

Faculty of Science, Technology and Arts

**Department of Computing  
Project (Technical Computing)  
[55-604708]  
2019/20**

<b>Author:</b>	Sam Bennett
<b>Student ID:</b>	B6023435
<b>Year Submitted:</b>	2020
<b>Supervisor:</b>	Alessandro Di Nuovo
<b>Second Marker:</b>	Andrea Redhead
<b>Degree Course:</b>	Computer Science for Games
<b>Title of Project:</b>	Exploring artificial intelligence in video games

**Confidentiality Required?**

NO ☒

YES ☐

## Abstract

This project will explore artificial intelligence (AI) in video games to understand the knowledge thus far on the topic, researching how video games have changed with improvements and lower technical limitations.

The main body of research will be conducted through researching existing texts and products such as video games. Additionally, by developing a miniature game in a popular game engine to further understand the limitations of developing AI that a company may face.

By researching existing products and texts, this will allow for a well-rounded understanding and fair judgement of the current position of artificial intelligence in games which will be further reinforced by experiencing developing AI for games using a popular game engine.

There are some matters that this methodology may not help to explain. These might include the use of other game engines and strict deadlines that programmers might face in the games industry. As there will be multiple developers working on a game of a much larger scale, which I would be unable to replicate in this timeframe.

# Contents

Abstract.....	ii
Investigation.....	5
Aim's & Objectives .....	5
What is Artificial Intelligence? .....	5
Project Background.....	5
Research.....	5
Artificial Intelligence in Games .....	5
Progression of artificial intelligence in games .....	6
Artificial Intelligence methods .....	6
Design.....	7
Tools and services .....	7
Development.....	8
Visual.....	8
Logical .....	8
Testing.....	10
Evaluation .....	12
Critical Evaluation .....	12
Conclusion.....	14
Bibliography .....	16
Appendices.....	17
Appendix A – Project Specification .....	17
General Details.....	21
2. Research with Human Participants .....	23
3. Research in Organizations.....	24
Appendix B – Sourced Assets .....	2
Appendix C – Level Design .....	3

Appendix D – Firing Range .....	3
Appendix E – Finite State Machine .....	4
Appendix F – Pathfinding .....	5
Appendix G - Item pickup and Inventory .....	6
Appendix H - Win/lose condition .....	7
Appendix I- Player movement .....	8
Appendix J – Keybinds.....	9
Appendix K – Main Menu.....	10
Appendix L – Loading Screen .....	10
Appendix N – Survey Results, Played the provided build .....	<b>Error! Bookmark not defined.</b>
Appendix O – gaming experience rating.....	<b>Error! Bookmark not defined.</b>
Appendix P – Interesting NPCs.....	<b>Error! Bookmark not defined.</b>

# Investigation

## Aim's & Objectives

Recreate a level of a video game that showcases techniques displayed in video games on a simpler scale. Combining and demonstrating pathfinding, player detection, and state machines to create a believable enemy character that is used to add an element of gameplay to a level. The capabilities of the agent will be displayed through visual representations toggleable through the user interface. This allows for the ability to showcase the effect that the agents have on the gameplay and visualize what they are doing programmatically.

This will be achieved by using a popular game engine, creating a singular level with a simple objective that leaves the player to explore and experience the AI without time restrictions and other gameplay features such as narrative, listed objectives, a mini-map etc. This will enable the player to explore the game world and find out for themselves what they need to be doing to create an enjoyable experience. Although, due to time limitations, the deliverable of this project would be on a simpler/smaller scale of that found in production due to development team and games are developed over 1-2 years.

## What is Artificial Intelligence?

Artificial intelligence (AI) is a wide-ranging branch of computer science with the intention of making a computer able to perform tasks that would typically require human intelligence. (Artificial Intelligence, n.d.) In video games, AI varies massively from game to game.

An example of good AI in video games would be by Turn 10 who developed an AI for Forza Motorsport 5 that uses machine learning by taking human players data and applied it to AI, fed through a virtual controller to recreate lifelike opponents. (Staff, 2014) This creates a realistic environment as if the player were playing against other players. Emulating other player's behaviours and habits good and bad, which reduces the need to give the AI an unfair advantage to compete with the player which occurs in other games.

On the other hand, in older games such Need for Speed Underground 2 (2004) by Electronic Arts (EA), did not have access to streaming data through the cloud and the consoles were far more technically inferior in comparison. Used simple AI that never let the player get too far ahead or fall too far behind, creating the impression that they are always "lurking". (Reed, 2005)

## Project Background

An example of AI being used to create an exciting environment is Left 4 Dead (2008) by Valve Corporation. As they model the player's stress levels, factoring in information such as ammo, health, zombie population and density, and adjusting how the zombies attack to create a unique and diverse experience. (Moss, 2016) Whereas this is an extreme case in terms of scope compared to the aims of this project, it perfectly demonstrates the ability that the AI can have over the gameplay and enjoyability of a video game.

As video games become more popular and technological limitations become laxer, having a fundamental understanding of the purpose of AI is becoming more important. Despite being able to use techniques such as a neural network, they may not be the optimal method of approaching an issue. Knowing simpler, less resource and time dependent methods can lead to a more reliable AI. An example of this is in Forza Horizon 4, where most players are voicing their concerns that the AI is too difficult. Reporting the AI is going faster than the car should be able to go, faultless driving, and so on. On the other hand, people also report that the AI can throw themselves into someone's garden and not be seen again. (Griffin, n.d.)

## Research

### Artificial Intelligence in Games

Most games have some form of AI in their games. This could be as simple as the opponent paddle in Pong, which can be capable of learning how to play using a neural network reading screen data (reinforcement

learning) alone (Karpathy, 2016) Though, AI doesn't have to be restricted to just enemies or opponents. Games are capable of creating environments using AI, commonly referred to as procedural generation. This creates an element of difficulty in designing the gameplay experience as each level provides a unique experience. In Warframe, Digital Extremes manage this with a variable called Influence. As players move down a hall, their line of sight will move, creating influence. This is used to ensure enemies are always spawned in the direction the player is travelling, which is used to calculate the pace of the game. (staff, 2013)

In some situations, rather than assigning each agent their own independent AI that will seek out the player and try to complete their own objectives. It may be beneficial to have an invisible AI overseer that commands all the agents to working together as a unit. Dishonoured 2 did this by assigning a member a commander of a small group of agents that would then work together to corner and trap the player. Something individual guards acting on their own motives would not be capable of carrying out. (Davenport, 2017) This demonstrates that every agent in a game doesn't need to be amazing, but the ability to combine guards to make them work together creates a much more humanlike approach to the encounter than any individual agent would be able to carry out without cooperating with fellow agents.

### Progression of artificial intelligence in games

One of the earliest examples of artificial intelligence used in games are in arcade games where stored patterns were used to direct enemy movement. (Pearson, n.d.). As games have developed into what they are today where games such as Forza are using cloud-based machine learning techniques to create more realistic AI opponents that behave like humans. An example of this is Microsoft recording telemetry from the player's gameplay about what car was used, and the car was driven. (Orland, 2013) On the other hand, as games become more technically advanced and the internet is available to over 4.33 billion people. Some developers are resorting to online only models, such as Apex Legends, Evolve, Fortnite, etc. Games like these have very limited AI or none as they are utilising other people to be the opponents. This has benefits as the resources spent on AI can be redirected into gameplay or visual features; but staggers the progress of AI. (Loomis, 2015)

### Artificial Intelligence methods

#### Pathfinding

At its simplest level, "Pathfinding is the process of moving the position of a game character from its initial location to a desired destination." (Seemann & Bourg, 2004) In video games, this also means to avoid obstacles cleverly and calculate the most efficient path when navigating difficult terrain. This demonstrates the demand for more efficient Pathfinding algorithms to be developed, or A\* to be improved to satisfy the requirements of the games industry as technology becomes more advanced. (Cui, 2011) Games often want to find the most effective way of travelling from the starting position to the desired destination taking into account the shortest distance and the fastest travel time where obstacles are present. (Red Blob Games, 2016) There are multiple algorithms that can be used to achieve this mechanic such as Breadth First Search, Dijkstra's Algorithm and A\*.

#### Navigation Mesh

A navigation mesh (NavMesh) is a data structure which is built into the level geometry and defines surfaces in the game world that are walkable, this allows the agent to find a path from one location to another. In flat environments, a 2D raycast or laser scan can be used to safely perceive obstacles. (Pütz, Wiemann, Sprickerhof, & Hertzberg, 2016) Before navigation meshes, we had waypoint networks which were more efficient when resources were more valuable. (MiiMii1205, 2018) As technology as advanced, it is possible to use NavMeshes to achieve a more natural movement through environments at the cost of efficiency in return for more flexibility. Although it has a negative impact on performance compared to waypoint networks. A network dense enough to provide the same flexibility and functionality of a NavMesh would be much more inefficient than a NavMesh. Therefore, it is the preferred approach to navigating rough terrain. Although in games such as Real Time Strategies (RTS), it may be preferable to use other search algorithms such as A\* to navigate through a tile

based map, but this should be evaluated on a per-game basis and research the requirements and flexibility of the AI.

## State Machines

Finite State Machines (FSMs) are one of the most used AI techniques in video games as they are easy to understand, debug and program. (Rabin, 2002) An FSM is a defined number of states that the agent can transition between; the Agent can only be in one given state at any time. The transitions between states are defined based on conditions such as acquiring a line of sight with the player or the agent is close enough to the player to attack them. (Chaudhari, 2018) Finite State Machines are used in almost all games and combined with other mechanics such as pathfinding, machine learning etc so that the developer can maintain areas of the code and maintain a structure so that the code can be debugged and understood.

