***Analysis and Design Specification***

***Student Name:*** *Jamie Mulvaney*

***Student Number:*** *x12358831*

***Email address:*** [*x12358831@student.ncirl.ie*](mailto:x12358831@student.ncirl.ie) *or* [*jamiex304@hotmail.com*](mailto:jamiex304@hotmail.com)

***Degree Programme Name:*** *BSc (Hons) in Computing*

***Specialisation:*** *Gaming and Multimedia*

***Date :*** *04/12/2015*

*Contents*

[*Introduction* 2](#_Toc436593317)

[*Purpose of the Product Design Specification Document* 2](#_Toc436593318)

[*General Overview and Design Guidelines/Approach* 2](#_Toc436593319)

[*Assumptions / Constraints / Standards* 2](#_Toc436593320)

[*Architecture Design* 3](#_Toc436593321)

[*Hardware Architecture* 3](#_Toc436593322)

[*Special Resources for VR Setup/ Interaction* 3](#_Toc436593323)

[*Software Architecture* 4](#_Toc436593324)

[*Main Character Model and Animation Design Preview* 5](#_Toc436593325)

[*Game Architecture* 6](#_Toc436593326)

[*System Design* 7](#_Toc436593327)

[*Use-Cases* 7](#_Toc436593328)

[*Main Use Case Diagram* 7](#_Toc436593329)

[*Use Case 1: Play Game/Movement* 8](#_Toc436593330)

[*Use Case 2: Save Game* 10](#_Toc436593331)

[*Use Case 3: Load Game* 12](#_Toc436593332)

[*Use Case 4: Exit Game* 14](#_Toc436593333)

[*Use Case 5: Start a new Game/Modify Settings* 16](#_Toc436593334)

[*Use Case 6: Load/Complete Tutorial Level* 18](#_Toc436593335)

[*Use Case 7: Interact with in-game items/systems* 20](#_Toc436593336)

[*Use Case 8: VR Setup/Interaction* 22](#_Toc436593337)

[*User Interface Design* 24](#_Toc436593338)

[*Performance* 26](#_Toc436593339)

[*Section 508 Compliance* 26](#_Toc436593340)

# *Introduction*

## *Purpose of the Product Design Specification Document*

The purpose of this document is to set out the design and analysis specifications for the development of a sci-fi horror game called Twin Darkness that uses a VR interface (Oculus Rift) for my final year project for the gaming and multimedia stream of the BSHC programme.

The intended customers are horror game players, those infested in games that are built for the VR market and those that enjoy a well scripted story with choices that matter.

The scope of twin darkness is to develop a full VR driven horror game that takes place in a sci-fi like environment with enemy’s and puzzles that the player must beat in order to escape.

The game shall have the following:

* ***5 levels, that are designed and built using the unreal engine 4***
* ***Interactive props that are designed and built using blender***
* ***A fully build character model with rigging that is also built using blender***
* ***Custom Animations for the main character***
* ***A detailed and explained story so the player can have the full impact of the game***
* ***Full VR interface using the Oculus as the base VR headset***

The game will have a hide and seek layout (as in there will be little to no combat the player must just escape and hide from enemy’s) the reason for this is to up the fear level. Enemy’s will get harder per level. There will be a number of choices (such as what paths to take, which hopefully will make the game more fun to play through more than once) finally the player must work towards escaping herself or helping the other person that is trapped escape as well (this is where one must choose themselves or others hence embracing your Darkness)

# *General Overview and Design Guidelines/Approach*

## 

## *Assumptions / Constraints / Standards*

* **AI** – Artificial Intelligence
* **VR** – Virtual Reality, Oculus Rift
* **Unreal** – Unreal Engine 4
* **Blender** – Blender 3d Modelling Tool
* **NPC** – Refers to either and enemy or a friendly A.I driven character
* **Puzzles** – Puzzles mean simple tasks the player must perform in order to process further or get special prizes.
* **Interactive Props** – These range from tables and draws that pull out to being able to lock and unlock doors to throw of enemy’s

# *Architecture Design*

## *Hardware Architecture*

### *Special Resources for VR Setup/ Interaction*

For the user to enable the VR input they need the following hardware (or similar) a VR system is not essential to play the Game:

* **Xbox Controller (Standard Wireless and Plugged)**

The Controller will be used to control and play the game since the user will be using the oculus rift and not a keyboard and mouse.

If the user has opted out of using the oculus rift, the user may then decide to use either the controller or keyboard and mouse set up as both will be enabled in my project



* **The Oculus Rift headset (Development or Customer Version)**

The oculus rift headset will be used to play the game at its full potential. Along with the Hardware the user must have the latest Oculus rift runtime installed on their system and the system must meet the requirements outlined by the Oculus rift website.



## *Software Architecture*

***List of software being used:***

* ***Unreal engine 4***
* ***Blender***
* ***Photoshop***
* ***Adobe Fireworks***

How the software will be used:

The bulk of the level’s and technical functionality will be build using the Unreal engine 4 game system. I will use blender to create custom assets to be imported into the unreal engine 4.

What I will use unreal engine 4 for is the following:

* ***Level design***
* ***Basic mechanics and layout of a game (such as a start menu, save options and cut scenes)***
* ***Custom visuals using unreal engine 4’s particle system***
* ***Animation of assets created in blender using both custom and default animation setups that unreal offer***

What I will use Blender for is the following

* ***Main Character modelling (This will be the character the player will see in all cut scenes and the character they will be controlling) I will use blender to build and model the character and then create a custom skeleton rig using blenders own armature system. This model will then be imported into unreal for animation setup before being place in the game.***
* ***Main enemy modelling, this is the same as the process above for the Main character modelling the designs will be made in blender along with the Skelton rig before they are imported into unreal.***

Finally when it comes to building and making the custom textures and images for the game I will use a mix of Photoshop and Adobe Fireworks

