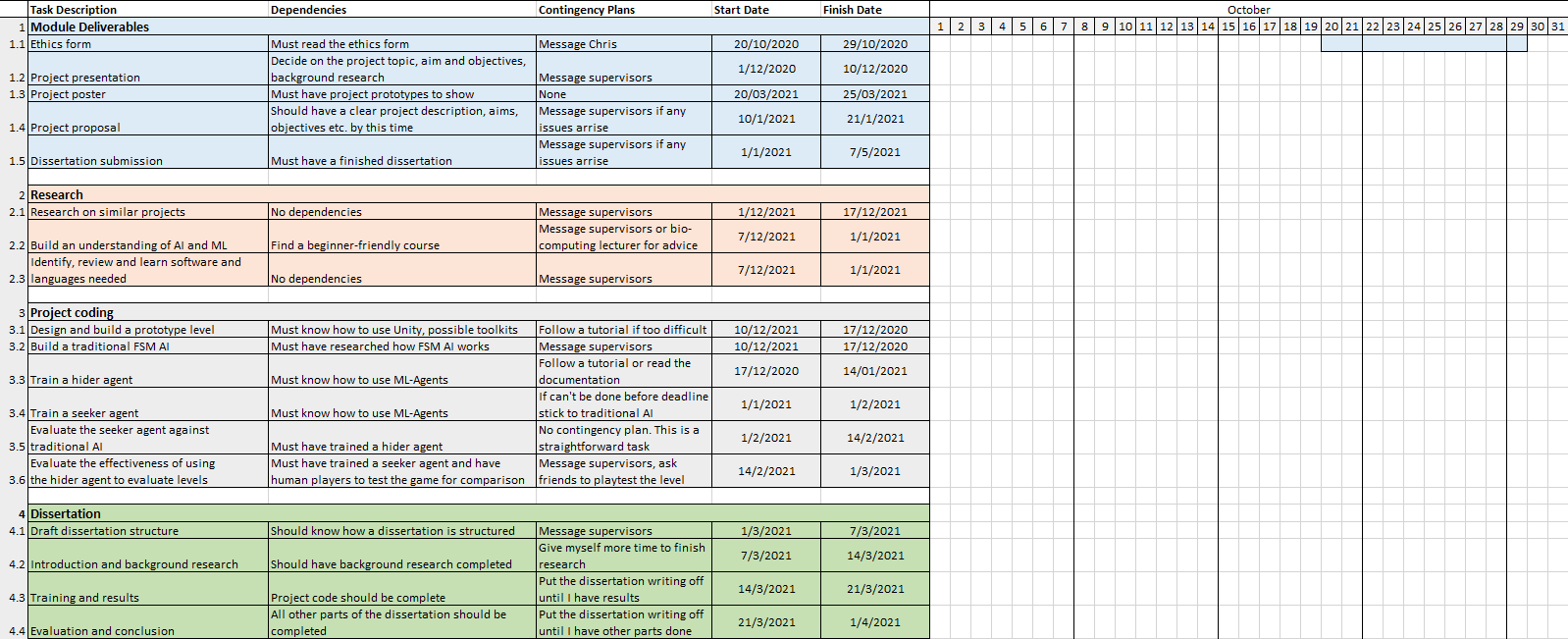
**Aims**

The aim of the project is to **train ML agents to evaluate their suitability to perform game level testing and to replace traditional Finite State Machine agents**. My approach is to use Unity’s Machine Learning Agents toolkit to develop a simple Hide and Seek game with a hider and seeker agents. The hider would collect targets around the level while trying to avoid being caught by the seeker. First, a fully scripted seeker will patrol the environment and chase the hider if spotted. Then another agent will be trained to mimic the behavior of the fully scripted seeker. Comparisons will be made between the seekers in terms of computational resource usage, as well as the similarity in their behaviors. The seeker’s success rate and average time taken to reach the target uncaught against both agents will be compared to real human players.  
In order to meet my aim, I broke it down to smaller objectives and made sure they were SMART (Specific, Measurable, Achievable, Relevant, and Time-bound). The objectives are as follows:

1. Use online courses, published articles, case studies as well as Unity’s ML-Agents documentation to research and identify a technique for implementing the AI.
2. Establish a set of game rules and agent behaviors in order to achieve the desired ‘hide and seek’ behavior.
3. Build at least 3 game level prototypes on which the training will take place.
4. Define a set of observable parameters for the hider and seeker agents which will be used to compare their performance to human players and a fully scripted agent.
5. Build a FSM seeker, implement and train the hider and seeker agents using the rules and behaviors defined in objective 2.
6. Evaluate if, and to what extent the behaviors of human players and FSM AI can be reproduced using Machine Learning agents using the observable parameters defined in objective 4.

