**University of Ulster**

**BSc (Hons) Computing Systems**

**COM617 – Project Management**

**Project Management Body of Knowledge**

**Dynamic cover-based system for Unity Game AI Players**

**Student Name: Groza Leontin**

**Student Number: 8888888**

**Supervisor: Tendai Mhlanga**

Table of Contents

[PMI Project Charter 3](#_Toc529739586)

[Statement 3](#_Toc529739587)

[Justification 3](#_Toc529739588)

[Scope 4](#_Toc529739589)

[Goal and Objectives 4](#_Toc529739590)

[Approach 5](#_Toc529739591)

[Initial Assumptions 5](#_Toc529739592)

[Constrains 6](#_Toc529739593)

[Risks 6](#_Toc529739594)

[Project Breakdown Structure 7](#_Toc529739595)

[Gantt Chart 12](#_Toc529739596)

[Main Tasks 13](#_Toc529739597)

[PERT 14](#_Toc529739598)

[Unity Engine Skills Development 14](#_Toc529739599)

[C# Programming Skills Development 15](#_Toc529739600)

[Develop Coding Skills 15](#_Toc529739601)

[Separate Individual Threads from the Main Thread 16](#_Toc529739602)

[Develop Self-Awareness for AI Objects 17](#_Toc529739603)

[Conclusions 17](#_Toc529739604)

[References 18](#_Toc529739605)

# PMI Project Charter

The best tool for a project development is a Project Charter. A project charter is a document created by the initiator of the project that confirms its existence and provides authority to the managerial coordinator to apply organisational resources over the activities of the project. It is created at the very start of a project when the aims and objectives begin to develop. It represents the connection between the development process and the managerial process of the project. Project managers with limited experience frequently misunderstand the *project charter* term considering a project charter as a formal document created by lawyers carrying a certain legal weight. A project charter is a legal document, simple, short and authorises a temporary endeavour.

## Statement

The main objective of the project is to temper with the foundations of multithreading and create stages to house thread-switching hardware in game developing software. The definition of a thread refers to the execution path of a software. When a software comprises complex and time-consuming operations, it is a good practice to create separate execution threads, each thread implementing a precise job. Threads are demarcated as lightweight processes. Used in operating systems, threads will save important CPU resources and increase the functionality and performance of an application. However, in game development multiple threads must run concurrently to suit the software needs taxing the CPU’s features and devaluing the abilities of the software. In C# (sharp) programming software when a program starts the *main thread,* or the main process, will be created. The main thread is a class created to work with threads, will tolerate the creation and access for individual threads. The individual threads will be created in the main thread and will be synchronised in the main thread once the process is completed.

Intense use of multithreading in game development will tax the rendering capabilities of the machine creating the need of expensive hardware upgrades, a process which will produce undervalue for the developed system. Software systems have been created and put in place to help and improve the rendering process and enhance the software performance. Such a system has been implemented in Unity game engine, named Unity Jobs System. Unity Jobs System allows the developer to write multithreading code that interacts extremely well with Unity, creating an optimal game frame. This is a very important step in creating a game frame with multiple objects, all the object performing a job at the same time. The main drawback is that objects need to be grouped and a group of objects must perform the same action in order to keep optimum performance, it is developed similar to a stage design created to improve the visual interface of the game.

## Justification

Multithreading has become a simple and powerful way to develop concurrent programs. The process of multithreading can be explained as the execution of multiple processes in a single address memory location, through multithreading the process of transaction processing takes place. Considering the difficulty of speeding up a programming task, most computers have been taught to multitask between the multiple threads of a program. If one thread will get multiple catch misses the other threads will take advantage of the CPU resources that have not been used, implying that if a thread cannot use the CPU features those resources will be used by another thread instead of becoming idle. An issue in multithreading is multithread interference, when sharing the resources of the CPU, and as a result the execution time can be degraded.

As explained in the statement, the multithreading process in a game software is created in the main thread. All the individual processes will be created in the main thread and upon completion everything will be synchronised with the main thread. A highly complex operating software will contain an increased number of individual processes, each designed for a specific purpose, that will use a substantial amount of the CPU’s resources such as computing units, CPU catches, and translation buffers. The high number of threads will create a processing queue, threads will be processed at a specific time in a designated order. Complications will arise with the low capabilities of the CPU that will be obligated to drop different threads to increase the performance, process that will have an impact on the rendering capabilities.

The proposed project is to design a software prototype capable of allowing the creation of individual threads outside the bonds of the main thread allowing only the synchronisation with the main thread at the process completion. The development process will make use of the complex architecture of the CPU enabling other delivery paths of the processes eliminating the complexity of the queue created by the main thread and improving significantly the rendering process in a game development software.

Assuming the functionality of the prototype, game objects could benefit of significant improvement features, generating a new type of AI game objects capable of enabling environment self-awareness, raising the standards of game development with high improvement of the software rendering process. Devices without high hardware resources available would be able to run complex developed software without significant improvements in hardware.

## Scope

Following a general approach, the main aims and objectives will be prioritised and ranked in the order of their importance. The development process will allow the possibility of revising the priorities in order to adjust the direction of the project.