## *Main Character Model and Animation Design Preview*

***Gemini (This still is taken from Unreal engine 4 after import from blender)***

******

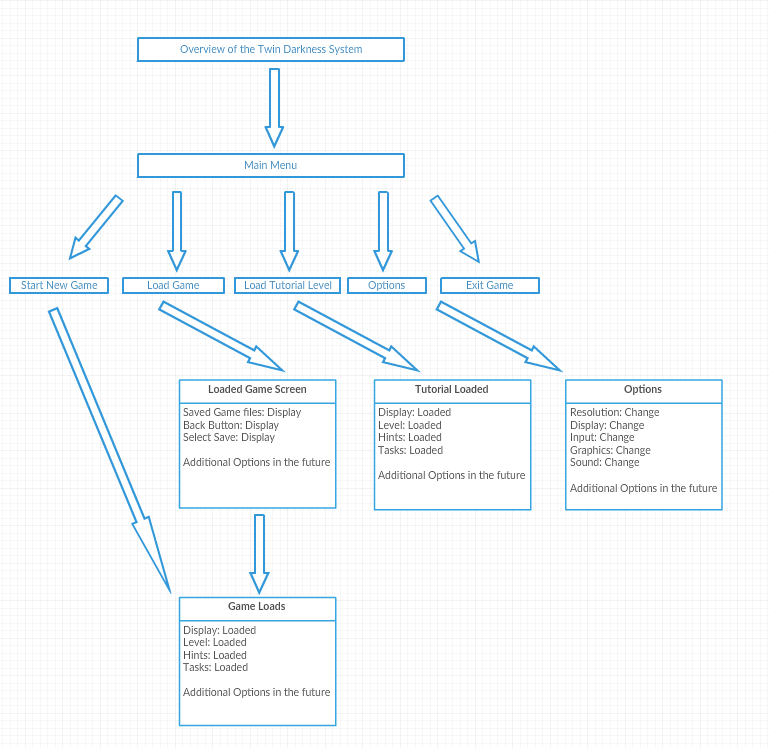
***This video shows all the current animations in unreal designed for my main character***



## *Game Architecture*

The system architecture for twin darkness follows the main stream structure when it comes to system design, with a main menu with all the actions available to the user (start new game, load game, load tutorial level, options and exit game)

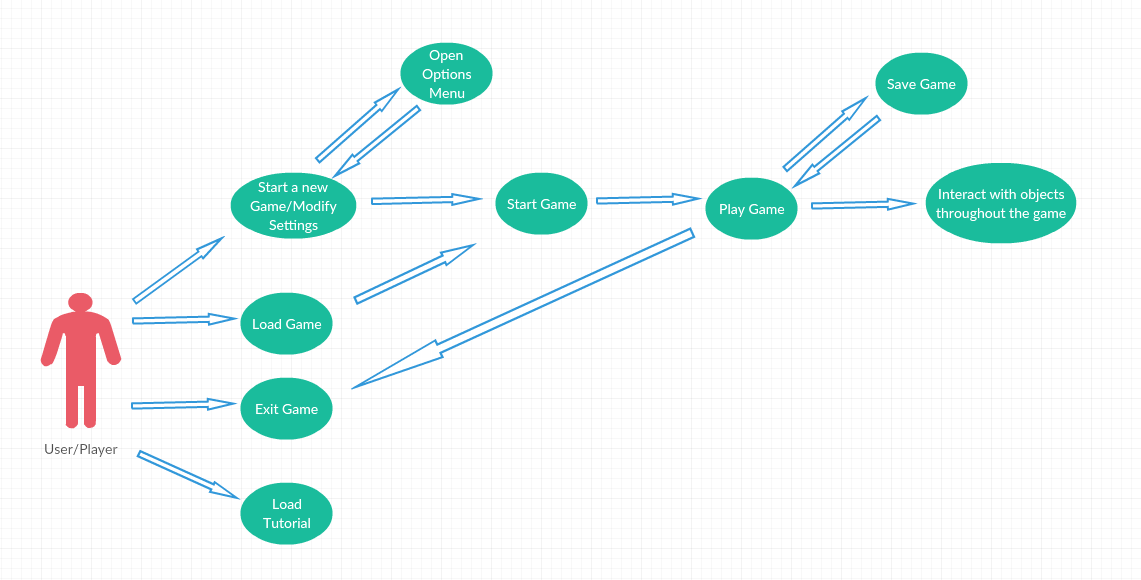
***Class Diagram of the Twin Darkness system architecture***



# *System Design*

## *Use-Cases*

### *Main Use Case Diagram*



***Each of the above use cases are broken down below into their induvial blueprints showing what is need for each to perform there functions***

### *Use Case 1: Play Game/Movement*

#### Description & Priority

The user must be able to play the game (by play I am mean achieve movement using input controls) thus being able to explore the levels of the game, hence the priority on this requirement is large.

#### Use Case

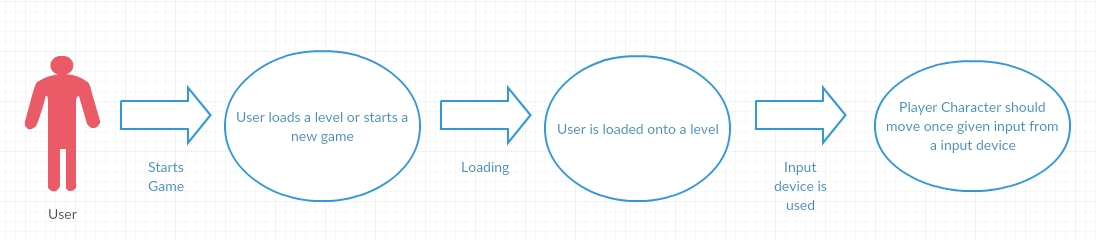
**Scope**

The scope of this use case is to allow the user to play the game hence moving around the open levels of the game

**Description**

This use case describes the functions of movement that the player must be able to do in order to play and enjoy the experience.