Expanding on Finite state machines are Fuzzy state machines (FuSM). Which process input and outputs to create fuzzy values. This has the effect of reasoning with degrees of truth to provide a more humanlike behaviour. (Champandard, 2004) Unlike Finite state machines, FuSMs can be simultaneously in both the on and off states. In video games, this means that the non-playable character does not have to be happy with the player. They can be pleased, very happy or delighted with the player, enabling the non-playable character to behave differently in each situation. (Sweetser & Wiles, 2002)

## Design

### Tools and services

The deliverable was created in Unity, using version 2019.2.10f1. Unity was chosen as the preferred game's engine of choice due to its high-quality documentation, free art assets, and online resources available to study. Compared to engines such as Unreal, which documentation is difficult to understand, has a reduced quantity of tutorials available online, and the asset marketplace isn't as established as the Unity asset store, which would result in a lot of time spent on creating assets for the environment. Other engines such as Amazon Lumberyard exist but have similar issues as Unreal.

As Microsoft's Visual Studio has built in Unity support, that was chosen as the integrated development environment (IDE) of choice. Also, previous programming experience in Visual Studio would save time due to knowing shortcuts and the workflow of the program. Other IDEs would have been just as suitable. Such as Sublime Text Editor, Visual Studio Code, Atom etc. These would have been suitable alternatives; except they have additional steps to consider in the setup process and result in a slight advantage as the programs are more light weight than Visual Studio.

Most art assets for the deliverable were sourced through the Unity asset store for free. Though some areas of the environment needed specific assets to complete the look and feel. To develop these assets, Autodesk's 3DS Max was used to create these custom assets. Whereas programs such as Blender and Maya could also be used to create these assets, there was no advantage/disadvantage to using a specific software. So 3DS Max was used as the software of personal preference.

Some assets were modified using Adobe Photoshop to either make textures for the custom models or alter the materials of sourced assets to incorporate them into the environment more. The tools available in Photoshop made it simple to apply masks and re-colour models to make them fit the colour scheme of the level.

Steamlabs OBS, Adobe After Effects 2020 and Adobe Media Encoder 2020 were used to record, edit, and render the demonstration video for this project. This software was chosen due to its reliability and experience in use from personal hobbies. Alternative software is available such as directly recording from a video capture card such as the Elgato HD60 S. Video editing software such as Adobe Premier Pro would also be a viable alternative

to editing video footage. Although as stated previously, After Effects was chosen to perform this task due to it's easy to use interface and YouTube tutorials to assist with any features that weren't so easy to achieve.

## Development

### Visual

After researching colour theory, a flat user interface (UI) design approach was used for the UI of the game level. Using this palette (<https://flatuicolors.com/palette/gb>) as guidance to create a minimalist UI that fit with the look and feel of the game. The panels used as the background image of the UI screens are sourced from <http://kenney.nl/>. These sourced assets allowed for a minimalist and sophisticated UI which assisted in pulling the game together between scenes.

The game world is composed of assets from the Unity asset store and models that was created using 3DS Max. To distinguish sources assets in the deliverable, they were placed in a folder under the path "Assets > Sourced Assets" and broken down by source. See appendix B for a detailed breakdown of sourced assets used in this project.

The level was developed for the most part using the sci-fi modular pack to maintain the look and feel of the game throughout the level. Each room perceived as having a function to make them memorable and relatable by a name (see appendix C). With additional models developed to assist in portraying the function of the room. Such as the firing range (see appendix D), where most of the models were custom created to achieve the desired affect whilst taking heavy influence from the pre-existing structure.

Animations were implemented to make it clear to the player what the agent was going. Making them walk, idle and die created the impression that they were actual enemies rather than just static objects. The animations were provided by sourced assets but were very limited and basic as it was a free sample of the actual product. There was no walk animation provided in the sample, this led to the animations being manipulated slowing down the run animation to create the impression that the agent would walk between nodes.

The animations were then controlled by the finite state machine implemented on the agents. To give them an element of personality. This could have been further developed to give the agents voice lines, footsteps, and random deviations in the agent's patrol pattern to make the gameplay more interesting. Although, this was not possible due to the time restrictions of the project and a lack of resources as I do not have the equipment to record sounds in a reasonable quality to improve the quality of the game.

Rendering the path that the AI is taking was implemented to give the player warning if a guard was approaching from a blind corner as well as demonstrate the path the AI would take to reach its objective. This allowed the player to avoid frustration due to poor timing by walking around a corner at the same time as a guard. This also allowed to test the quality of the pathfinding on the NavMesh, ensuring objects were tagged correctly and reduced the time it would take to test the guards to make sure that they were functioning as intended. As a result of this, it created a more pleasurable playing experience, at the cost of immersion as the player can see where the guard is going to go and plan their actions accordingly, but avoid stressful situations where the guard would just appear in front of the player causing them in losing the game due to sheer misfortune.

### Logical

The concept of the deliverable was to develop an AI that would block the player from easily collecting the objectives in the game world. This demonstrates that the AI are vital parts to a game's playability and entertainment value that they provide. Without the AI agents, the player would randomly traverse the game world until they found the items that they were looking for and the experience would be plain. The implementation of the AI allows for an element of difficulty by adding obstacles between the player and the objective for the player to overcome. The intention of this is to create a more enjoyable experience for the player.



A finite state machine was implemented for the agents (See appendix E). This was to control the guards behaviours in a simple way that a developer can follow and understand, creating a foundation to create more complex functionality on top. From a development perspective, this allowed for the state of the agent to be displayed above their head to show what state they were in and locate errors in functionality easily based on their state and observing their behaviour.

Unity's built-in pathfinding was used to create the pathfinding of the guards, utilising Unity's Navmesh and NavMeshAgent's destination features (See appendix F). This allowed to take away the pressure of developing a full pathfinding system for the AI of the game. As this would be a very lengthy and time-consuming process when a functional system with the same results is already in place. A benefit from developing a pathfinding system would allow the path rendering functionality to render the entire circuit of the pathfinding. But this would have negative performance implications on the game as it would have to do additional calculations to achieve the desired result.

The game utilises the ability for the player to pickup an object by casting a raycast on an object layer just in front of the player to check for an acquirable object. The object that can be picked up contains a trigger volume that will trigger the UI to show a prompt to let the player know that they are in range of an object that they can interact with (See appendix G).

If the raycast successfully hits the object, it will activate the pickup script that will create a UI element that will add the object to the inventory (a list of game objects), and then destroy the game object to stop it being picked up multiple times. This gives the player a visual representation of the items that they have collected throughout the playthrough. The inventory is designed with three slots to imply that there are three items that the player must collect to complete the level. This reduces the amount of clutter required for the user interface as it does not have to explicitly explain that the player needs three items to complete the level.

To expand the Inventory, the code can be set to add/remove the Inventory slots to accommodate more/less items. This would be done on the start of the specified level by getting the items in that level and adjusting the UI elements to accurately reflect this. As the code generates a UI element from the object that is collected, no additional information is required to be added to the pickup or player scripts. This requires an extra few seconds on the loading time but saves time and file-space as only one inventory screen is required and it is flexible enough to suit the requirements of the designer.

By collecting the 3 items and taking them to the AI core, the player can re-program the guard's AI and return them back to their original, friendly state. Saving the station and winning the level. The player will be presented with a win screen where they can choose to play the level again or exit to the main menu (See appendix H). This pulls together the gaming experience and brings it to a close, otherwise the player would be left to wonder the level with nothing to do.

The player loses the game by being caught by a guard. This makes the player be more careful when trying to look around for the objectives. As guards are placed to guide the player through the level and deter them from going into certain areas of the map as there is not a collectible in that area. An example of this would be the firing range, as there is a large quantity of guards in this area, it should deter the player from entering that area. But the areas are still made accessible so that there is not an objective in every room as this would create a boring experience as you easily discover every object if you stay in the room long enough. When the player is caught by a guard, a game over screen is displayed to inform the player that they have been caught and it is game over. They are then given the option to restart the level or exit back to the main menu.

Although the concentration of the project is based on the artificial intelligent agents residing in the map and their affect on how the player interacts with the game world. The player must be able to navigate the game world in a way that suits them (see appendix I). They have been provided with an arsenal of movement abilities that are found in most modern first-person games such as jumping and sprinting. This allows the player the freedom to play the game in a way that suits them and allows much more possibilities in ways to approach the game as they are able to escape guards and jump over objects to escape them.

The guard moves at the same speed as the player when the player is not sprinting. This is to create an element of difficulty so that the player can easily be caught by the guard if they are not paying attention. In some areas of the game world, there are not many objects that the player can jump over to escape the player. This would create a stressful experience as they would be chased by the guard, that would slowly gain on them as they would be taking the most efficient path to the player to catch them. In order to give the player a chance of escaping the guard, they are able to sprint, giving them an increased forward movement speed so that they can get away a break the guards line of sight. Although the sprint speed is not a massive boost, combining this with the ability to jump should allow the player to outrun the guard enough to find somewhere to hide without running into other guards and finding themselves trapped without an escape.

Every player experiences a game differently and as a result, have different preferences in what keys they press to achieve a desired result. To make the game accessible to all, all keys used in the game are stored in a text file. Allowing the player to change and save their preferred keybinds to a file that will persist throughout playthroughs. This allows the game to be more accessible to players that are left-handed and prefer a more ergonomic setup.