The development environment for the project is a 3rd person shooter game developed in Unity game engine, *Patient Zero.* The game is a perfect environment to implement and test the main aims and objectives stated in the project plan and could bring significant improvements features to the game with the development of the project.

### Goal and Objectives

The aims of the project are:

1. Review Unity Jobs multi-threading System in relation to multi-threading objective.
2. Run the AI game objects on individual threads separate from the Main Thread.
3. Allocate game object dynamic run-time cover with the use of Ray-cast edge detection framework opposed of pre-set cover positions
4. Prototype coordination framework for AI game objects

The objectives to be completed throughout the development are:

1. Analise system development in the classic multi-threading system and the Unity Jobs system, understand the stages developed for this specific upgrade and develop a prototype coding method to separate the individual threads from the main thread. The developed prototype will be implemented in the game frame and tested to be evaluated for performance purposes.
2. Regardless of the native programming, a game object, AI, needs to have supplementary coding implemented to achieve self-learning.
3. To achieve self-learning, a game object, requests additional coding methods to accomplish a self-learning capability.

Unity provides basic game object features, limited to basic programming such as pathfinding, *NavMesh*. A project goal is to create game objects environmental self-awareness. Prototype methods could be built to *teach* an AI to use pathfinding or complete a task without user guidance in an environment.

### Approach

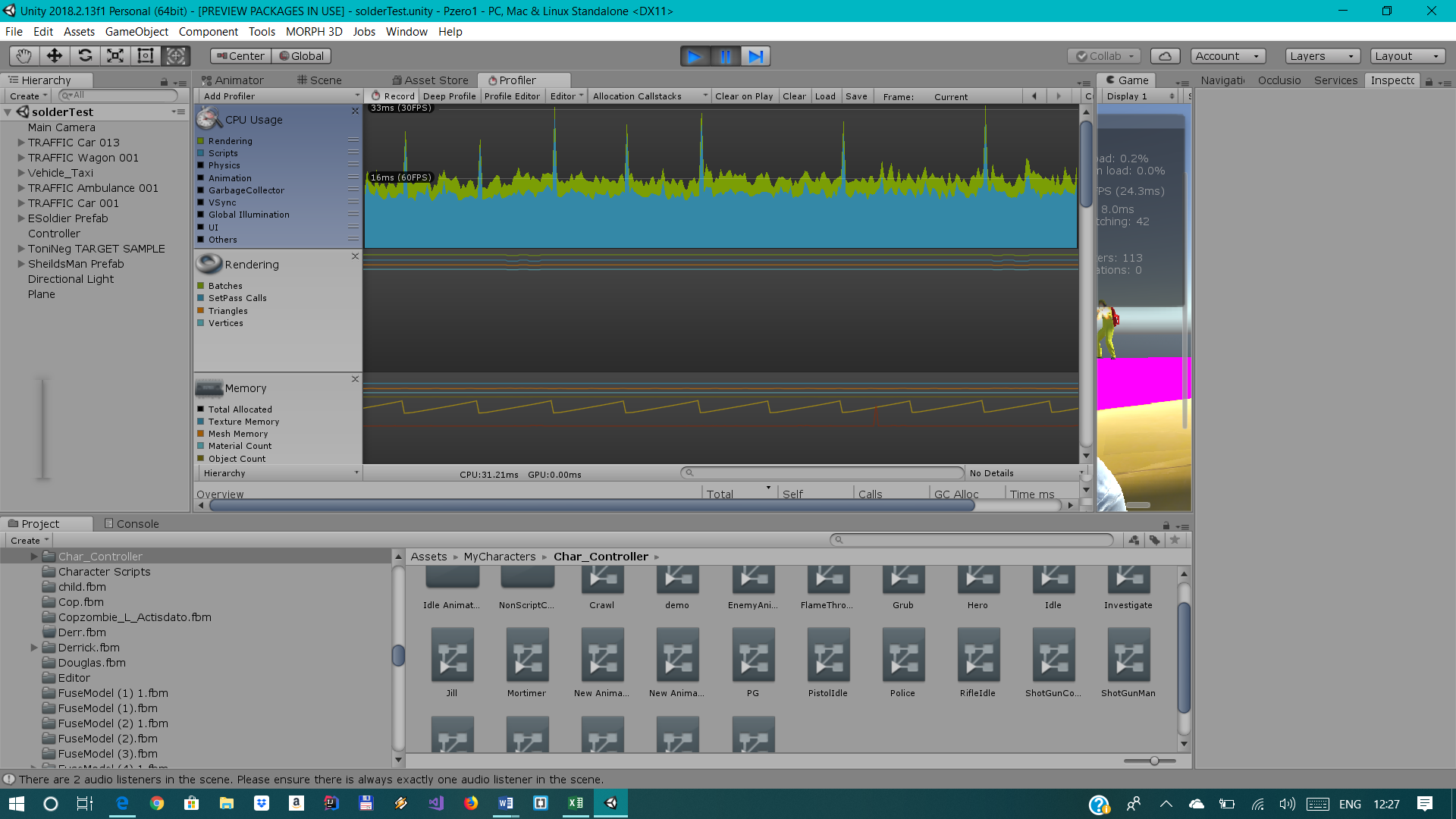
There are a set of activities to be completed to attain the completion of the objectives:

1. Develop programming skills in in *C#*  language, a Microsoft developed programming language.
2. Become familiar with Visual Studio, a Microsoft development environment.
3. Learn single-thread, multi-thread concept and their dependencies with the CPU.
4. Create a development environment with game objects extracted from *Patient Zero.*
5. Prototype different coding methods.
6. Research AI dynamic programming.
7. Research different programming languages and their library extensions to analyse resource allocation in terms of multi-threading.
8. Carry out testing through the development process.
9. Integrate the prototype methods in the game frame and to the end of the project in *Patient Zero*.
10. Create a coding report and instructions.