**Use Case Diagram**

****

***Flow Description***

**Precondition**

The Game is installed and the Twin Darkness game icon is on the desktop of the machine. Your desktop or laptop should meet the minimum requirements. A game level should be loaded

**Activation**

This use case starts when a User/Player uses their input device (keyboard, mouse or controllers) to make the character move in a given direction

**Main flow**

1. The game loads a level for the user
2. The user sends input using the mouse and keyboard
3. The game reads the input and moves the character
4. Steps 2 and 3 are done at nearly the same time so the user does not have to worry about lag time

**Alternate flow**

A1: Input from a Controller

1. The game loads a level for the user
2. The user send input using the controller
3. The Use Case continues at position 3 of the main flow

**Exceptional flow**

E1: Optimal Way of Loading the game

1. The game loads a level for the user
2. The user sends input using the mouse and keyboard
3. The game reads the input and moves the character
4. Steps 2 and 3 are done at nearly the same time so the user does not have to worry about lag time

**Termination:** Process only stops once the play has stop giving input through a device otherwise it does not terminate

**Post condition**: (There is no post condition as movement and input from a device will always be needed in some way when playing the game)

### *Use Case 2: Save Game*

#### Description & Priority

The user must be able to start the game and load into the level from there they must be able to save the game at any point

#### Use Case

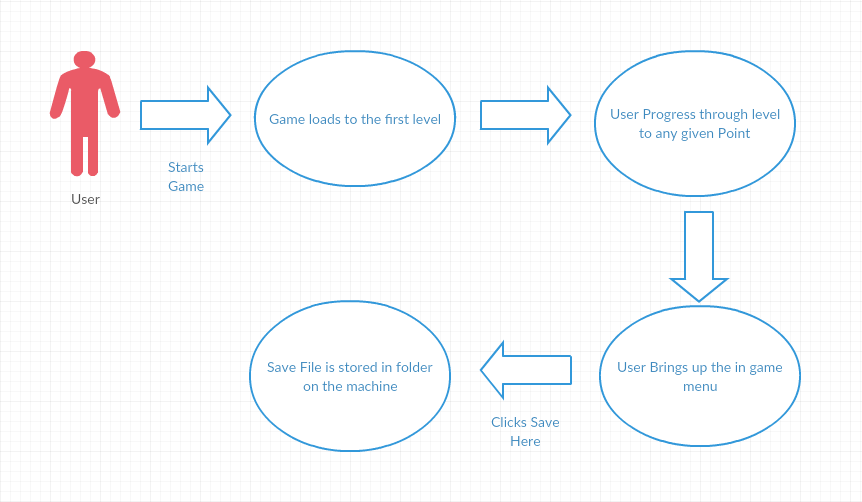
**Scope**

The scope of this use case is to allow the user to save the game, hence the location and current status on a given level

**Description**

This use case describes the function of saving an instance of the game, the save will store the users current status (Location etc.) and the Level Status.

**Use Case Diagram**

****

***Flow Description***

**Precondition**

The Game is installed and the Twin Darkness game icon is on the desktop of the machine. Your desktop or laptop should meet the minimum requirements. The user should have also have start a new game or loaded a previous version.

**Activation**

This use case starts when a User/Player brings up the in game menu after starting the game and loading into a level.

**Main flow**

1. The game has loaded the level
2. The user can move there character etc. and progress through the level
3. The user brings up the in-game menu
4. The game brings up the in-game menu
5. The user selects Save Here
6. The game creates a save file
7. The game stores the file on the machine

**Alternate flow (Does Not Apply)**

**Exceptional flow**

E1: Optimal Way of Saving the game

1. The game has loaded the level
2. The user can move there character etc. and progress through the level
3. The user brings up the in-game menu
4. The game brings up the in-game menu
5. The user selects Save Here
6. The game creates a save file
7. The game stores the file on the machine

**Termination:** The game saves the file to a target location and returns the user to the game and the level

**Post condition**: The game system goes into a waiting state, the user may save as much as they want whenever they want

### *Use Case 3: Load Game*

#### Description & Priority

The user must be able to start the game from the .exe file (see requirement 1 for the number of ways to achieve this), the game should then show the user the main menu. From there a user must be able to click load game, from there they can load a previous saved version of the game and continue on.

#### Use Case

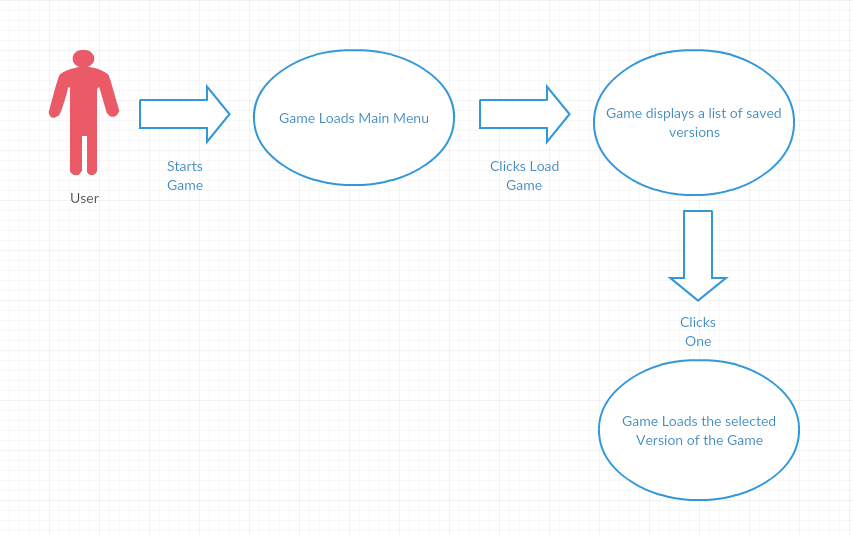
**Scope**

The scope of this use case is to allow the user to load a saved version of the game, hence continuing on from where they were last

**Description**

This use case describes the function of loading an instance of the game, the loaded game will allow the user to pick up where they last left off. Insuring they saved the game previously.

**Use Case Diagram**

****

***Flow Description***

**Precondition**

The Game is installed and the Twin Darkness game icon is on the desktop of the machine. Your desktop or laptop should meet the minimum requirements. The user should also have saved the game at least once (see Requirement 2).

**Activation**

This use case starts when a User/Player starts the game from the .exe file and the main menu loads.

**Main flow**

1. The user starts the game from the .exe file
2. The game loads the main menu
3. The user selects the Load Game button on the main menu
4. The game displays a list of saved game files
5. The user selects the one they want
6. The game starts from that saved location

**Alternate flow**

A1: Load a saved file from an active instance of the game

1. User brings up the in-game menu
2. User selects the Load New Game from the in-game menu
3. The Use Case continues at position 4 of the main flow

**Exceptional flow**

E1: Optimal Way of Loading the game

1. The user starts the game from the .exe file
2. The game loads the main menu
3. The user selects the Load Game button on the main menu
4. The game displays a list of saved game files
5. The user selects the one they want
6. The game starts from that saved location

**Termination:** The game loads the save file and starts the game from that location.

**Post condition**: The load game system is inactive during the gameplay

### *Use Case 4: Exit Game*

#### Description & Priority

The user must be able to exit the game from either the main menu or the in-game menu at any time during the instance of the game.