On launch, the game will check the root build folder for a keybinds.txt file. If one does not exist, it will create one using the default keybinds. If one is found, it will clear the keybinds dictionary and replace it with the custom defined keybinds that the player has created. Each key has been given a reasonable name for easy reference.

The main menu (appendix K) allows the player to choose between a demonstration of the guard mechanics and the actual game itself. This can be used to test the guards for errors and bugs in a small, controlled environment which exposes issues with the guard agents. Then the game world allows the player to load into the actual game and experience the game as it has been designed to be played. Dropped into the world filled with evil guards, left to figure out how to beat the game. Causing the player to explore the map aimlessly until they can piece together the puzzle to solve the level.

The loading screen was implemented to visually represent that the game level was loading (appendix L), and the game was still responsive. As without it, the screen would just freeze until the game has loaded and they would just pop into the game world. Although the loading times are currently very short, it is scalable and would become increasingly important as the game level increases in complexity and size.

## Testing

### Game build issue

During testing a bug was encountered between playing in editor and a build of the game. The AI agents were not behaving the same way in the game build as it was in the editor. This issue required a development build, which required the installation of a package in Visual Studio that was not previously installed. Once installed, running a development build explained that the agents could not find a reference to the player. This was resolved by replacing the code of the agent from `"GameObject.FindWithTag("Player");"` to creating a public variable and assigning the player to the agent through the inspector in Unity.

### Line Rendering

The development of rendering the path of the guard agent to the player was increasingly difficult. As the destination of the guard was generated at real time. Due to this, the guard had to generate the path that it was

going to take, then a line would have to be drawn from the starting position to the destination of the path finding node, taking into account all of the corners (if any) along the way.

The first implementation created a line from the start position to the last position using `Debug.DrawLine()`.

The second implementation, I added the line renderer component which was set to world space which created a line from the player to the local position of the destination, but in world space so it went off miles into the air to the wrong location.

The third implementation, after a few parameters were modified in the inspector to use local space instead of world and the code was designed to iterate through the corners to create the path that the agent would take.

### Agent issues

Many issues were encountered when developing the agent behaviour that only exist in the game world. The guard struggles to detect and chase the player. When the guard successfully detects the player, they will give up after a few seconds and go to another guard's patrol node. This gives the impression that the guard is losing reference to the player in the game world.

When the game was tested in the Unity editor. The agents work as intended and do not have any unintended behaviours. But in a build, they act in an unpredictable manner. Building a development build of the solution immediately exposed an error with the lineRenderer stating "LineRenderer.SetPosition index out of bounds" which did not exist in the editor solution.

This error was resolved by setting the position count of the line renderer of the Guard line renderer before iterating through the corners. Although this did not resolve the erratic behaviour of the guards. As this did not expose the issue, this raised further questions as the AI behaved as expected in the demo as it does in the game level. As time was a very restrictive factor on this discovery, completely rewriting the pathfinding algorithm was not possible in the remaining time frame. As a result, I decided to upgrade the Unity project from version 2019.2.10f1 to version 2019.3.9f1 in hopes that the build and editor would produce the same result so that if the issue remained, it could be debugged and solved. Upgrading the solution to a newer version of Unity did not result in any changes. As a result of this, it would be required to rewrite the AI class that controls the pathfinding script. As it works in the editor, there would be no way of testing the method until it was complete. Due to the pathfinding working in the demonstration, it is clear that Unity was having some referencing issues dealing with multiple patrol guards. As a result, the guards were left as they were as I would not be able to test a new system within the remaining time frame.

This issue was resolved by adding more agents to the demonstration scene. The multiple agents worked together in the demonstration scene to deliver the desired result. This disclosed that the error was in the Game scene; rather than the AI itself. As a result, the demonstration scene was cloned, and the game was slowly copied into the new cloned scene. When this was completed, the agents worked as intended and the deliverable was functional.

### Cursor visibility

As development was nearing completion, it was observed that the cursor was always still visible. This led to a cursor moving around the screen as the play is trying to navigate the game world. This was easily fixed by setting the visibility of the cursor to true or false depending on the active screen. If the user was in the game or demo world, the cursor was false. Then in all other scenarios, it was set to true. This allowed the user to easily interact with the user interface menus whilst being able to play the game without the distraction of the mouse cursor.

It was discovered that in setups with multiple monitors that the cursor was not locked when the player was in a game scene, so they would see their monitor appear on other screens as they moved around the level. As this

was a minor issue and did not affect gameplay in any way. It was decided that this issue was too minor to be fixed as it had no impact on single-screen setups which was one of the limitations of development.

## Evaluation

### Critical Evaluation

To explore the implications and opinions of artificial intelligence in video games, a survey was created to gather information regarding the subject (See appendix M). A deliverable of the game was provided and the option to play the game before participating in the survey. Although, it is understandable that some participants would be wary of running an uncertified .exe file, as a result, an alternate route was provided. This would ask more generic questions that would not refer to the deliverable so that participants could provide a valid contribution without feeling obliged to run the software. The survey was created to inquire what participants believed artificial intelligence contributed to the playability of video games.

To adhere to General data protection regulations 2016 (GDPR), a disclaimer was provided outlining how the data collected from the survey would be processed. This ensured to the participant that their data would not be mishandled or sold to any third parties. By keeping the survey, the participant can freely and openly complete the survey knowing that there is no correlation between their identity and the answers that they have submitted. 100% of participants agreed to the terms of the disclaimer.

8 responses

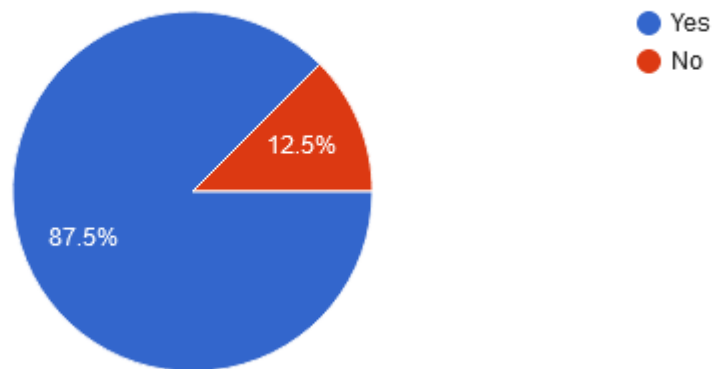


Figure 1: participants that played the game provided in the survey

The first section got to understand the participant, gauging their knowledge of artificial intelligence in video games, their opinions on notable games with interesting artificial intelligence and the conditional question asking if they have played the game which directed the participant down one of two branches. 87.5% of participants that responded to the survey played the deliverable provided in the survey. If the participant did not play the game, asking them about the game would not result in valuable information and it would be a waste of the participants time as they would not be able to answer the questions. To overcome this, two branches were developed. This allows people who did not play the game to voice their opinion and contribute to the research. Also enquires about what kind of game they believe has a notable artificial intelligence. This opens the subject for further study as to why the participant may believe that AI is notable and what features does it do differently compare to other AI to make it stand out. As a result, 100% of participants that did not play the game believed that artificial intelligence has strongly impacted games, thus reinforcing the importance of researching artificial intelligence in video games.

In the portion of the survey where the one participant that did not play the game was asked if they found non-playable characters in video games that they have played to be found believable, they answered yes. Stating

that believable moments, speed and interactions are imperative when creating a believable artificial intelligence. This is also applicable when the player is not directly interacting with the NPC. They should be able to interact with the world around them as the player does. For example, sitting on furniture, turning on/off equipment and lights to assist them in their task and interact with other NPCs.

How would you rate your experience playing this game?

7 responses

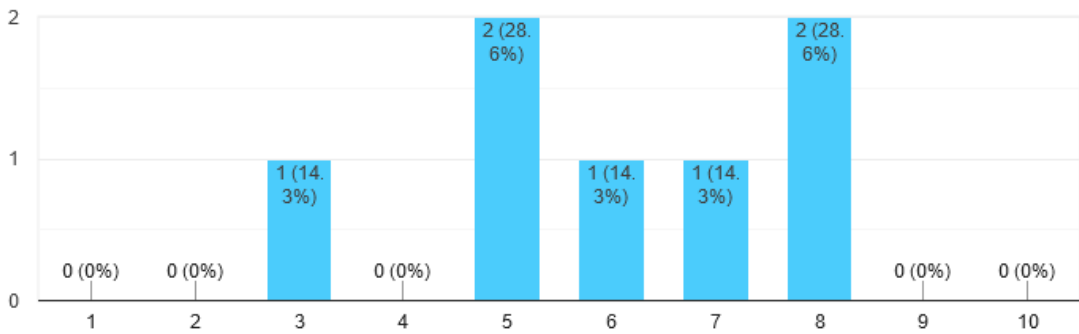
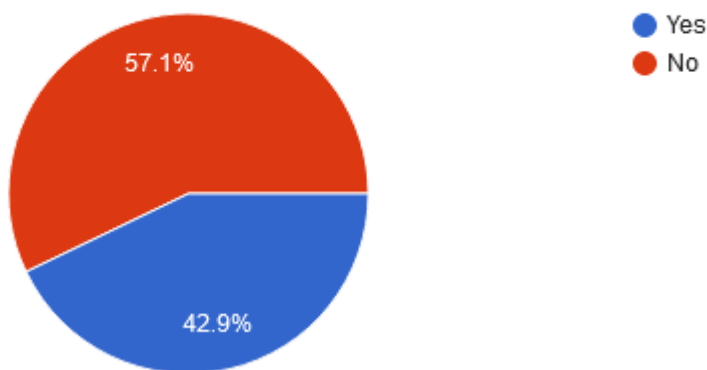


Figure 2: Participants rating the game provided in the survey on a scale of 1-10

The average rating out of 10 for the experience the player had playing the deliverable was 6. Mentioning that the enemy guards added “a sense of urgency and the feeling of impending consequence”, “a level of intricacy that would have otherwise not been there”, and “the guards had to be observed to monitor their patrol routes before the player could safely pass”. This shows that the guards add an element of gameplay that could not be achieved from other methods that do not include artificial intelligence. Although, if the artificial intelligence were developed further by a team of developers, the guards could be made much more intelligent and provide a more in-depth and immersive experience for the player. Such as adding more general AI behaviour and audio prompts that give the player an indirect indication of what is happening in the game.