### Initial Assumptions

Before the start of the project a measuring tool is required. Unity game engine has introduced a software that allows the control and optimisation of a software. *Unity Profiler* enables optimisation by reporting the time spent in different stages of the application. Gives feedback on animation, rendering and game optimisation.

To get accurate readings and progress measurements, this software is compelled to make measurement on the same device at the beginning and end of the project. Switching the device throughout the development process will make all initial measurements redundant.



Unity Profiler is focused on the CPU capabilities in terms of the rendering and script process. Successful completion of the project, and implementation of prototype methods will allow visual improvements to be seen in the rendering process.

The proposed prototype programming methods will be available for implementation during the development process assuming the Unity game engine development software allows multi-lingual threading methods created individually apart of the main thread. Otherwise, a different approach will be implemented by using a *pointer*  programming method. A pointer is a programming language object with a set value which refers to another value stored in a different location memory using its memory address.

Before the development process several assumptions are made that can improve the feasibility of the project:

* Software components already exist which can save weeks of development
* Support from an active development community capable of providing updates and answers
* Existence of reusable code
* Discussions with professional developers

### Constrains

The outcome of the project will be highly influence by the time allowed for development. In the development process, several learning stages will have to be completed such as developing C# programming skills, familiarise with Visual Studio development platform, acquire abilities to create Unity software. A strict time schedule needs to be implemented to keep the project on the right development path.

### Risks

The project will begin will light processes and gradually grow towards the main aims and objectives. Lengthways with the time management risk several other risks rise with the implementation of prototype programming methods, managing resources and changing the environment. The main cause for project failures are the technical risks. A technical risk is defined as the possibility of a software to fail to produce the intended software product. In this scenario the technical risks may cause the project to be taxing for the CPU, have a late delivery or simply unacceptable and not for the intended purpose.

Technical risks are:

* Developing wrong prototypes
* Adding more functionality than necessary
* Real-time performance issues
* Ignoring resource capabilities
* Shortfall in performance
* Incorrect detailing

Effective risk management is an important tool to be employed for the success of a project.

Risk management is not extensively used and can prove to be a competitive advantage for developers which implement risk management processes in their projects.

# Project Breakdown Structure

A work breakdown structure is a deliverable breakdown of a project into smaller components. The work breakdown structure represents the most important project deliverables which organise the project development into controllable stages.

The important keys of the proposed project are comprised in the following structure:

The project has been divided into five stages, each stage implementing several task and sub-tasks to aid the development of the proposed project.

The main stages are :

* Develop Unity Game development skills
  + Evaluate the difference between multithreading and individual threading
  + Review Unity Jobs System
* Separate the Main Thread from Individual Threads
  + Develop C# programming skills
  + Familiarize with Visual Studio
  + Develop coding skills
* Alter Ray-cast Framework
  + Set up a development environment
  + Enhance the A\* Pathfinder method
  + Improve the Edge-detection method
  + Optimize the Field of View method
* Improve and develop AI self-awareness
  + Develop prototype methods for AI coordination framework
  + Research any existing AI dynamic programming
  + Create AI development prototype methods
  + Increase the game frame size to allow programming flexibility (add more objects to the scene and observe the rendering process)
* Documentation of progress and findings