#### Use Case

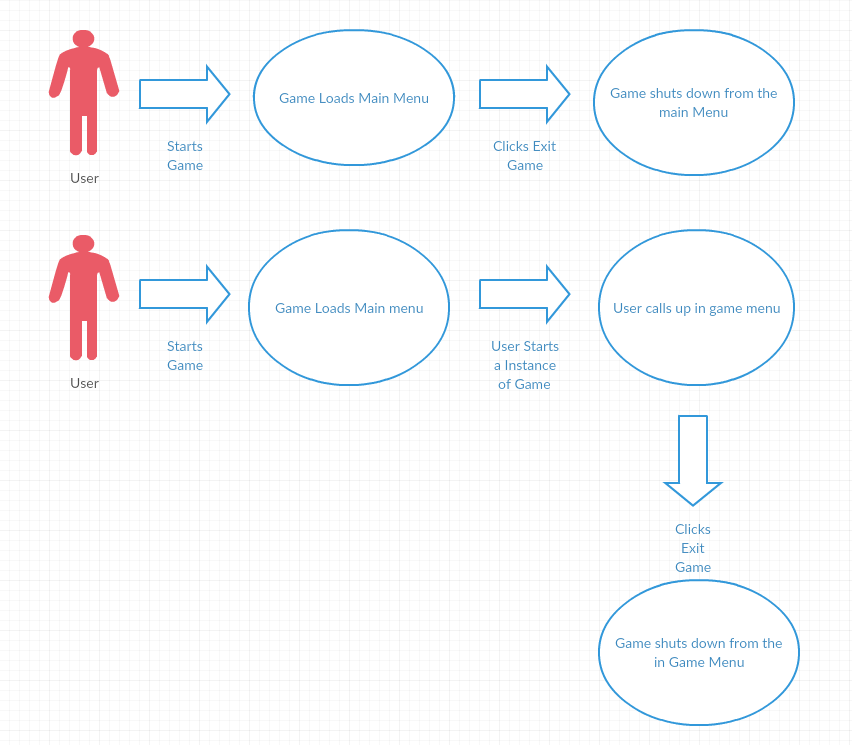
**Scope**

The scope of this use case is to allow the user to exit the game from either the main menu or the in-game menu.

**Description**

This use case describes the function of exiting the game from either the main menu or the in-game menu at any time they want, the game will then shut down and the user’s desktop will be displayed again and all game functions will have stopped.

**Use Case Diagram**

****

***Flow Description***

**Precondition**

The Game is installed and the Twin Darkness game icon is on the desktop of the machine. Your desktop or laptop should meet the minimum requirements.

**Activation**

This use case starts when a User/Player starts the game from the .exe file and the main menu loads.

**Main flow**

1. The user starts the game from the .exe file
2. The game loads
3. The game displays the main menu
4. The user clicks exit
5. The game shuts down all functions
6. The user is then returned to their desktop screen

**Alternate flow**

A1: Exit from the in-game menu

1. The user starts the game from the .exe file
2. The game loads
3. The game displays the main menu
4. The user starts an instance of the game
5. The user then brings up the in-game menu
6. The game bring up the in-game menu
7. The Use Case continues at position 4 of the main flow

**Exceptional flow**

E1: Optimal Way of Loading the game

1. The user starts the game from the .exe file
2. The game loads
3. The game displays the main menu
4. The user clicks exit

**Termination:** The game exits correctly and functions are stopped

**Post condition**: The game has been stopped successfully

### *Use Case 5: Start a new Game/Modify Settings*

#### Description & Priority

The user must be able to start the game from the .exe file (icon), choose settings to match their machine and start a new instance of the game. This requirement has a massive priority as the game must start for the user.

#### Use Case

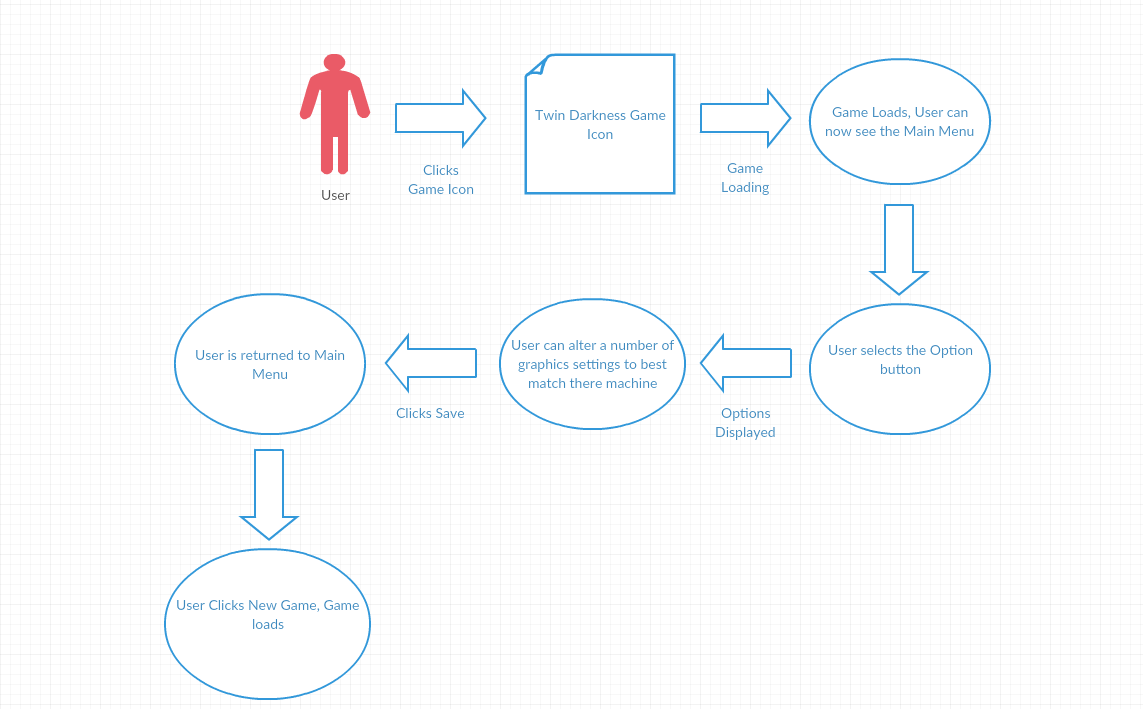
**Scope**

The scope of this use case is to allow the user to start a new instance of the game and choose settings that match their machines

**Description**

This use case describes the function of starting a new instance of the game from the .exe file icon of the desktop screen and allowing the user to modify the settings to best match their machines capability for optimal gameplay.