Were the non-playable characters (NPCs) interesting?

7 responses



57.1% of the participants found the non-playable characters to be not very interesting to play against. Explaining in the follow up question where the participant was asked to explain their answer, participants claimed that the cone of vision was too tight and forgiving and that they patrolled their locations without deviations which made it apparent to the player that the guard was a very simple ai. Despite this, the participants claimed that the guards were a “benign but not stupid adversary”. As a result of this, the guards

could be improved by increasing their cone sizes to factor in peripheral vision, deviations, and voice lines to make the AI more challenging and interesting to overcome.

The game could be improved by implementing a fuzzy state machine rather than the finite state machine that was implanted. This would result in a larger variety of guard behaviours. Reflecting on one participant's response, which claims that the "Variance in the kind of behaviour guards have" would be improved by implementing this method. By adding more functionality into the guard's finite state machine and creating a fuzzy value that would affect how the guard patrolled their route. Including how often they deviated, how far they deviated, if they could interact with missing/nearby guards. This would create more interesting elements of gameplay that would have a positive impact on making the AI appear more humanlike in the way that it behaves.

In the conclusion of the survey, the aim was to inquire about what the participant believes could change about artificial intelligence in video games. When the participant was asked what gameplay elements could be added to video games to improve the 'intelligence' of non-playable characters (NPC), most responses directly related to the results of the survey. Having agents being aware of their surrounding and NPC to NPC interactions to make the NPCs appear as if they are a part of a team that work together to achieve a common goal. It is also observed that participants appreciate it when the NPCs have an understanding of the background/cultural relevance of the game and the characters around them. So that the NPCs 'feel' the deaths of them around them, not just the player.

Participants were asked what they believe is the "next big step" for artificial intelligence in games. The responses from this question were varied. One answer that was voiced multiple times was the implementation of machine learning or neural networks. Although, no information was provided on how or where these methods could be applied video games. It is assumed that similar methods used in Forza where the game acquires telemetry data from the player as they play so that the information can be applied to NPCs to create humanlike behaviour in video games. It was expressed that the rise of multiplayer video games has regressed AI in video games. That developers should take a step backwards instead of forwards and focusing on AI more to create a more immersive experience for the player instead of online-only games such as Fortnite, League of Legends and Overwatch as these games have none or very little/simple AI.

Reviewing the survey as a whole, it provided extremely insightful information that would assist in further developing the artificial intelligence to become more believable in the game world. Including some key features of artificial intelligence that were accidentally overlooked during development. It also opens the eye of the viewer to see how each person identifies artificial intelligence as something slightly different. Some participants believe that the way forward is advanced artificial intelligence methods such as machine learning or neural networks to be implemented into video games, where some participants believe that the way forward is to backtrack on what is already learnt and improve what we already have rather than discovering new methods of completing a task. This response adequately sums up the aim of this project. Do we need faster, more efficient ways of doing the tasks that are already done in video games, or is it safe to use what we have already developed to create a strong fundamental that can be adopted by all developers.

## Conclusion

The objective of this project was to recreate an AI agent in a video game to display how they enhance gameplay. Overall, the agents provide a challenge that makes the environment feel more immersive and engaging. The ability to attack the agents adds a layer of gameplay that was not expected from the initial idea as it opens the level up for different approaches which makes the game more re-playable as a result. If the project were to be repeated, it would be interesting to see what other mechanics could be added to the game world. Such as environmental obstacles that dealt damage, shadows that hid the player, variety of guards with abilities, and adding objectives to guards so that they player had to either kill the guard or steal the object from them when they weren't looking.

At its core, the deliverable expresses the impact that the AI agents have on the game world. As if they were not there, the world would feel empty and bland. Their presence gives the level a sense of purpose and life. The fundamentals explored in this project open many possibilities that the AI agents could be applied to in effort to enhance the gameplay. Combining agents that are able to interact and create obstacles for the player would create a level of immersion that would make the AI appear smart and aware of its surroundings. Examples of this would be lighting torches to reduce the shadows that the player can hide in, notice objects that have been moved or taken by the player and reinforce areas of the level that the player has narrowly escaped from making it more difficult for them to return.

Should the project be reproduced, more time should be spent on the pathfinding of the agents. As it uses the in-built Unity AI which calculates the route between path nodes. Recreating the algorithm using A\* or Dijkstra's search algorithm would allow to compare statistics. Despite this, the concept of the project was to explore how different AI techniques did it take to create a believable AI in a game world. If the project was to be reproduced using Unity's NavMesh, spend the time that was unavailable to this project to amend the problem that persisted in Unity's build that stopped the guards from working within the game world, but not in the editor. This is a problem that may be fixed when a new version of the compiler is created or a new method to control the agent. But as the game is still functional in the demonstration and in the editor, the development of the agent was considered to be successful.

More interesting mechanics and variations of guards would be preferable if the project were to be reproduced. This would create a more diverse challenge for the player, as in its current state. If they are able to defeat one opponent, they can defeat all of them. Enemies that are immune to being attacked or were smaller and faster would create an array of enemies that would keep the player on their toes. Creating a more challenging and interesting environment which would create a more enjoyable playing experience. Alternatively, add obstacles to the environment that the player could interact with such as being able to destroy doors and displace obstacles that would hinder the guards pathfinding or distract them for a period of time whilst they set the obstacles back to how they were.

Going forward in the world of artificial intelligence in games, it would be recommended to cement the fundamentals of the agents as if the foundations of the AI isn't sturdy, it creates unnecessary obstacles when trying to add additional features onto the pre-existing code. As technology becomes more advanced and it becomes faster and easier to implement more intricate and advanced techniques such as neural networks and machine learning into video games. Most of the time, it isn't required or even necessary and the same output and enjoyable experience can be achieved with much easier and simpler techniques like state machines, behavioural trees and simple path finding as the player won't experience as many bugs in the AI's programming as it is easier to debug and there is a higher chance that any issues will be caught during in development or by a Quality Assurance (QA) team.

The next steps if this project were to be continued would be to expand on the functionality of the artificial intelligence. Adding deviations to the patrol guards pathing, interactions with the other guard agents within the level both of a visual level, talking to each other and acting as a human guard would do. But also, an invisible layer that would control the AI and make them work together as one unit to achieve their common goal of catching the player. This would create more interesting interactions for the player to overcome and create a more immersive playing experience as a result.

In summary, the report has briefly scratched the surface of that we refer to as artificial intelligence. The methods that have been developed to create immersive and lifelike behaviours in non-playable characters are invaluable resources of knowledge. The solution to every AI isn't machine learning or neural networks, but a well-designed, simple AI that can perform its task to an exemplary standard of realism can out-perform an AI that has trained itself to complete tasks perfectly with a margin of error.