|  |  |  |  |
| --- | --- | --- | --- |
| **WBS #:** | 1.1 | **Task:** | Develop Unity Game Engine Skills |
| **Est. Level of Effort:** | 14 days | **Owner:** | Project Manager |
| **Resources Needed:** | Unity Docs | **Work Products:** | Unity Game Engine |
| **Description of Task:** | Develop understanding in Unity Game development platform, build, test and debug an application in basic drag and drop implementation | | |
| **Input:** | * Project Supervisor/Software Development Tutors | | |
| **Dependencies:** | * Game Software Development | | |
| **Risk:** | * Time constrains as software development platform is highly complex | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **WBS #:** | 1.2 | **Task:** | Evaluate Multithreading and Individual Threading |
| **Est. Level of Effort:** | 3 days | **Owner:** | Project Manager |
| **Resources Needed:** | C# Microsoft Documentation | **Work Products:** | C# Development Platform |
| **Description of Task:** | Oversee the process flow control, application execution path, method execution and CPU resource sharing in multithreading and individual threading processes | | |
| **Input:** | * Project Supervisor/Software Development Tutors | | |
| **Dependencies:** | * C# Development Platform | | |
| **Risk:** | * Complex software, not possible to alter the multithread implementation | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **WBS #:** | 1.3 | **Task:** | Review Unity Jobs System |
| **Est. Level of Effort:** | 5 days | **Owner:** | Project Manager |
| **Resources Needed:** | Unity Jobs Documentation | **Work Products:** | Unity Game Engine |
| **Description of Task:** | Understand the classic multithreading process in game development and the new implemented Unity Jobs System | | |
| **Input:** | * Project Supervisor/Software Development Tutors | | |
| **Dependencies:** | * Unity Game Engine Platform | | |
| **Risk:** | * Failure to develop Understanding in the new implemented system | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **WBS #:** | 2.1 | **Task:** | Develop C# Programming Skills |
| **Est. Level of Effort:** | 14 days | **Owner:** | Project Manager |
| **Resources Needed:** | C# Development Courses | **Work Products:** | C# Development Platform |
| **Description of Task:** | Develop programming skills in C# programming development software, create software programming abilities focused on game development | | |
| **Input:** | * Project Supervisor/Software Development Tutors | | |
| **Dependencies:** | * C# Development Platform | | |
| **Risk:** | * Time constrain as methods involved in game development differ from basic C# programming methods. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **WBS #:** | 2.2 | **Task:** | Familiarize with Visual Studio Development Platform |
| **Est. Level of Effort:** | 4 days | **Owner:** | Project Manager |
| **Resources Needed:** | Visual Studio Development Courses | **Work Products:** | Visual Studio Software |
| **Description of Task:** | Understand the workflow, debug and resource allocation process in Visual Studio software. | | |
| **Input:** | * Project Supervisor/Software Development Tutors | | |
| **Dependencies:** | * Visual Studio Software | | |
| **Risk:** | * Failing to understand the debugging process in Visual Studio. Error detection is unclear and platform software point to several errors instead of the main error | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **WBS #:** | 2.3 | **Task:** | Develop Coding Skills |
| **Est. Level of Effort:** | 14 days | **Owner:** | Project Manager |
| **Resources Needed:** | Microsoft Documentation for C# programming in Visual Studio | **Work Products:** | C# Programming  Visual Studio |
| **Description of Task:** | Develop software programming skills in Visual Studio platform using C# programming language specific for game development | | |
| **Input:** | * Project Supervisor/Software Development Tutors | | |
| **Dependencies:** | * C# Development in Visual Studio | | |
| **Risk:** | * Failure in understanding game development programming in Visual Studio | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **WBS #:** | 2.4 | **Task:** | Separate Main Thread from Individual Threads |
| **Est. Level of Effort:** | 15 days | **Owner:** | Project Manager |
| **Resources Needed:** | Microsoft C# Programming Documentation | **Work Products:** | C# Programming Platform |
| **Description of Task:** | Create prototype methods for individual threads outside the bounds of the main thread, implement new CPU resource allocation scheme allowing multiple individual threads to process concurrently without taxing the CPU’s capabilities | | |
| **Input:** | * Project Supervisor/Software Development Tutors | | |
| **Dependencies:** | * C# Development Platform | | |
| **Risk:** | * Development Platform will allow the implementation of individual threads but will fail to update the completion of individual threads with the main thread. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **WBS #:** | 3.1 | **Task:** | Set Up a Development Environment |
| **Est. Level of Effort:** | 2 days | **Owner:** | Project Manager |
| **Resources Needed:** | Unity Documentation | **Work Products:** | Unity Engine |
| **Description of Task:** | Create a simple game frame implementing game objects and characters imported from *Patient Zero,* to temper with the software coding without affecting the main game | | |
| **Input:** | * Project Supervisor/Software Development Tutors | | |
| **Dependencies:** | * Unity Game Engine Software | | |
| **Risk:** | * Incorrect implementation of game objects | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **WBS #:** | 3.2 | **Task:** | Alter Ray-cast Framework |
| **Est. Level of Effort:** | 8 days | **Owner:** | Project Manager |
| **Resources Needed:** | Unity Documentation  C# Microsoft Documentation | **Work Products:** | Unity Engine |
| **Description of Task:** | Temper Ray-cast programming to allow improvements in AI edge-detection and obstacle detection. Implement prototype methods to improve the behavior of game-objects | | |
| **Input:** | * Project Supervisor/Software Development Tutors | | |
| **Dependencies:** | * Unity Game Engine Software * C# Development Platform | | |
| **Risk:** | * Game object programming is saturated and will not allow further method implementation | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **WBS #:** | 3.3.1 | **Task:** | Enhance A\* Pathfinder Method |
| **Est. Level of Effort:** | 8 days | **Owner:** | Project Manager |
| **Resources Needed:** | Unity Documentation  C# Microsoft Documentation | **Work Products:** | Unity Engine |
| **Description of Task:** | Extension of Ray-cast method. The basic A\* Pathfinder method will be evaluated and upgraded to increase flexibility of game object movement in frames develop a dynamic behavior of game objects | | |
| **Input:** | * Project Supervisor/Software Development Tutors | | |
| **Dependencies:** | * Unity Game Engine Software * C# Development Platform | | |