**Use Case Diagram**

****

***Flow Description***

**Precondition**

The Game is installed and the Twin Darkness game icon is on the desktop of the machine. Your desktop or laptop should meet the minimum requirements.

**Activation**

This use case starts when a User/Player clicks the desktop icon for the game Twin Darkness, which is a shortcut to the .exe file that was installed when installing the game.

**Main flow**

1. The User clicks the Twin Darkness game icon on the desktop
2. The Game loads to the Main menu
3. The User clicks the option menu from the main menu display screen
4. The Game loads the default options
5. The User can then modify or leave the options the way there are
6. The User saves there options
7. The Game returns the user to the main menu
8. The User clicks Start New Game
9. The Game loads

**Alternate flow**

A1: Start Game from the .exe file in the install folder

1. The User finds the install location of the Twin Darkness Game, and locates the .exe file and clicks that instead of the desktop shortcut
2. The Use Case continues at position 2 of the main flow

A2: Start Game from the .exe file in a download zip folder

1. The user downloads the Twin Darkness Zip folder from the site
2. The system downloads
3. The user unzips the folder
4. The user finds the .exe file and clicks it
5. The Use Case continues at position 2 of the main flow

**Exceptional flow**

E1: Optimal Way of starting the game

1. The User clicks the Twin Darkness game icon on the desktop
2. The Game loads to the Main menu
3. The User clicks Start New Game
4. The Game loads

**Termination:** The user selects the exit button on the main menu, the system shuts down and returns the user to their desktop.

**Post condition**: The game system goes into a waiting state on the desktop until started up again by the user

### *Use Case 6: Load/Complete Tutorial Level*

#### Description & Priority

The user must be able to start the game and load the main menu, from there they must be able to select Tutorial Level from the main menu. The tutorial level will be for first time users.

#### Use Case

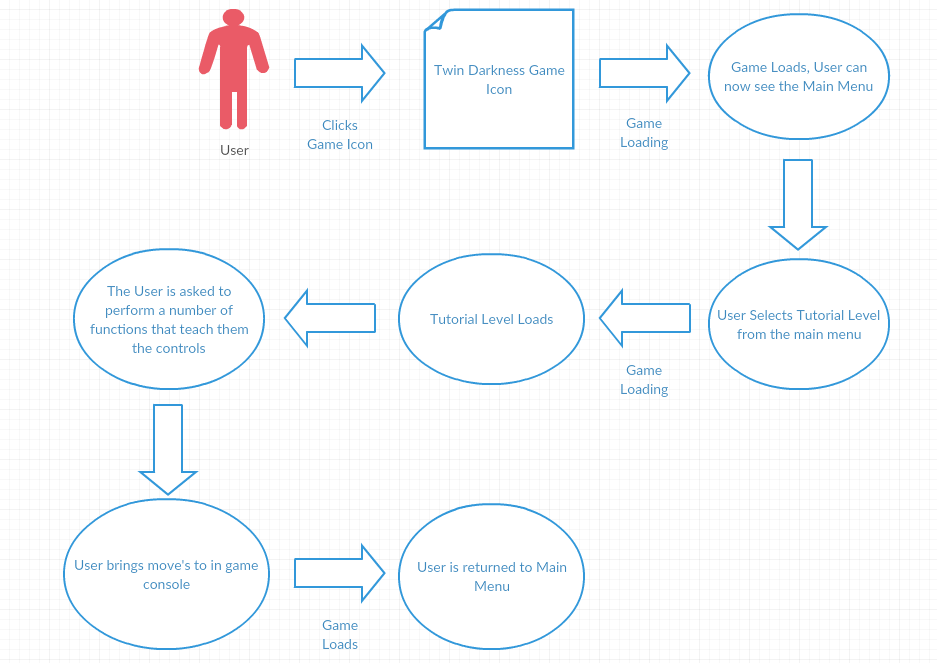
**Scope**

The scope of this use case is to allow the user to load the tutorial level which is designed for first time players and can be used as a platform for testing the game on your system

**Description**

This use case describes the function of loading the simple tutorial level, so that first time players that either need hints or want to test what options work best, this tutorial will be simple to understand and will not be part of the main story.

**Use Case Diagram**

****

***Flow Description***

**Precondition**

The Game is installed and the Twin Darkness game icon is on the desktop of the machine. Your desktop or laptop should meet the minimum requirements.

**Activation**

This use case starts when a User/Player starts the game from the .exe file and the main menu loads.

**Main flow**

1. The user starts the game from the .exe file
2. The game loads to the main menu
3. The user selects the Tutorial level button on the main menu
4. The game loads the tutorial level
5. The user is then asked by the game to complete some simple tasks to show them the controls for the game.
6. The user can then move around and check if there options are ok to play the game
7. The user then clicks and in-game console to exit
8. The game returns the player to the main menu

**Alternate flow (Does Not Apply)**

**Exceptional flow**

E1: Optimal Way of Loading and completing the tutorial level

1. The user starts the game from the .exe file
2. The game loads to the main menu
3. The user selects the Tutorial level button on the main menu
4. The game loads the tutorial level
5. The user is then asked by the game to complete some simple tasks to show them the controls for the game.
6. The user then clicks and in-game console to exit
7. The game returns the player to the main menu

**Termination:** The game exits correctly back to the main menu after the user uses the in-game console.

**Post condition**: The tutorial level has been completed it will always be available from the main menu for the user

### *Use Case 7: Interact with in-game items/systems*

#### Description & Priority

The user must be able to start an instance of the game (hence load or start a level) during the course of the game the user must be able to interact with the in-game items and systems (Examples: unlock/lock doors, use elevators, find key cards, find papers etc.) These functions have a massive priority as they are key functions of the core game.