## Bibliography

- Artificial Intelligence*. (n.d.). Retrieved from builtin: <https://builtin.com/artificial-intelligence>
- Champanand, A. J. (2004). *AI Game Development: Synthetic Creatures with Learning and Reactive Behaviors*. Indianapolis: New Riders. Retrieved from <https://flylib.com/books/en/2.71.1.296/1/>
- Chaudhari, K. (2018, May 25). *AI for game developers: 7 ways AI can take your game to the next level*. Retrieved from packtpub: <https://hub.packtpub.com/ai-for-game-developers/>
- Cui, X. (2011). A\*-based Pathfinding in Modern Computer Games. *IJCSNS International Journal of Computer Science and Network Security*, VOL.11 No.1, January 2011, 125-130.
- Davenport, J. (2017, March 1). *How Dishonored 2's guards came to be driven by a Left 4 Dead-like mastermind AI*. Retrieved from pcgamer: <https://www.pcgamer.com/uk/dishonored-2s-guards-are-driven-by-a-left-4-dead-like-mastermind-ai/>
- Griffin, B. (n.d.). *FORZA HORIZON DEVELOPERS TO SORT OUT THE 'BROKEN' AI DRIVATARS*. Retrieved from drivetribe: <https://drivetribe.com/p/forza-horizon-developers-to-sort-LfTur2zTSY-lp-E0qkXvEg?iid=CgvKwvjqQLixrlhfz8mdmQ>
- Karpathy, A. (2016, May 31). *Deep Reinforcement Learning: Pong from Pixels*. Retrieved from Andrej Karpathy blog: <http://karpathy.github.io/2016/05/31/rl/>
- Loomis, M. (2015, January 27). *The Rise of Online Multiplayer Games And The Collapse of Couch Co-Op*. Retrieved from Gamerant: <https://gamerant.com/rise-of-online-multiplayer-games-couch-coop/>
- MiiMii1205. (2018, April 27). *Navigation Meshes and Pathfinding*. Retrieved from gamedev.net: <https://www.gamedev.net/tutorials/programming/artificial-intelligence/navigation-meshes-and-pathfinding-r4880/>
- Moss, R. (2016, April 5). *7 examples of game AI that every developer should study*. Retrieved from Gamasutra: [https://www.gamasutra.com/view/news/269634/7\\_examples\\_of\\_game\\_AI\\_that\\_every\\_developer\\_should\\_study.php](https://www.gamasutra.com/view/news/269634/7_examples_of_game_AI_that_every_developer_should_study.php)
- Orland, K. (2013, October 10). *How Forza 5 and the Xbox One use the cloud to drive machine-learning AI*. Retrieved from ars technica: <https://arstechnica.com/gaming/2013/10/how-forza-5-and-the-xbox-one-use-the-cloud-to-drive-machine-learning-ai/>
- Pearson, C. (n.d.). *The Evolution of AI in Gaming*. Retrieved from BIG Cloud: <https://bigcloud.io/the-evolution-of-artificial-intelligence-in-gaming/>
- Pütz, S., Wiemann, T., Sprickerhof, J., & Hertzberg, J. (2016). *3D Navigation Mesh Generation for PathPlanning in Uneven Terrain*. n.p.: FAC-PapersOnLine.
- Rabin, S. (2002). Implementing a State Machine. *AI Game Programming Wisdom*, 314-320.
- Red Blob Games*. (2016, June 26). Retrieved from Introduction to the A\* Algorithm: <https://www.redblobgames.com/pathfinding/a-star/introduction.html>
- Reed, K. (2005, May 11). *Need for Speed Underground 2*. Retrieved from Eurogamer: [https://www.eurogamer.net/articles/r\\_nfsu2\\_x](https://www.eurogamer.net/articles/r_nfsu2_x)
- Seemann, G., & Bourg, D. M. (2004). *AI for Game Developers*. Sebastopol: O'Reilly.
- staff, M. (2013, March 25). *GDC 13: Handling AI in procedural levels*. Retrieved from MCV / Develop: <https://www.mcvuk.com/development-news/gdc-13-handling-ai-in-procedural-levels/>



Staff, X. W. (2014, September 30). *Forza Horizon 2: What's a Drivatar, and Why Should I Care?* Retrieved from Xbox: <https://news.xbox.com/en-us/2014/09/30/games-forza-horizon-2-drivatars/>

Sweetser, P., & Wiles, J. (2002). Current AI in games : a review. *Australian Journal of Intelligent Information Processing Systems*, 24-42.

## Appendices

### Appendix A – Project Specification

#### **PROJECT SPECIFICATION - Project (Technical Computing) 2019/20**

<b>Student:</b>	<b>Sam Bennett</b>
<b>Date:</b>	<b>11 April 2020</b>
<b>Supervisor:</b>	<b>Alessandro Di Nuovo</b>
<b>Degree Course:</b>	<b>BSc Computer Science for Games</b>
<b>Title of Project:</b>	<b>Exploring artificial intelligence in video games</b>

#### **Elaboration**

This project will explore artificial intelligence (AI) in video games to understand the knowledge thus far on the topic, researching how video games have changed with improvements and lower technical limitations.

The main body of research will be conducted through researching existing texts and products such as video games. Additionally, by developing a miniature game in a popular game engine to further understand the limitations of developing AI that a company may face.

By researching existing products and texts, this will allow for a well-rounded understanding and fair judgement of the current position of artificial intelligence in games which will be further reinforced by experiencing developing AI for games using a popular game engine.

There are some matters that this methodology may not help to explain. These might include the use of other game engines and strict deadlines that programmers might face in the games industry. As there will be multiple developers working on a game of a much larger scale, which I would be unable to replicate in this timeframe.

## **Project Aims**

- Explore the origin of AI
- Develop an in-depth knowledge of artificial intelligence (AI) in video games and how they have developed over time.
- Understand how AI is implemented into a game to create an immersive experience without being impossible to beat.
- Explore how AI has affected other industries

### Project deliverable(s)

I will deliver a mini stealth game which will have the following features:

- Utilising the game engine's AI utilities to create a traversable environment
- AI agents that patrol an environment in search of the player
- State machine that responds to the player

### Action plan

Name	Type	Deadline
Find a project supervisor	Milestone	11/10/2019
Research current technologies	Task	20/10/2019
Submit project specification and ethics form	Milestone	25/10/2019
Research the origin of artificial intelligence	Task	1/11/2019
Research and justify game engine choice	Task	10/11/2019
Create a basic workspace in chosen engine	Task	12/11/2019
Create outline of a game design document	Task	15/11/2019
Research games with unique artificial intelligence	Task	1/12/2019
Submit Information review	Milestone	6/12/2019
Design, implement and document a level layout	Task	10/12/2019
Implement static AI into the level	Task	12/12/2019
Utilise built-in AI methods to make patrolling agents	Task	14/12/2019
Create player movement and controls	Task	18/12/2019
Research early games with artificial intelligence	Task	18/1/2020
Design and implement objectives and narrative	Task	20/1/2020
Implement player detection method	Task	1/2/2020
Design and implement a user interface	Task	8/2/2020
Implement game screens and transitions	Task	12/2/2020
Create customisable key binds	Task	15/2/2020
Refine the game world	Task	19/2/2020
Provisional Contents page	Milestone	21/2/2020
Reflect on methodology compared to approach taken	Task	27/2/2020
Review report for errors, conciseness and consistency	Task	26/3/2020
Draft critical evaluation	Milestone	27/3/2020
Review report and make amendments	Task	01/4/2020
Submit report to Turnitin	Milestone	22/4/2020
Submit physical project report and deliverable	Milestone	23/4/2020
Demonstration of work	Milestone	Before 12/5/2020

### BCS Code of Conduct

I confirm that I have successfully completed the BCS code of conduct on-line test with a mark of 70% or above. This is a condition of completing the Project (Technical Computing) module.

Signature: 

### Publication of Work

I confirm that I understand the "Guidance on Publication Procedures" as described on the Bb site for the module.

Signature: 

### GDPR

I confirm that I will use the "Participant Information Sheet" as a basis for any survey, questionnaire or participant testing materials. This form is available on the Bb site for the module.

Signature: 

### Ethics

Complete the SHUREC 7 (research ethics checklist for students) form below. If you think that your project may include ethical issues that need resolving (working with vulnerable people, testing procedures etc.) then discuss this with your supervisor as soon as possible and comment further here.

Both you and your supervisor need to sign the completed SHUREC 7 form.

## RESEARCH ETHICS CHECKLIST FOR STUDENTS (SHUREC 7)

This form is designed to help students and their supervisors to complete an ethical scrutiny of proposed research. The SHU [Research Ethics Policy](#) should be consulted before completing the form.

Answering the questions below will help you decide whether your proposed research requires ethical review by a Designated Research Ethics Working Group.

The final responsibility for ensuring that ethical research practices are followed rests with the supervisor for student research.

Note that students and staff are responsible for making suitable arrangements for keeping data secure and, if relevant, for keeping the identity of participants anonymous. They are also responsible for following SHU guidelines about data encryption and research data management.

The form also enables the University and Faculty to keep a record confirming that research conducted has been subjected to ethical scrutiny.

For student projects, the form may be completed by the student and the supervisor and/or module leader (as applicable). In all cases, it should be counter-signed by the supervisor and/or module leader and kept as a record showing that ethical scrutiny has occurred. Students should retain a copy for inclusion in their research projects, and staff should keep a copy in the student file.

Please note if it may be necessary to conduct a health and safety risk assessment for the proposed research. Further information can be obtained from the Faculty Safety Co-Ordinator.

### General Details

Name of student	Sam Bennett
SHU email address	B6023435@my.shu.ac.uk
Course or qualification (student)	Computer Science for Games
Name of supervisor	Alessandro Di Nuovo
email address	a.dinuovo@shu.ac.uk
Title of proposed research	Exploring artificial intelligence in games
Proposed start date	25/10/2019
Proposed end date	22/4/2020

<p>Brief outline of research to include, rationale &amp; aims (250-500 words).</p>	<p>This project will explore artificial intelligence (AI) in video games to understand the knowledge thus far on the topic, researching how video games have changed with improvements and lower technical limitations.</p> <p>The main body of research will be conducted through researching existing texts and products such as video games. Additionally, by developing a miniature game in a popular game engine to further understand the limitations of developing AI that a company may face.</p> <p>By researching existing products and texts, this will allow for a well-rounded understanding and fair judgement of the current position of artificial intelligence in games which will be further reinforced by experiencing developing AI for games using a popular game engine.</p>
<p>Where data is collected from individuals, outline the nature of data, details of anonymization, storage and disposal procedures if required (250-500 words).</p>	<p>Data is collected in an anonymous form (no names, personal details, email address, etc is stored). The participant of the survey is provided with an email where they can request their information can be disposed of. The data is stored in google forms, protected by a password protected account, only accessed from a password protected computer in private residence.</p>

## 1. Health Related Research Involving the NHS or Social Care / Community Care or the Criminal Justice Service or with research participants unable to provide informed consent

Question	Yes/No
<p>1. Does the research involve?</p> <ul style="list-style-type: none"> <li>Patients recruited because of their past or present use of the NHS or Social Care</li> <li>Relatives/carers of patients recruited because of their past or present use of the NHS or Social Care</li> <li>Access to data, organs or other bodily material of past or present NHS patients</li> <li>Foetal material and IVF involving NHS patients</li> <li>The recently dead in NHS premises</li> <li>Prisoners or others within the criminal justice system recruited for health-related research*</li> <li>Police, court officials, prisoners or others within the criminal justice system*</li> <li>Participants who are unable to provide informed consent due to their incapacity even if the project is not health related</li> </ul>	No
<p>2. Is this a research project as opposed to service evaluation or audit?</p> <p><i>For NHS definitions please see the following website</i>  <a href="http://www.hra.nhs.uk/documents/2013/09/defining-research.pdf">http://www.hra.nhs.uk/documents/2013/09/defining-research.pdf</a></p>	No

If you have answered **YES** to questions **1 & 2** then you **must** seek the appropriate external approvals from the NHS, Social Care or the National Offender Management Service (NOMS) under their independent Research Governance schemes. Further information is provided below.