| **Risk:** | * Failure to upgrade the basic A\* Pathfinder method | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **WBS #:** | 3.3.2 | **Task:** | Improve Edge-Detection Method |
| **Est. Level of Effort:** | 8 days | **Owner:** | Project Manager |
| **Resources Needed:** | Unity Documentation  C# Microsoft Documentation | **Work Products:** | Unity Engine |
| **Description of Task:** | Extension of the A\* Pathfinder method. With the development of the A\* pathfinder method, prototype methods will be implemented to create enhanced game object environmental awareness. | | |
| **Input:** | * Project Supervisor/Software Development Tutors | | |
| **Dependencies:** | * Unity Game Engine Software * C# Development Platform | | |
| **Risk:** | * Prototype methods implemented have a negative impact for static game object implementation and have an impact on the AI behavior. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **WBS #:** | 3.3.3 | **Task:** | Optimize Field of View Method |
| **Est. Level of Effort:** | 12 days | **Owner:** | Project Manager |
| **Resources Needed:** | Unity Documentation  C# Microsoft Documentation | **Work Products:** | Unity Engine |
| **Description of Task:** | Extension of the Ray-cast method and A\* Pathfinder method. Prototype dynamic methods to optimize and enhance the AI game object environment self-awareness for an improved game experience. | | |
| **Input:** | * Project Supervisor/Software Development Tutors | | |
| **Dependencies:** | * Unity Game Engine Software * C# Development Platform | | |
| **Risk:** | * Prototype methods implemented can have a negative impact in the AI environmental self-awareness due to complex detection of game frame. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **WBS #:** | 4.1 | **Task:** | Prototype AI Coordination Framework |
| **Est. Level of Effort:** | 7 days | **Owner:** | Project Manager |
| **Resources Needed:** | Unity Documentation  C# Microsoft Documentation | **Work Products:** | Unity Engine |
| **Description of Task:** | Extension of the Ray-cast method, A\* Pathfinder method and Field of View Method. Prototype methods developed, based on the completion of previous stages will upgrade the coordination ability of game objects improving the time to complete task and decrease percentage of resources used in the process. | | |
| **Input:** | * Project Supervisor/Software Development Tutors | | |
| **Dependencies:** | * Unity Game Engine Software * C# Development Platform | | |
| **Risk:** | * Methods implemented can harm the proper functionality of game objects and of the software by incorrect allocation of resources and have a negative effect for the rendering capabilities of the software. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **WBS #:** | 4.2.1 | **Task:** | Develop Self-Awareness for AI Objects |
| **Est. Level of Effort:** | 14 days | **Owner:** | Project Manager |
| **Resources Needed:** | Unity Documentation  C# Microsoft Documentation | **Work Products:** | Unity Engine |
| **Description of Task:** | Extension of the Ray-cast method, A\* Pathfinder method and Field of View Method. Dynamic methods will be developed to enhance the awareness and coordination of the AI game-object in the game frame following the completion of the previous stages in the project development. Completing the stage leads to the development of improved game objects and quality of the designed software. | | |
| **Input:** | * Project Supervisor/Software Development Tutors | | |
| **Dependencies:** | * Unity Game Engine Software * C# Development Platform | | |
| **Risk:** | * Prototype methods implemented can have a negative impact in the AI environmental self-awareness, coordination framework due to complex detection of game frame, resources and multithreading. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **WBS #:** | 4.3 | **Task:** | AI Prototype Programming |
| **Est. Level of Effort:** | 10 days | **Owner:** | Project Manager |
| **Resources Needed:** | Unity Documentation  C# Microsoft Documentation | **Work Products:** | Unity Engine |
| **Description of Task:** | Extension of the Ray-cast method, A\* Pathfinder method and Field of View Method. Dynamic methods will be developed to enhance the awareness and coordination of the AI game-object in the game frame following the completion of the previous stages in the project development. Completing the stage leads to the development of improved game objects and quality of the designed software. | | |
| **Input:** | * Project Supervisor/Software Development Tutors | | |
| **Dependencies:** | * Unity Game Engine Software * C# Development Platform | | |
| **Risk:** | * Prototype methods implemented can have a negative impact in the AI environmental self-awareness, coordination framework due to complex detection of game frame, resources and multithreading. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **WBS #:** | 4.4 | **Task:** | Increase Frame Size for Programming Flexibility |
| **Est. Level of Effort:** | 2 days | **Owner:** | Project Manager |
| **Resources Needed:** | Unity Documentation  C# Microsoft Documentation | **Work Products:** | Unity Engine |
| **Description of Task:** | Size of the game frame used in project development to this stage will be increased in size adding new components and new game-objects to test the functionality of prototype methods developed and test the functionality on a higher scale. | | |
| **Input:** | * Project Supervisor/Software Development Tutors | | |
| **Dependencies:** | * Unity Game Engine Software * C# Development Platform | | |
| **Risk:** | * Rendering issues are expected to occur. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **WBS #:** | 4.2.2. | **Task:** | Research for Existing AI Dynamic Programming |
| **Est. Level of Effort:** | 2 days | **Owner:** | Project Manager |
| **Resources Needed:** | AI Dynamic Object Programming Documentation | **Work Products:** | AI Object Development Platforms |
| **Description of Task:** | Research will be conducted to compare different AI game objects dynamic programming. The findings will be evaluated, and methods will be developed to improve and implement the findings in the proposed project. | | |
| **Input:** | * Project Supervisor/Software Development Tutors | | |
| **Dependencies:** | * Web Research | | |
| **Risk:** | * Findings are not relevant to the proposed project | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **WBS #:** | 5.1 | **Task:** | Document Findings |
| **Est. Level of Effort:** | 2 days | **Owner:** | Project Manager |
| **Resources Needed:** | Unity Documentation  C# Microsoft Documentation  Web Research | **Work Products:** | Microsoft Word  Microsoft Researcher |
| **Description of Task:** | All the stages of the proposed project will be documented with the outcomes and improvements added to the proposed project. | | |
| **Input:** | * Project Supervisor/Software Development Tutors | | |
| **Dependencies:** | * Unity Game Engine Software * C# Development Platform | | |
| **Risk:** | * Stages of the proposed project will not be completed in the time constrain. | | |