#### Use Case

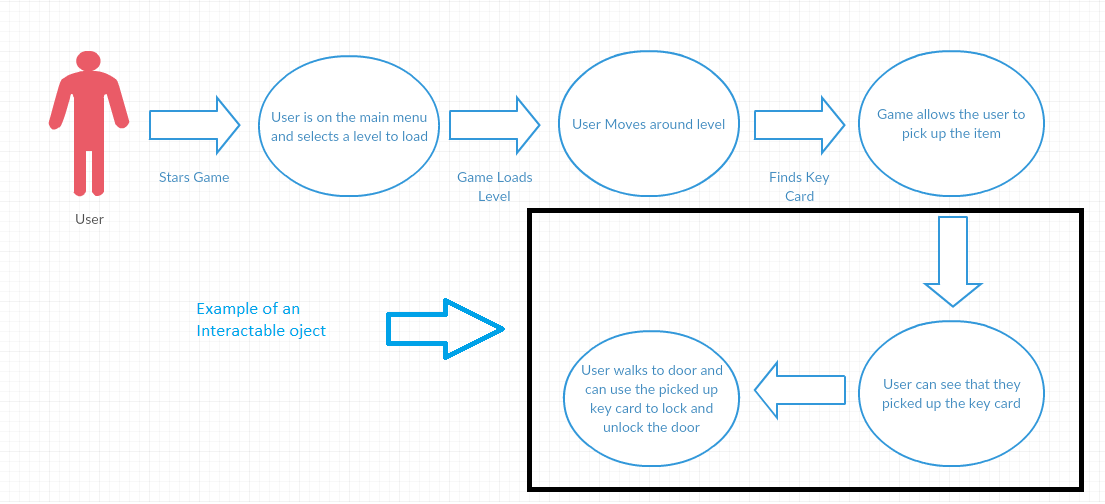
**Scope**

The scope of this use case is to allow the user to interact with all the different in-game items and systems.

**Description**

This use case describes the function of the user interacting with the in-game items and systems, Example of such is being able to find a key card and using it to unlock/lock a door. These functions are key to the core gameplay

**Use Case Diagram**

****

***Flow Description***

**Precondition**

The Game is installed and the Twin Darkness game icon is on the desktop of the machine. Your desktop or laptop should meet the minimum requirements. A level is loaded and the user is in game.

**Activation**

This use case starts when a User/Player starts the game, loads a level and finds an object that can be interacted with in-game.

**Main flow**

1. The Game loads a level
2. The user moves the character around through the level
3. The user finds an object that can be interacted with.
4. The game allows the user to pick up the item

**Alternate flow (Does Not Apply)**

**Exceptional flow**

E1: Optimal Way of Loading and completing the tutorial level

1. The Game loads a level
2. The user moves the character around through the level
3. The user finds an object that can be interacted with.
4. The game allows the user to pick up the item

**Termination:** The game lets the user pick up the item

**Post condition**: The user can pick up and use as many items as they can find, items are reusable as many times as the user desirers.

### *Use Case 8: VR Setup/Interaction*

#### Description & Priority

The user must be able to start an instance of the game (hence load or start a level) and enter the VR Mode of the Game using preferred interface Oculus Rift either using in game menu or keyboard shortcuts. Currently the default setup is Alt Enter to enter the VR mode in the Unreal engine 4. (VR input is optional the game can be played with or without it)

#### Use Case

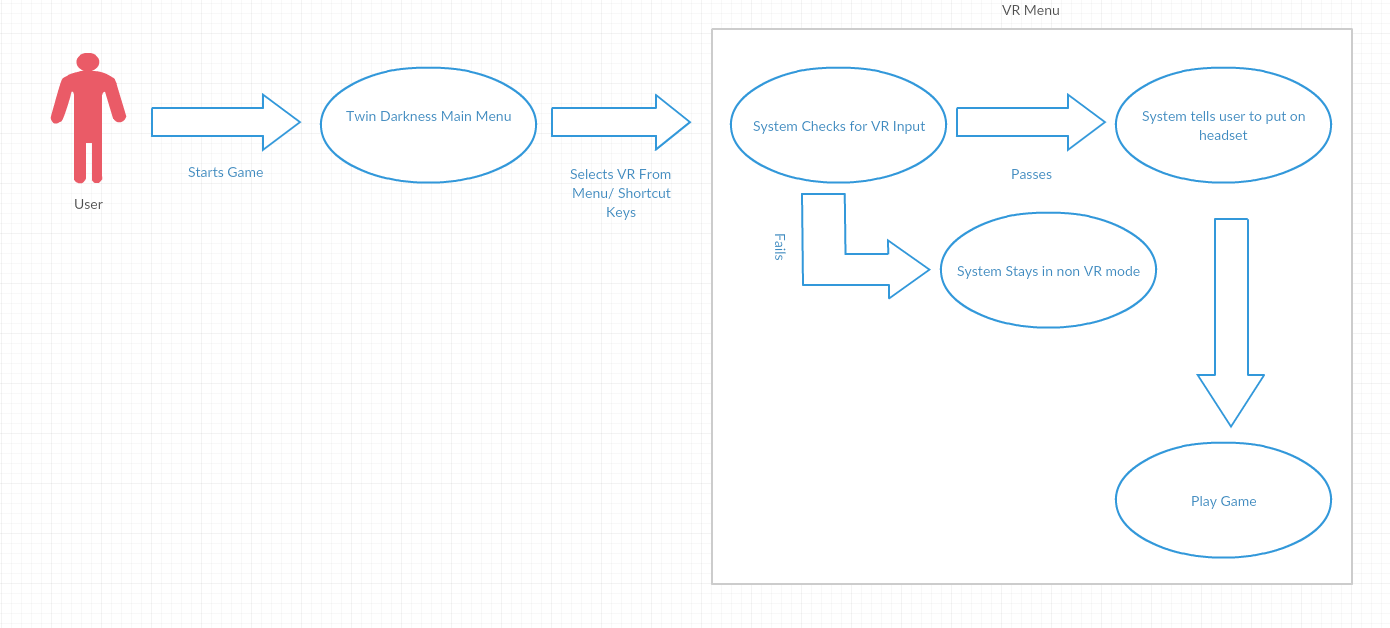
**Scope**

The scope of this use case is to allow to enter the VR mode of the Game either with main menu input or keyboard shortcuts.

**Description**

This use case describes the function of entering the VR mode of the game using the Oculus rift as the preferred input

**Use Case Diagram**

****

***Flow Description***

**Precondition**

The Game is installed and the Twin Darkness game icon is on the desktop of the machine. Your desktop or laptop should meet the minimum requirements. The user should have all the necessary setup steps for VR input.