NHS <https://www.myresearchproject.org.uk/Signin.aspx>

\* All prison projects also need National Offender Management Service (NOMS) Approval and Governor's Approval and may need Ministry of Justice approval. Further guidance at: <http://www.hra.nhs.uk/research-community/applying-for-approvals/national-offender-management-service-noms/>

**NB** FRECs provide Independent Scientific Review for NHS or SC research and initial scrutiny for ethics applications as required for university sponsorship of the research. Applicants can use the NHS proforma and submit this initially to their FREC.

## 2. Research with Human Participants

Question	Yes/No
Does the research involve human participants? This includes surveys, questionnaires, observing behaviour etc.	Yes
Question	Yes/No
<p>1. <i>Note If YES, then please answer questions 2 to 10</i>  <i>If NO, please go to Section 3</i></p>	
<p>2. Will any of the participants be vulnerable?</p> <p><i>Note: Vulnerable' people include children and young people, people with learning disabilities, people who may be limited by age or sickness, etc. See definition on website</i></p>	No

3. Are drugs, placebos or other substances (e.g. food substances, vitamins) to be administered to the study participants or will the study involve invasive, intrusive or potentially harmful procedures of any kind?	No
4. Will tissue samples (including blood) be obtained from participants?	No
5. Is pain or more than mild discomfort likely to result from the study?	No
6. Will the study involve prolonged or repetitive testing?	No
7. Is there any reasonable and foreseeable risk of physical or emotional harm to any of the participants? <i>Note: Harm may be caused by distressing or intrusive interview questions, uncomfortable procedures involving the participant, invasion of privacy, topics relating to highly personal information, topics relating to illegal activity, etc.</i>	No
8. Will anyone be taking part without giving their informed consent?	No
9. Is it covert research? <i>Note: 'Covert research' refers to research that is conducted without the knowledge of participants.</i>	No
10. Will the research output allow identification of any individual who has not given their express consent to be identified?	No

If you answered **YES only** to question **1**, the checklist should be saved and any course procedures for submission followed. If you have answered **YES** to any of the other questions you are **required** to submit a SHUREC8A (or 8B) to the FREC. If you answered **YES** to question **8** and participants cannot provide informed consent due to their incapacity you must obtain the appropriate approvals from the NHS research governance system. Your supervisor will advise.

### 3. Research in Organizations

Question	Yes/No
1. Will the research involve working with/within an organization (e.g. school, business, charity, museum, government department, international agency, etc.)?	No
2. If you answered YES to question 1, do you have granted access to conduct the research? <i>If YES, students please show evidence to your supervisor. PI should retain safely.</i>	NA
3. If you answered NO to question 2, is it because: A. you have not yet asked B. you have asked and not yet received an answer C. you have asked and been refused access.  <i>Note: You will only be able to start the research when you have been granted access.</i>	NA

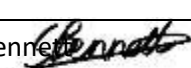
### 4. Research with Products and Artefacts


Question	Yes/No
1. Will the research involve working with copyrighted documents, films, broadcasts, photographs, artworks, designs, products, programmes, databases, networks, processes, existing datasets or secure data?	No



<p>2. If you answered YES to question 1, are the materials you intend to use in the public domain?</p> <p><i>Notes: 'In the public domain' does not mean the same thing as 'publicly accessible'.</i></p> <ul style="list-style-type: none"> <li>Information which is 'in the public domain' is no longer protected by copyright (i.e. copyright has either expired or been waived) and can be used without permission.</li> <li>Information which is 'publicly accessible' (e.g. TV broadcasts, websites, artworks, newspapers) is available for anyone to consult/view. It is still protected by copyright even if there is no copyright notice. In UK law, copyright protection is automatic and does not require a copyright statement, although it is always good practice to provide one. It is necessary to check the terms and conditions of use to find out exactly how the material may be reused etc.</li> </ul> <p><i>If you answered YES to question 1, be aware that you may need to consider other ethics codes. For example, when conducting Internet research, consult the code of the Association of Internet Researchers; for educational research, consult the Code of Ethics of the British Educational Research Association.</i></p>	NA
<p>3. If you answered NO to question 2, do you have explicit permission to use these materials as data?</p> <p><i>If YES, please show evidence to your supervisor.</i></p>	NA
<p>4. If you answered NO to question 3, is it because:</p> <p>A. you have not yet asked permission</p> <p>B. you have asked and not yet received an answer</p> <p>C. you have asked and been refused access.</p> <p><i>Note: You will only be able to start the research when you have been granted permission to use the specified material.</i></p>	NA

#### Adherence to SHU policy and procedures

<b>Personal statement</b>	
<p>I can confirm that:</p> <p>– I have read the Sheffield Hallam University Research Ethics Policy and Procedures</p> <p>– I agree to abide by its principles.</p>	
<b>Student</b>	
Name: Sam Bennett 	Date: 03/10/2019
<b>Signature:</b>	
<b>Supervisor or other person giving ethical sign-off</b>	
<p>I can confirm that completion of this form has not identified the need for ethical approval by the FREC or an NHS, Social Care or other external REC. The research will not commence until any approvals required under Sections 3 &amp; 4 have been received.</p>	
Name: Alessandro Di Nuovo (supervisor)	Date: 11/04/2020

Signature: 

## Appendix B – Sourced Assets

Ball Pack v1.0 – Yougen Tech

<https://assetstore.unity.com/packages/3d/props/ball-pack-446>

Bucky Ball model

Lean Tween v2.50 – Dented Pixel

<https://assetstore.unity.com/packages/tools/animation/leantween-3595>

Tweening script

PBR RPG/FPS Game Assets (Industrial Set v1.0) – Dmitrii Kutsenko

<https://assetstore.unity.com/packages/3d/environments/industrial/pbr-rpg-fps-game-assets-industrial-set-v1-0-146519>

Containers in storeroom

Low Poly Storage Pack v1.1 – Broken Vector

<https://assetstore.unity.com/packages/3d/environments/urban/low-poly-storage-pack-101732>

Middle floor storeroom decoration

Sci Fi Warrior PBR HP Polyart (Free version) v1.2 – Dungeon Mason

<https://assetstore.unity.com/packages/3d/characters/robots/sci-fi-warrior-pbr-hp-polyart-106154>