**Background**

|  |  |
| --- | --- |
| **The citation** | **Summary and relevancy to the project** |
| Unity ML-Agents Documentation [1] | The documentation for Unity’s ML-Agents toolkit. The documentation has helped me to set up a learning environment, apply the desired technique for training the agents, as well as answer any questions about the toolkit. |
| *An Introduction to Unity ML-Agents* [2] | A free Unity ML-Agents course with several tutorial projects to follow. These example projects have introduced me to ML-Agents and taught me how to train agents on my own. |
| *Traditional AI Vs. Modern AI* [3] | A great blog post about the evolution of Artificial Intelligence which helped me find out what changed the course of AI and triggered a movement towards Machine Learning. |
| *Finite State Machine: How It Has Affected Your Gaming For Over 40 Years* [4] | This source has given me great insight into the history of Artificial Intelligence and the emergence of Finite State Machines and how they are still used in modern video games. |
| *Human-Like Playtesting With Deep Learning* [5] | This paper presents an approach to learn and deploy human-like playtesting in computer games based on deep learning from player data. This paper gave me insight into machine learning playtesting and how to make my approach different. |
| *OpenAI Five* [6] | This blog introduces OpenAI Five – a team of five neural networks that can defeat professional human teams at Dota 2. This blog introduced me to the idea of using Reinforcement Learning and Proximal Policy Optimization to train my agents. |
| *Finite State Machine: How It Has Affected Your Gaming For Over 40 Years* [7] | This source has given me great insight into the history of Artificial Intelligence and the emergence of Finite State Machines and how they are still used in modern video games. |
| *Emergent Tool Use From Multi-Agent Interaction* [8] | OpenAI’s paper on multi-agent interaction which looks at the emerging agent strategies in a simple hide-and-seek environment. This paper on multi-agent competition was the first inspiration for my project. |

**Timeline, work plan**

**Explanation of work plan**

Before starting the implementation, I created a GANTT chart to layout the tasks ahead. The chart highlights the core stages of the project over the course of the year. The process starts with the research phase. The bulk of the research has been done by January 2021, meaning the completion of objective 1. The project coding phase spans over 3 months from December to February. So far, I have identified the software, tools, training strategy, observable parameters, game rules and agent behaviors, marking the completion of objective 2 and 4. I have built a simple FSM agent and currently I am finishing up the training of the seeker agent, almost completing objective 5. Further training will need to be done on two other level prototypes to meet objective 4 and begin the evaluation process for objective 6.

**Risks and Contingencies**   
The initial project risks included me not having any experience with Machine Learning and the Unity toolkit, not being able to finish training in time and not being able to gather enough human playtest data to then effectively evaluate my agents. The graph includes contingency plans in case a task takes longer than expected or I run into issues. The dissertation writeup stage is scheduled to be completed by April, giving me 30 days of leeway before the submission deadline in May. However, I currently feel very confident in my progress and the amount of time I have left. I believe that with the help of other students I will find enough players to gather enough data for an accurate comparison.

**References**1. Unity-Technologies. 2017. *Unity ML-Agents Toolkit*. [online] Available at: <https://github.com/Unity-Technologies/ml-agents/tree/master/docs> [Accessed 7 December 2020].

2. Simonini, T., 2020. *An Introduction To Unity ML-Agents*. [online] towardsdatascience. Available at: <https://towardsdatascience.com/an-introduction-to-unity-ml-agents-6238452fcf4c> [Accessed 7 December 2020].

3. Bajwa, A., 2019. *Traditional AI Vs. Modern AI*. [online] Medium. Available at: <https://towardsdatascience.com/traditional-ai-vs-modern-ai-5117b469a0c9> [Accessed 15 January 2021].

3. Buttice, C., 2029. *Finite State Machine: How It Has Affected Your Gaming For Over 40 Years*. [online] Techopedia.com. Available at: <https://www.techopedia.com/finite-state-machine-how-it-has-affected-your-gaming-for-over-40-years/2/33996> [Accessed 15 January 2021].

5. Gudmundsson, S., 2018. *Human-Like Playtesting with Deep Learning*. [online] ResearchGate. Available at: <https://www.researchgate.net/publication/328307928\_Human-Like\_Playtesting\_with\_Deep\_Learning> [Accessed 23 January 2021].

6. OpenAI. 2017. *OpenAI Five*. [online] Available at: <https://openai.com/blog/openai-five/> [Accessed 23 January 2021].

7. Baker, B., Kanitscheider, I., Markov, T., Wu, Y., Powell, G., McGrew, B. and Mordatch, I., 2020. *Emergent Tool Use From Multi-Agent Interaction*. [online] OpenAI. Available at: <https://openai.com/blog/emergent-tool-use/> [Accessed 7 December 2020].

8.elementsofai. 2020. *A Free Online Introduction To Artificial Intelligence For Non-Experts*. [online] Available at: <https://www.elementsofai.com> [Accessed 7 December 2020].