* A VR Interface (Oculus Rift Preferred)
* All necessary VR software (Oculus Runtime) information for VR input/setup will be display on Main Menu VR page.

**Activation**

This use case starts when a User/Player starts the game, and uses one of the two ways of trigging the VR setup options. (Main Menu select or shortcut key select), VR interface must switched on and in a ready mode.

**Main flow**

1. User starts the game
2. System brings the user to the main menu
3. User selects VR input from the options menu
4. System checks if there is a suitable VR system installed and ready to go
5. If Pass, VR mode is engaged and the user is prompted to put on their headset
6. If Fail, VR mode is not engaged and user is kept in normal mode.
7. User plays the game in what mode is selected.

**Alternate flow (Does Not Apply)**

**Exceptional flow**

E1: Optimal Way of Loading and completing the tutorial level

1. User starts the game
2. System brings the user to the main menu
3. User selects VR input from the options menu
4. System checks if there is a suitable VR system installed and ready to go
5. If Pass, VR mode is engaged and the user is prompted to put on their headset
6. If Fail, VR mode is not engaged and user is kept in normal mode.
7. User plays the game in what mode is selected.

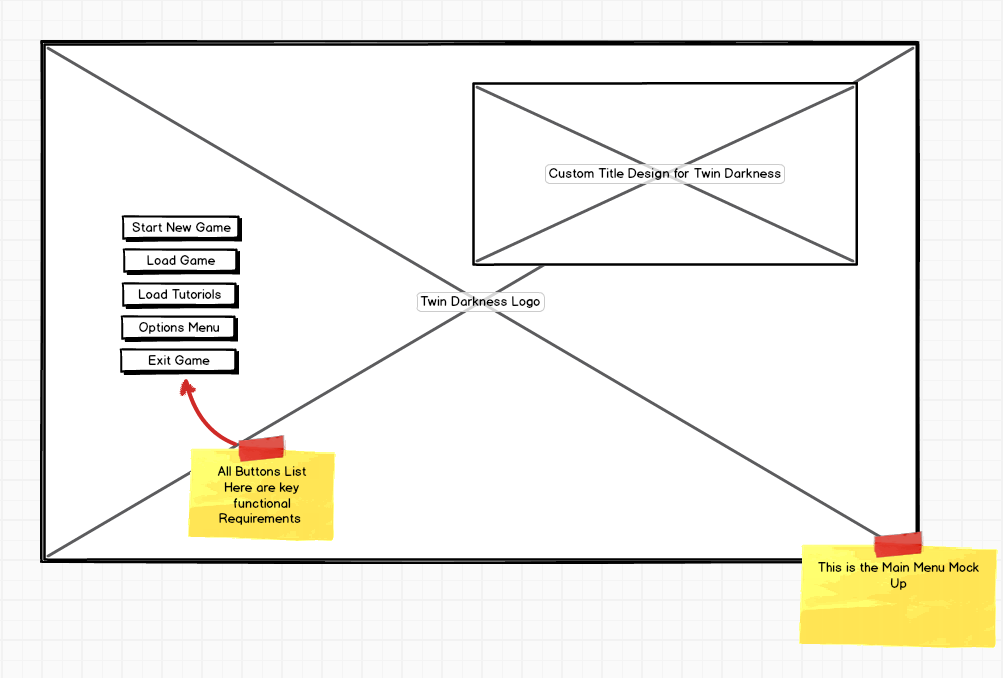
**Termination:** The game engages into either one of the game modes VR or Normal.