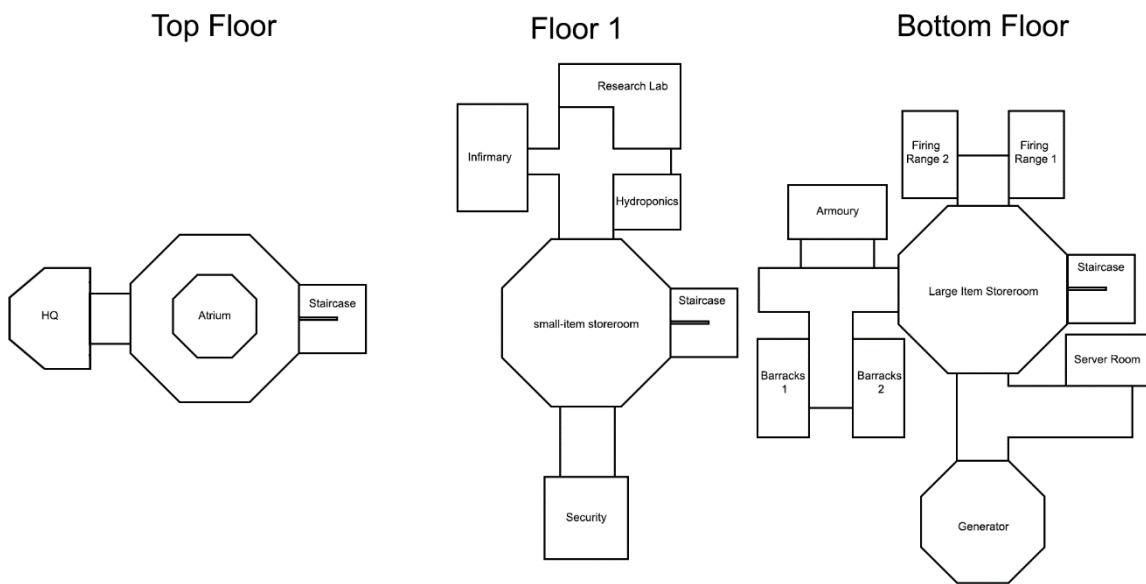
Agent and player model

Sci-Fi Styled Modular Pack v1.1 – karboosx

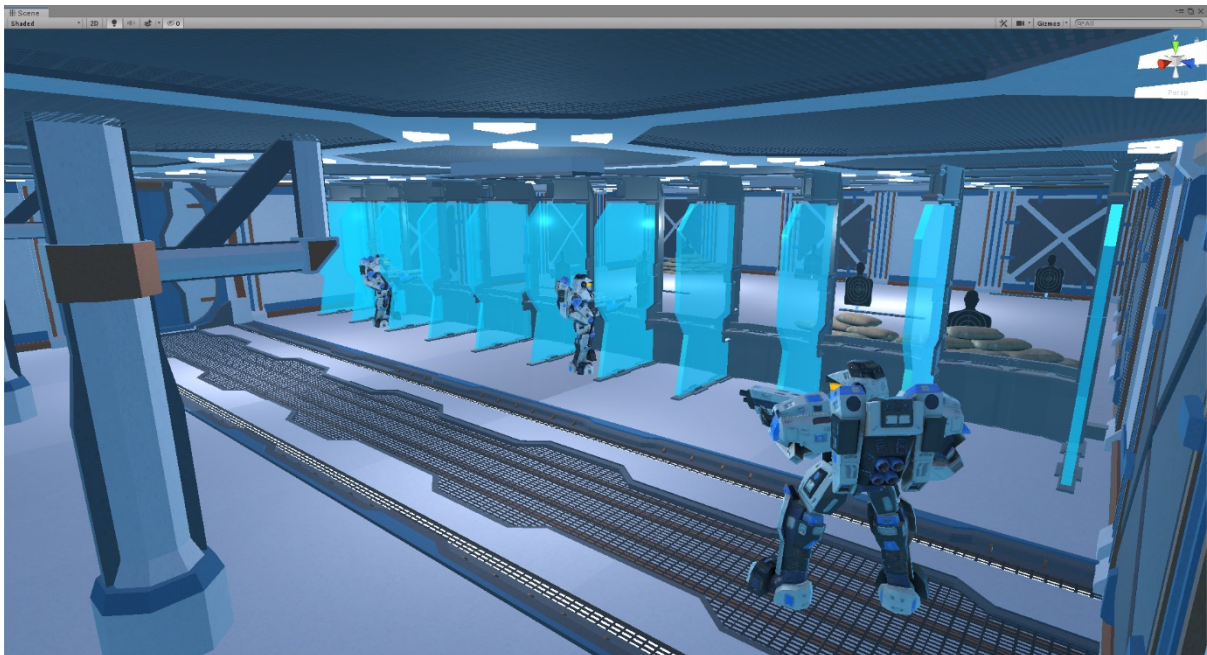
<https://assetstore.unity.com/packages/3d/environments/sci-fi/sci-fi-styled-modular-pack-82913>

Structure of game world and decoration

## Appendix C – Level Design



## Appendix D – Firing Range



## Appendix E – Finite State Machine

```
public enum State
{
    Guarding,
    Patrolling,
    Chasing,
    Searching,
    Alerted,
    Dead
};

switch (currentState)
{
    //Chase the player
    case State.Chasing:
        var playerPosition =
GameObject.FindGameObjectWithTag("Player").transform.position;
        if (agent.hasPath)
            agent.ResetPath();
        agent.destination = playerPosition;
        chasingPlayer = true;
        break;
    //Return to post
    case State.Guarding:
        if (chasingPlayer)
        {
            agent.ResetPath();
            agent.destination = startingPosition;
            chasingPlayer = false;

            StartCoroutine(Wait(3));
        }
        break;
}
```

## Appendix F – Pathfinding

```
private void Start()
{
    //Get reference to the AI Script
    ai = GetComponent<AIClass>();
    foreach (var node in nodes)
    {
        node.gameObject.GetComponent<MeshRenderer>().enabled =
ui.pathfindingNodesVisible;
    }
    //patrol guard so overwrite initial and current state to be patrolling
    ai.SetState(AIClass.State.Patrolling);
    ai.initialState = AIClass.State.Patrolling;
    ...

    //Get Guard's NavMeshAgent Component
    agent = GetComponent<NavMeshAgent>();

    //Start Patrolling
    GoToNextNode();
}

private void GoToNextNode()
{
    if (ai.GetState() != AIClass.State.Patrolling) return;
    agent.destination = nodes[nextNode].position;
    //Update nextNode for next iteration
    nextNode = (nextNode + 1) % nodes.Length;
}
```

## Appendix G - Item pickup and Inventory

```
public void Interact()
{
    const int LayerMask = 1 << 10; //Interactable Objects Layer
    //Raycast forward from the player
    if (Physics.Raycast(new Vector3(transform.position.x, transform.position.y -
0.5f, transform.position.z), transform.TransformDirection(Vector3.forward), out
var hit, 2.5f, LayerMask))
    {
        //Reference the player
        var player = gameObject.GetComponentInParent<Player>();
        //Object to Interact with
        var interactableObject = hit.transform.gameObject;
        //Pickup the object
        if (interactableObject.name == "Objective")
            player.WinCheck();
        else
            player.Pickup(interactableObject);
    }
}
PlayerCameraControls.cs
```

```
public void Pickup(GameObject interactiveObject)
{
    var pickup = interactiveObject.GetComponent<Pickup>();
    var itemContainer = GameObject.Find("Item" + (gameManager.inventory.Count +
1)).GetComponent<Image>();

    //Create an empty GameObject as a child of the Inventory
    var item = new GameObject("Item");
    item.transform.parent = itemContainer.transform;

    //Add a TextMeshPro component to the newly created GameObject
    var itemText = item.AddComponent<TextMeshProUGUI>();

    //Format the text and set the text to the pickup name
    itemText.color = Color.white;
    itemText.text = pickup.name;
    itemText.transform.localScale = new Vector3(1, 1, 1);
    itemText.fontSize = 21;
    itemText.alignment = TextAlignmentOptions.Midline;

    //Add item to Inventory List
    gameManager.inventory.Add(interactiveObject);

    //Clear the pickup prompt
    GameObject.Find("Prompt").GetComponent<TextMeshProUGUI>().text = "";

    //Destroy the pickup so it can't be picked up again
    Destroy(interactiveObject);
}
Player.cs
```

## Appendix H - Win/lose condition

### Win Condition

```
public void WinCheck()
{
    //If we have all of the items necessary to win
    if (gameManager.inventory.Count == 3)
    {
        gameManager.Win();
    }
}
```

### Lose Condition

```
private void OnTriggerEnter(Collider other)
{
    //Lose Condition
    if(other.CompareTag("Enemy"))
        gameManager.Lose();
}
```

```
public void Lose()
{
    loseScreen.gameObject.SetActive(true);
    paused = true;
    Time.timeScale = 0;
}
```

```
public void Win()
{
    winScreen.gameObject.SetActive(true);
    paused = true;
    Time.timeScale = 0;
}
```

## Appendix I- Player movement

```
private void InputHandler()
{
    foreach (var keyBind in Keybinds.KeyBindsDictionary.Keys)
    {
        if (!Input.anyKey)
        {
            //Idle
            anim.SetInteger("AnimationState", 0);
            anim.speed = 1;
        }
        else
        {
            var playerCameraControls = cam.GetComponent<PlayerCameraControls>();
            if (Input.GetKey(Keybinds.KeyBindsDictionary[keyBind]))
            {
                switch (keyBind)
                {
                    //---Movement---
                    case "Forward":
                        anim.speed = 1;
                        anim.SetInteger("AnimationState", 1);
                        if(!bSprinting)
                            currentSpeed = movementSpeed - 2;
                        break;
                    case "Backward":
                        anim.SetInteger("AnimationState", 2);
                        anim.speed = 0.5f;
                        currentSpeed = movementSpeed - 2;
                        break;
                    case "Left":
                        anim.SetInteger("AnimationState", 6);
                        anim.speed = 1;
                        currentSpeed = movementSpeed - 2;
                        break;
                    case "Right":
                        anim.SetInteger("AnimationState", 7);
                        anim.speed = 1;
                        currentSpeed = movementSpeed - 2;
                        break;

                    //---Player Actions
                    case "Attack":
                        playerCameraControls.Attack();
                        break;
                    case "Interact":
                        playerCameraControls.Interact();
                        break;

                    //---Game Management---
                    case "Pause":
                        gameManager.PauseGame();
                        break;
                }
            }
        }
    }
}
```