# Gantt Chart

A Gantt chart reveals the schedule of a project. It shows the dependencies between the activities and the current time schedule, compares the task that are in progress with the tasks that have been completed. A Gantt chart visualises the passing of time and stands to reduce the idle time.

The Gantt chart constructed for the proposed project has been built in Microsoft Excel, due to limited features, the milestones of the project are poorly presented in the chart. The development process contains four milestones, each milestone depicts a culmination of a series of processes as presented in the work breakdown structure.

The start date of the proposed project is set on the 15th December 2018 as shown in the Gantt chart and due to be completed before May 24th. The stages of the development process will succeed one another throughout the development process according to the project breakdown structure. Each of the proposed stages is allocated a specific development time frame according to the complexity of the stage. If a development process is completed before the scheduled time, the idle time shall be used as backup time in the development of different stages of the project if any of the processes require extra time.

The risk of process development failure will be present throughout the whole development process. If one or several stages of the development process cannot be completed in the designated time, new prototype programming methods will be explored, implemented and documented. In case of stage development process failure all the results will be documented.

# Main Tasks

The complete development process comprises several processes. Every process has an important role in project completion and will sustain the development of the processes to follow. The development process is bound to follow all the stages in the Gantt chart. According to the work breakdown structure and the Gantt chart, in order to complete the final stages of the project all the processes need to be followed and aimed for completion to obtain the intended results at the end of the development process.

The stages considered as turning points for the development process are:

* Separate individual threads from the Main Thread
* Enhance A\* Pathfinder method
* Improve Edge Detection method
* Optimize Field of View method

With the successful completion of these stages an important part of project deliverables will be obtained and will lead to the development of new prototype methods which will improve the rendering process, increase software compilation speed and improve resource allocation process.

Several stages presented in the Gantt chart are highly time-consuming and will request an increased attention in accomplishment creating an information base for software development and software development platforms.

The largest processes shown in the Gantt chart are:

* Develop Unity Game Engine Skills
* Develop C# Programming Skills
* Develop Coding Skills
* Separate Individual Threads from the Main Thread
* Develop Self-Awareness for AI Objects

### PERT

Program Evaluation and Review Technique (PERT) will be implemented in the project management. PERT is tool to estimate the duration of a project with an increased degree of uncertainty concerning the time-frame allocated for an individual process completion. Probabilistic time estimates will be implemented. The time-frame allocated for each individual process is calculated using optimistic, pessimistic and most-likely time estimates set for the duration of an individual process. PERT is also referred to as *a three-point estimate.*

The following formula will be used in in calculating time-frames for individual processes:

*PERT = (Optimistic time + (4 X Most likely time) + Pessimistic time) / 6*

The time is set to be measured in *days to completion*.

### Unity Engine Skills Development

A game engine is a framework for game development. The framework provides game developers with access to several core areas such as graphics, audio, logic, physics, AI, and network connection for a fast and easy development of a game software. In the early days of game development all the functions had to be hard coded throughout the whole development process. Using frameworks provides tools and structures that every game requires to function. Using game engines, a developer does not have to re-invent the wheel every time a game is developed. Game engines have increased in popularity with the development of 3D games and as technology increased in complexity. Many game development studios begun to use game engines to decrease the development time of a software.

Unity game engine supports high programming languages such as C# and JavaScript and contains many tools that allow developers to bypass extensive coding. Unity has released numerous free tutorials, assets and software contents making the engine a prime choice for junior developers.

A large time frame has been allocated to develop Unity game development skills for the best course direction of the proposed project. Online documentation will provide the necessary documentation to grow game development abilities. Video tutorials present the actual game implementation sequence. As a junior game developer in Unity, it is required to start the course from the beginner stage to developer skills. The developer course iterates through Unity user interface, scene, assets handling, builds and publishing. Advanced development features, scriptable objects, graphics and physics topics will be covered throughout the process.

The time allocated for this stage is 14 days. An average of 6 hours per day will be used for the development of the project. Online basic Unity game development courses have a duration of 60 hours, approximately 8 days.

Using PERT formula:

* optimistic time = 10
* most likely time = 12
* pessimistic time = 16

According to the PERT formula, the process will require approximately 12 days to complete, the idle time will be assigned to deepen the knowledge acquired in the process.