**Post condition**: The user plays the game in their selected game mode.

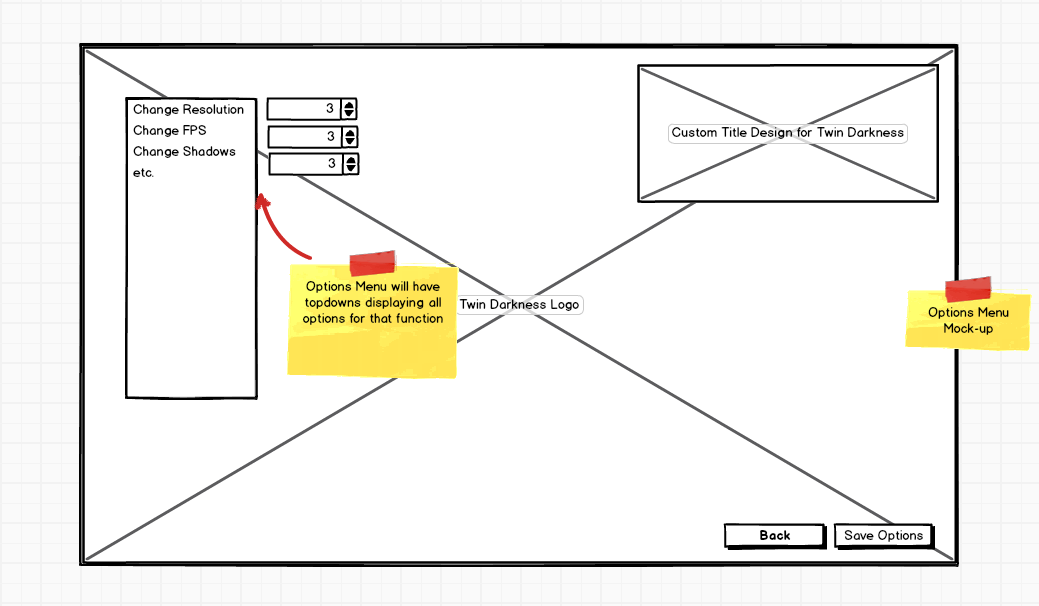
## 

## *User Interface Design*

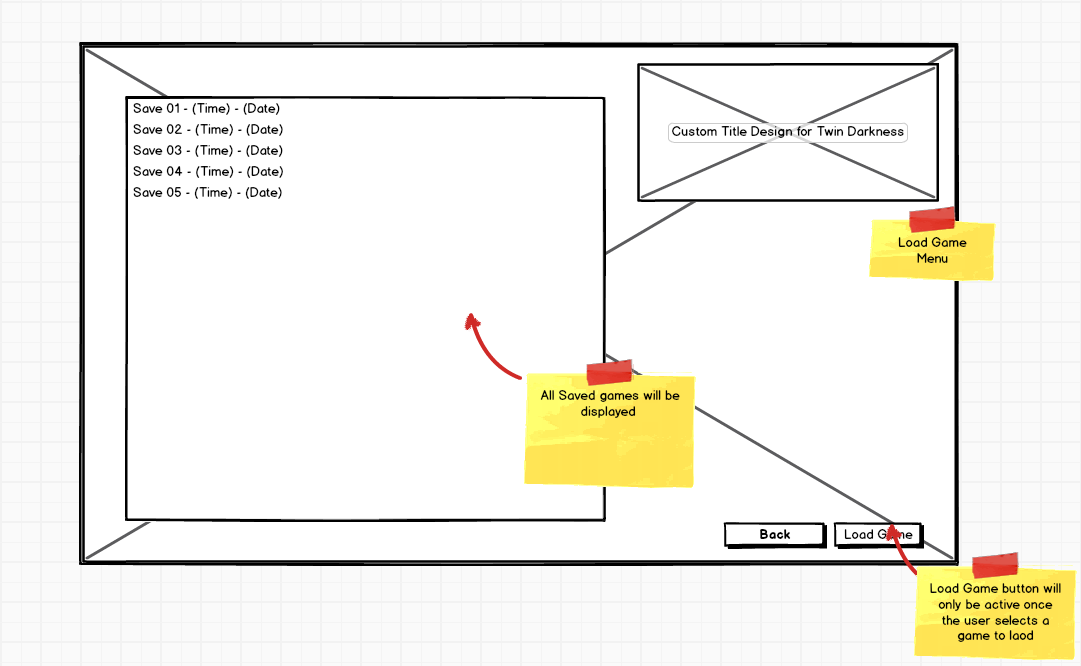
*Mock-up of the Main Menu*

**

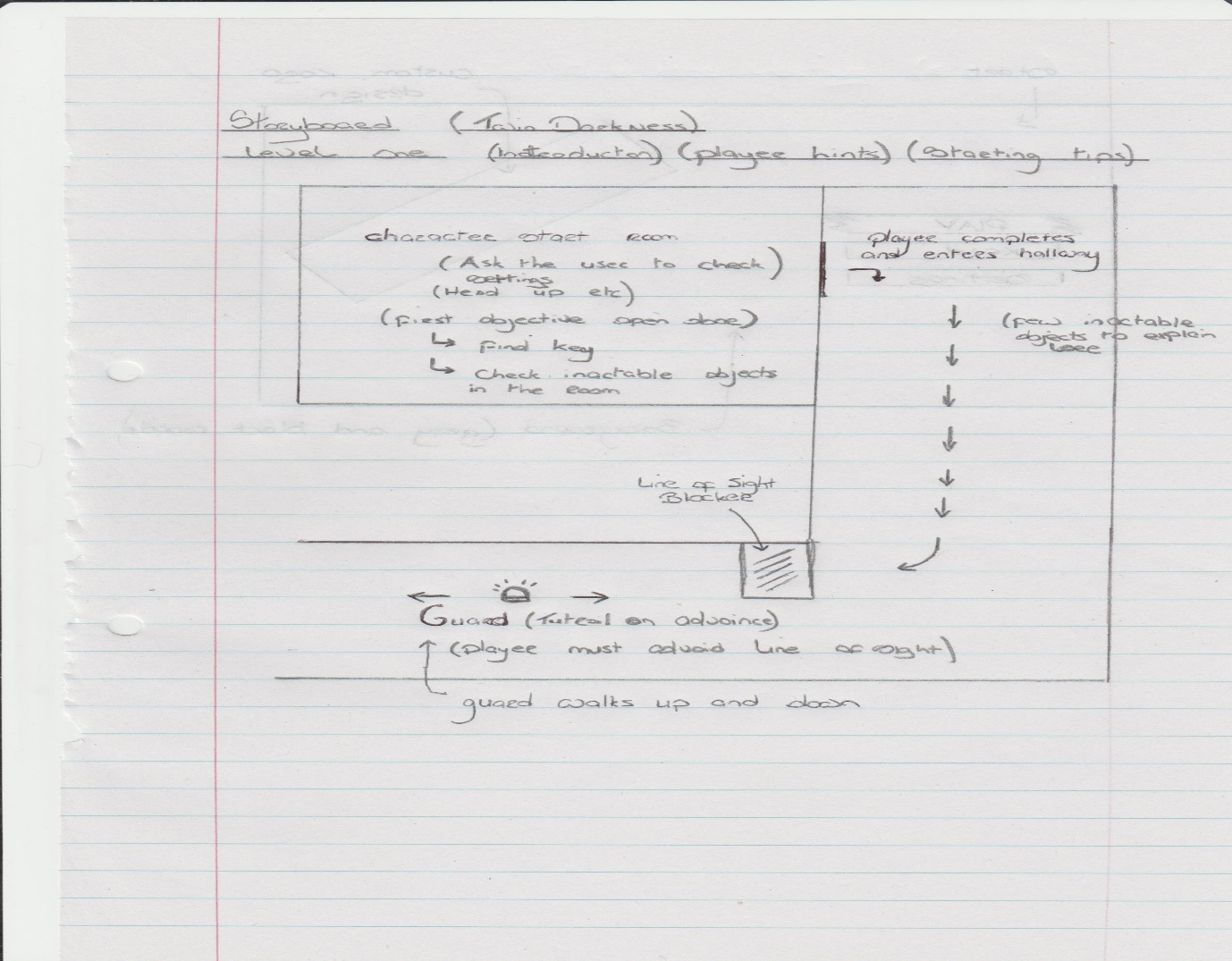
*Mock-up of the Options Menu*



*Mock-up of the Load Game Menu*

**

Sample of the level design



## *Performance*

The **performance** is decided by weather the machine being used meets the minimal or recommended settings for unreal engine 4

**Minimal Settings:**

*Windows 7,8 or 10, (Windows 7 preferred choice)*

*Minimum of 4GB of RAM installed*

*Dual-Core Intel or AMD processor, 2.0 GHz or Faster*

*NVIDIA GeForce 470 GTX or AMD Radeon 6870 HD series Card or Higher*

*Desktop PC, Mac or Laptop*

**Recommended Settings:**

*Desktop PC*

*Windows 7 64-bit or Mac OS X 10.9.2 or later*

*Quad-Core Intel or AMD processor, 2.5 GHz or faster*

*NVIDIA GeForce 470 GTX or AMD Radeon 6870 HD series Card or Higher*

*8 GB Ram*