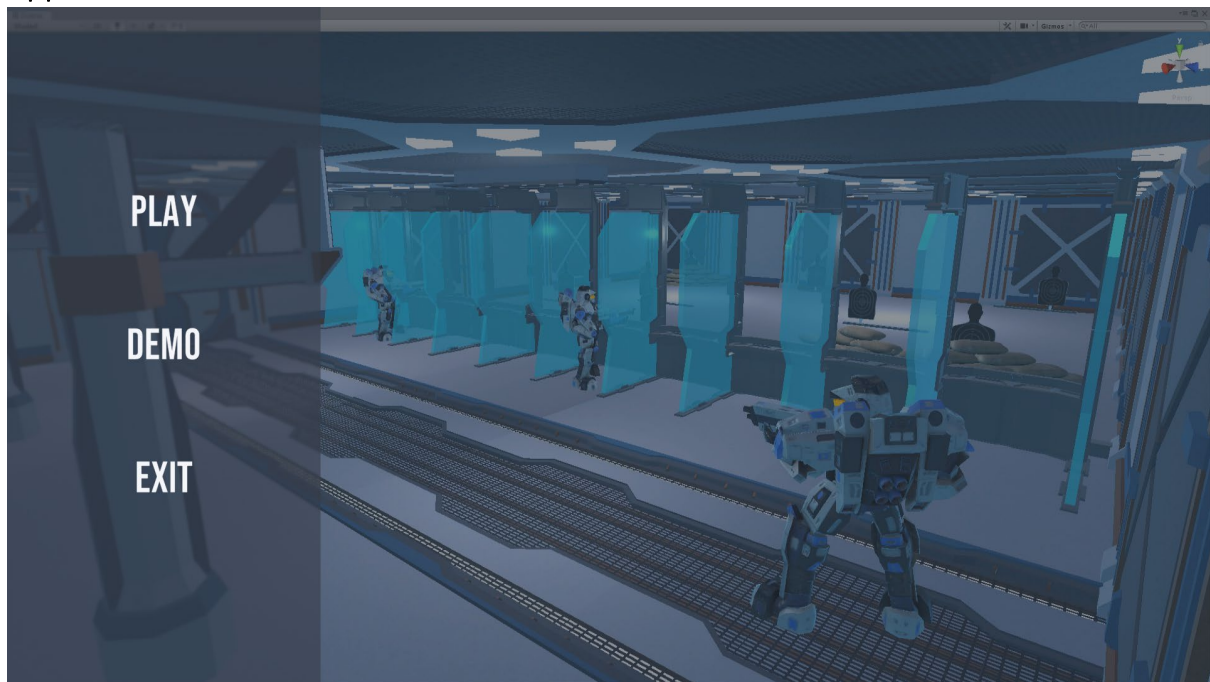
## Appendix J – Keybinds

```
public static readonly Dictionary<string, KeyCode> KeyBindsDictionary = new
Dictionary<string, KeyCode>
{
    { "Forward", KeyCode.W },
    { "Left", KeyCode.A },
    { "Backward", KeyCode.S },
    { "Right", KeyCode.D },
    { "Sprint", KeyCode.LeftShift },
    { "Attack", KeyCode.E },
    { "Pause", KeyCode.Escape },
    { "Interact", KeyCode.F }
};

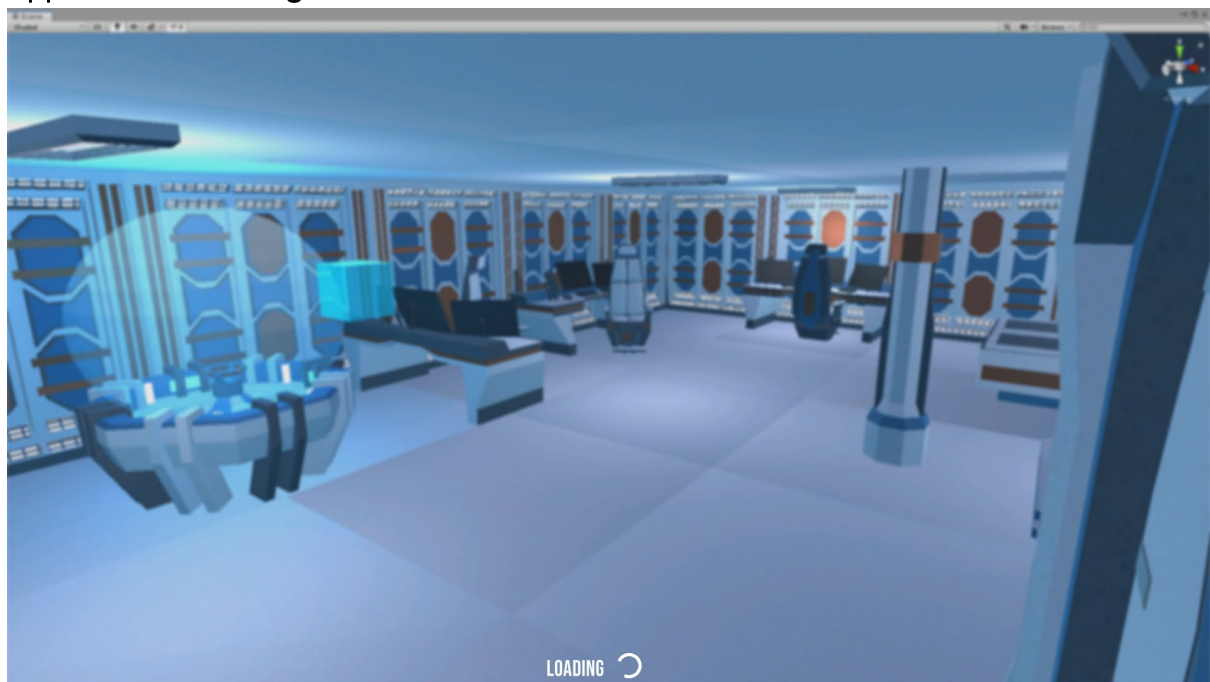
private static void SaveKeyBinds()
{
    //Write all KeyBinds to a text file for them to persist
    var fileText = KeyBindsDictionary.Aggregate(string.Empty, (current, keyBind)
=> current + (keyBind.Key + " = " + keyBind.Value + Environment.NewLine));
    File.WriteAllText("keybinds.txt", fileText);
}

public static void ReadKeybinds()
{
    //If we've got some keyBinds saved
    if (File.Exists("keybinds.txt"))
    {
        //Empty dictionary to avoid duplicate data
        KeyBindsDictionary.Clear();
        //string separator
        const string stringPattern = " = ";
        var sr = File.OpenText("keybinds.txt");
        // Add the file contents to a generic list of strings.
        var fileContents = new List<string>();
        while (!sr.EndOfStream)
        {
            //Create a list containing every line in the file
            fileContents.Add(sr.ReadLine());
        }
        foreach (var keyBind in fileContents.Select(line =>
System.Text.RegularExpressions.Regex.Split(line, stringPattern)))
        {
            KeyBindsDictionary.Add(keyBind[0],
(KeyCode)Enum.Parse(typeof(KeyCode), keyBind[1]));
        }
    }
    else
    {
        SaveKeyBinds();
    }
}
```

## Appendix K – Main Menu



## Appendix L – Loading Screen



Appem

Appe

```
readonly List<AsyncOperation> scenesLoading = new List<AsyncOperation>();

public void LoadGame()
{
    winScreen.gameObject.SetActive(false);
    loseScreen.gameObject.SetActive(false);

    backgroundImage.sprite = backgrounds[Random.Range(0, backgrounds.Length)];
    loadingScreen.gameObject.SetActive(true);

    scenesLoading.Add(SceneManager.UnloadSceneAsync((int)SceneIndexes.TITLE_SCREEN));
    scenesLoading.Add(SceneManager.LoadSceneAsync((int)SceneIndexes.GAME,
    LoadSceneMode.Additive));

    StartCoroutine(GetSceneLoadProgress());

    paused = false;
    Time.timeScale = 1;
}

public IEnumerator GetSceneLoadProgress()
{
    foreach (var scene in scenesLoading)
    {
        while (!scene.isDone)
        {
            yield return null;
        }
    }

    loadingScreen.gameObject.SetActive(false);
}
```

Appendix M – Survey



## Exploring Artificial Intelligence in Games

As artificial intelligence (AI) in video games becomes more advanced. It is important to have an understanding of what is meant by the term artificial intelligence.

This survey is designed to gauge your understanding of artificial intelligence and explore the pros and cons while thinking of possible directions that it could take in the future.

To complete this survey, it is advised that you download and play the game:  
<https://drive.google.com/open?id=1VKloMpA1d7Eg9kOhgSBbIZNzQ2vKqWUe> The game will need to be extracted (file size: 146MB). Then it can be run by opening "Artificial Intelligence in Games.exe"

\* Required

### Disclaimer \*

The information collected from this survey will be used to evaluate the implications of Artificial Intelligence in video games and its impact on the user experience. The data collected from this survey is securely stored in Google Forms in an anonymous form as no personal information about the participant will be collected during this survey. The data will be stored until the end of June this year (2020), when the research period is over. Your data will never be sold or shared with any third parties. If you have any questions or request your data be deleted, contact myself at [Sam.Bennett3@student.shu.ac.uk](mailto:Sam.Bennett3@student.shu.ac.uk).

☐ I accept the terms of this disclaimer

## Background Knowledge

A few questions to gauge your experience with artificial intelligence in video games.

How would you rate your understanding of artificial intelligence in video games?

\*

	1	2	3	4	5	
No understanding	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strong understanding

What is an example of a game with notable artificial intelligence that you have played?

Your answer

Have you played the game provided in this survey? \*

- ☐ Yes
- ☐ No

[Back](#)

[Next](#)



Page 2 of 5

## Artificial Intelligence in Games

How do you think artificial intelligence has affected video games? \*

	1	2	3	4	5	
No impact	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly impacted

Briefly explain your answer \*

Your answer

Do you find non-playable characters (NPCs) to be believable in video games that you have played? \*

- ☐ Yes
- ☐ No

Briefly explain what you believe contributes to a believable NPC. \*

Your answer

Back

Next



Page 3 of 5

If the participant hasn't played the game, they will be displayed this screen

### Artificial Intelligence in This Game

How would you rate your experience playing this game? \*

1 2 3 4 5 6 7 8 9 10

Poor ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ Outstanding

What do you feel the artificial intelligence added to the game? \*

Your answer

What could be improved? \*

Your answer

Were the non-playable characters (NPCs) interesting? \*

☐ Yes

☐ No

Briefly explain your answer \*

Your answer

If they did, they would be displayed this screen

# Exploring Artificial Intelligence in Games

\* Required

## Conclusion of Thoughts

What gameplay elements could be added to video games to improve the 'intelligence' of NPCs? \*

Your answer

What do you believe is the next big step for artificial intelligence in video games? \*

Your answer

[Back](#)

[Submit](#)



Page 5 of 5