### C# Programming Skills Development

C# (sharp) is a general-purpose OOP programming language developed by Microsoft. The object of the programming language is to set a series of activities to be performed by a computer to accomplish a specific task. C# is intended for application development to be used in embedded systems such as operating systems or small applications having dedicated functions. C# programming tasks include extracting data from a database, display game graphics, control external devices attached to the device and playing sound effects. Everything that a computer can do can be programmed in C#. Unity is a native C++ based game engine. The codes are written in JavaScript, Boo and most frequently in C#.

Similar to Unity game skills development stage, a large time-frame has been allocated to this stage. To develop a game in Unity it is important to first develop C# programming skills. A programming C# course for beginners will progress through a duration of approximately 50 hours. The topic presented in the course will cover the foundations of C# programming, method development for code reusability, flow control statements, live coding sessions, data manipulation working with functions and directives.

The development time allocated for this stage is 14 days. An average of 6 hours per day will be used for project development. Due to task complexity the optimistic time will be increased with an additional two days to allow further time to deepen the knowledge gained in the process.

Using PERT formula:

* optimistic time = 12
* most likely time = 14
* pessimistic time = 18

According to the PERT formula de stage will require approximately 14 days to complete.

### Develop Coding Skills

In the coding skills development stage all the knowledge gathered from Unity development, and C# programming courses will be merged to develop small Unity game frames, C# programming scrips and methods. It is a large time scale development stage that allows the development of prototype methods and small-scale game scenes with game objects and game characters taken from the *Patient Zero*. Prototype programming method development will begin at this stage, the project being now matured to allow this step. The game characters imported from Patient Zero are programmed with the basic A\* Pathfinder, Field of View and Ray-cast edge detection methods. Prototype programming methods will be developed to temper and improve the original methods to add additional game object functionality. Step by step each method will be revised, upgrade attempts will be carried out. The stage development develops a knowledge base for the self-awareness of AI object development stage. The existing game software will not be tempered with in this stage of proposed project development, all the coding methods are carried out outside the main game frame.

The time frame allocated for this stage is 14 days. During this time period major game development coding skills will be acquired that will help progress in the project development.

Using PERT formula:

* optimistic time = 10
* most likely time = 12
* pessimistic time = 16

According to the formula this stage will be completed in approximately 12 day with allowing idle time in the proposed project development.

### Separate Individual Threads from the Main Thread

The concepts of Main Thread and Individual Thread have been explained in large lines in the *Justification* part of the Project Charter. Multithreading is a blueprint of an execution model which allows multiple threads to exist in a process and to execute separately with sharing of own resources.

Threads contain information that is relevant to its application including exception handlers, registers, memory space address and the priority schedule. Threading is also known as multithreading.

Multithreading is useful in a single processor operating system where the main execution thread is allowed to respond to the user input while the working thread can execute long processes in the background without the intervention of the user. In a multiprocessor operating system thread are concurrently executed, broadcasted across multiple processors. This case enables increased process computation speed. Concurrent thread execution requires enhanced attention to programming due to deadlocks, racing conditions and non-intuitive behaviour.

Every operating system can use multithreading in two ways:

* pre-emptive multithreading, where the threading is controlled by the operating system
* cooperative multithreading, where the thread is responsible for context switching

An individual thread is defined as a single process which is the execution of an application. An individual thread can perform one single action at one time. To program a thread to execute more processes, it can be separated in lesser processes.

Every thread in a software designed in C# starts from the *Main Thread*. The aim of this stage is to separate the construction of individual threads from the main thread. Every individual thread constructed outside the main thread is obligated to synchronise with the main thread as soon as the process has been completed. Separating individual threads from the main threads can add major improvements in software functionality and software rendering limiting the tasks queue developed by the main thread to access features of the CPU.

The success rate of this stage is low but backup prototype methods including the use of *pointer programming*  will be enabled for development.

The time-frame allotted to this stage is the highest one in the proposed process development, 15 days to complete the stage.

Using PERT formula:

* optimistic time = 12
* most likely time = 13
* pessimistic time = 20

According to the formula the completion of the stage will require approximately 14 days with one day idle time to be used in intensive testing in case of successful implementation of prototype methods.

### Develop Self-Awareness for AI Objects

One of the final stages of the proposed process development. In this stage all the acquired skills and developed prototype programming methods in the proposed project development will be used in creating AI game objects with enhanced game features and environmental self-awareness capability. This stage will combine all the methods iterated and improved in the development process such as A\* Pathfinder, Ray-cast edge detection, Field of View, Dynamic Object Detection method and Coordination Framework. All the methods will be implemented to an AI game object to create enhanced environmental self-awareness. The threads of the AI will be constructed outside the bounds of the main frame increasing the speed of the computations and have a positive impact over the rendering capabilities of the software. Intensive testing and development will be carried out in this stage, the positive results will lead to game frame size increase.

The time-frame allocated for this stage is 14 days. The optimistic time frame will be increased by two days due the complexity of the implemented methods.

Using PERT formula:

* optimistic time = 12
* most likely time = 14
* pessimistic time = 18

According to the Pert formula this stage will require 14 days to complete.

# Conclusions

A Project Charter and a Gantt Chart are powerful tools for project planning. Their aim is to handle an issue or an opportunity for the good development of the project. A Project Charter behaves as a guideline for projects and it is a very important tool in the management system of an organisation. The Project Charter has helped the proposed project in:

* documenting reasons for undertaking the project,
* presenting the constraints and objectives in the project,
* provide solutions,
* enable an activity workflow.

The Gantt chart enables project managers to track progress throughout the entire development process. It can be used to track the resource usage in the project. It can be used in any project which involves deliverables, milestones, resources and effort. It can be used to show dependencies in a project and as an individual entity for managing small projects.

# References

Anon., 2008. *PMI | Project Management Institute.* ed. (): Project Management Institute.

Anon., 2015. *Before starting a new software project - a few things to consider.* [Online]   
Available at: https://www.software.ac.uk/resources/guides/starting-new-software-project-few-things-consider  
[Accessed 7 November 2018].

Anon., 2015. *Learning about C# for beginners.* [Online]   
Available at: https://www.thoughtco.com/learn-about-c-958280  
[Accessed 10 November 2018].

Anon., 2016. *C Pointers.* [Online]   
Available at: https://www.tutorialspoint.com/cprogramming/c\_pointers.htm  
[Accessed 7 11 2018].

Anon., 2016. *C# - Multithreading.* [Online]   
Available at: https://www.tutorialspoint.com/csharp/csharp\_multithreading.htm  
[Accessed 05 11 2018].

Anon., 2017. *Multithreading.* [Online]   
Available at: https://www.techopedia.com/definition/24297/multithreading-computer-architecture  
[Accessed 11 November 2018].

Kind, A. & Padget, J., 1998. *Multi-lingual threading.* [Online]   
Available at: http://ieeexplore.ieee.org/document/647230  
[Accessed 7 11 2018].

UKEssays, 2013. *Risk management in project development.* [Online]   
Available at: https://www.ukessays.com/essays/it-research/risk-management-in-project-development.php  
[Accessed 09 November 2018].

Anon., 2016. *Tutorials Point.* [Online]   
Available at: https://www.tutorialspoint.com/sdlc/sdlc\_iterative\_model.htm  
[Accessed 1 November 2018].

Anon., 2016. *Unity.* [Online]   
Available at: https://unity3d.com/  
[Accessed 28 10 2018].

Anon., 2018. *C# Job System Exposed.* [Online]   
Available at: https://docs.unity3d.com/Manual/JobSystemMultithreading.html  
[Accessed 29 10 2018].

Anon., . *C# Language Specification.* ed. (): Ecma International.

Anon., . *What is Project Management?.* [Online]   
Available at: http://www.pmi.org/About-Us/About-Us-What-is-Project-Management.aspx  
[Accessed 31 10 2018].

Anon., . *What's new in C# 7.2.* [Online]   
Available at: https://docs.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-7-2  
[Accessed 1 11 2018].

Bennett, C. & Sagmiller, D. V., 2014. *Unity AI Programming Essentials.* [Online]   
Available at: https://amazon.com/unity-programming-essentials-curtis-bennett/dp/1783553553  
[Accessed 29 10 2018].

Birrell, A., 2005. *An Introduction to Programming with C# Threads.* [Online]   
Available at: http://birrell.org/andrew/papers/035-threads.pdf  
[Accessed 29 10 2018].

Gajjar, A., 2013. *Learning C# by Developing Games with Unity 3D Beginner's Guide.* s.l.: Packt Publishing Ltd.

Kwak, Y. H. & Stoddard, J., 2004. Project risk management: lessons learned from software development environment. *Technovation,* , 24(11), pp. 915-920.

Phillips, K., 2016. *Developing a 3D game engine using multi-threading.* [Online]   
Available at: http://ssudl.solent.ac.uk/3400  
[Accessed 29 10 2018].

Rhalibi, A. E., Merabti, M. & Shen, Y., 2006. *Improving game processing in multithreading and multiprocessor architecture.* [Online]   
Available at: http://dblp.uni-trier.de/db/conf/edutainment/edutainment2006.html  
[Accessed 29 10 2018].

Snell, M. & Powers, L., 2010. *Microsoft Visual Studio 2010 Unleashed.* [Online]   
Available at: https://amazon.com/microsoft-visual-studio-2010-unleashed/dp/0672330814  
[Accessed 1 11 2018].

Subashni, S. & Kumar, N. S., 2010. *Software Testing Using Visual Studio 2010.* [Online]   
Available at: https://amazon.com/software-testing-using-visual-studio/dp/1849681406  
[Accessed 30 10 2018].

Yoo, J.-K. & Kim, Y., 2015. *Packet Discard Policy of Network Thread in an Unity Engine for Multi-player Online Games.* [Online]   
Available at: http://ocean.kisti.re.kr/downfile/volume/kgames/kgohcl/2015/v15n6/kgohcl\_2015\_v15n6\_97.pdf  
[Accessed 29 10 2018].

Zander, J., . *Couple of Historical Facts.* [Online]   
Available at: http://blogs.msdn.com/jasonz/archive/2007/11/23/couple-of-historical-facts.aspx  
[Accessed 30 10 2018].

### 

## 

## 