National College of Ireland

BHs in Computing

2017 / 2018

Stephen Lowry

x12356291

A.A.R

Arena Assault Racing

Technical Report

Table of contents

1: Executive Summary

The problem I have set out for this project is to build a fast paced & exiting racing game with great replicability. Arena Assault Racing is a game what will allow users to control weaponized vehicles to do battle with against AI & human controlled opponents. The user will be able to play with players as they will be able to join the game with local split screen multiplayer mode. The game will feature a last man standing game mode. The game will be built in a 3D environment & will give a the user 3rd person view of the car that they are controlling. The inspiration of the game comes from the such games as Crash Team Racing & Mario Kart series, battle modes. The development of this software will be done by using the game engine Unity 5. The game will be programmed primarily with C# which is the default language for Unity & Visual Studios which will be needed to for creating the code. Imaging creating & editing for the user interface & textures will be done with the help go Gimp. Blender will be used to create complex 3D model shapes & game objects for the game. Audacity will be used to edit music & sound effects to be implemented into the game.

2: Introduction

Background

Arena Assault Racing (or A.A.R for short) is a game that allows players to battle it out with weaponized vehicles against likewise AI & human enemies. The game features last man standing game. The users aim is to destroy all enemies in the arena & to try & achieve the a high score. Scoring will be based of the winners health remaining & the time it took them to win.

A.A.R is heavily influenced by the battle modes in such games as Crash Team Racing on the PlayStation & the Mario Kart series from Nintendo. These games will be a great aid for studying what gameplay features I should bring with me into A.A.R & to where I can improved upon them.

https://store.playstation.com/en-ie/home/games#!/en-nz/games/ctr-crash-team-racing/cid=EP9000-NPEE00026_00-GCRASHTEAM000001

https://kotaku.com/mario-kart-8-switch-makes-battle-mode-so-much-better-1794448420

Aims

- Create a simple gameplay system, that will allow users at any skill level be able to play.
- Create a simple UI design to help the user navigate through the game's menus.
- Create an arena that will allow the player(s) to do battle on.
- Have 3D models & animations for the user's cars, enemies, weapons & arenas.
- Create & implement music & sound effects into the game to help make the game more immersive.
- Implement an AI for the games enemies to be able to attack.
- Build a local stored database to allow users highest score to be saved.
- Test a games prototype with a number gamers to gather feedback.
- Test the usability of the program.
- Undertake unit testing of the functionalities of the applications once its completed.

Technologies

Unity:

- Cross-platform game engine developed Unity Technologies.
- Can be used to create 2D & 3D games.
- This program uses programming language such as C, C++ for run time & C# as the Unity API.
- Provides great support for development with the use of asset prefabs that can be obtainable via the asset store & other online services.
- The software is free for non-commercial use.
- Unity will be used to create

https://venturebeat.com/2014/10/23/john-riccitiello-sets-out-to-identify-the-engine-of-growth-for-unity-technologies-interview/

GIMP:

- GIMP is an open source image editing software.
- Originally created by The GIMP Development Team.
- Image editing software what will allow me to create environments, sprites, user interface layout & designs.
- Software is free.

https://www.gimp.org/

Blender

- Blender will be used to create & edit 3D models that will be implemented into the game.
- Created by the Blender Foundation.
- This software is free to download

https://www.blender.org/

Audacity

- Audacity is an open source sound editing software.#
- Created by
- This tool will be used to create & edit in game sound effects & music.
- This software is free to use.

https://www.audacityteam.org/

Microsoft's Visual Studios

- Microsoft's Visual Studio will be used to help create & edit the backend code in conjunction with Unity.
- Created by Microsoft.
- Visual Studios community version is free to obtain.

https://www.visualstudio.com

3: Structure

System requirements specifications

Information of the software system been developed. It shows off the functional & non-functional requirements. It also includes a use case & class diagrams that explains user interactions that the software will give.

Testing

An investigation into on how well the A.A.R's plays & looks. The test that will include usability, customer & unit testing. An evaluation of results, data received & feedback will also be documented.

Conclusion

Reflection on the project in hand. This section discusses what target where meet, what was missed & what would be done differently next time.

Further Development or Research

With more time & resources, where could the results of this project lead to?

References

Content & sources used to help create this document & project. These contain websites, video tutorials & books.

Appendix

Documentation & information gathered over the course of this project.

Definitions, Acronyms, & Abbreviations

AI	:	Arterial Intelligence
UI	:	User Interface
GUI	:	Graphical Interface
3D	:	3 Dimensions
AWS	:	Amazon Web Services
PC	:	Personal Computer
CPU	:	Central Processing Unit
GPU	:	Graphical Processing Unit
RAM	:	Random Access Memory
OS	:	Operation System
JDK	:	Java Development Kit
SDK	:	Software Development Kit
NDK	:	Native Development Kit
KD	:	Kill/Death Ratio
N/A	:	Not Applicable
e.g.	:	Example
DB	:	Database
WebGL	:	Web Graphics Library
API	:	Application Programming Interface
SteamOS	:	Linux-based Operating System
Unity	:	Game engine, game development software

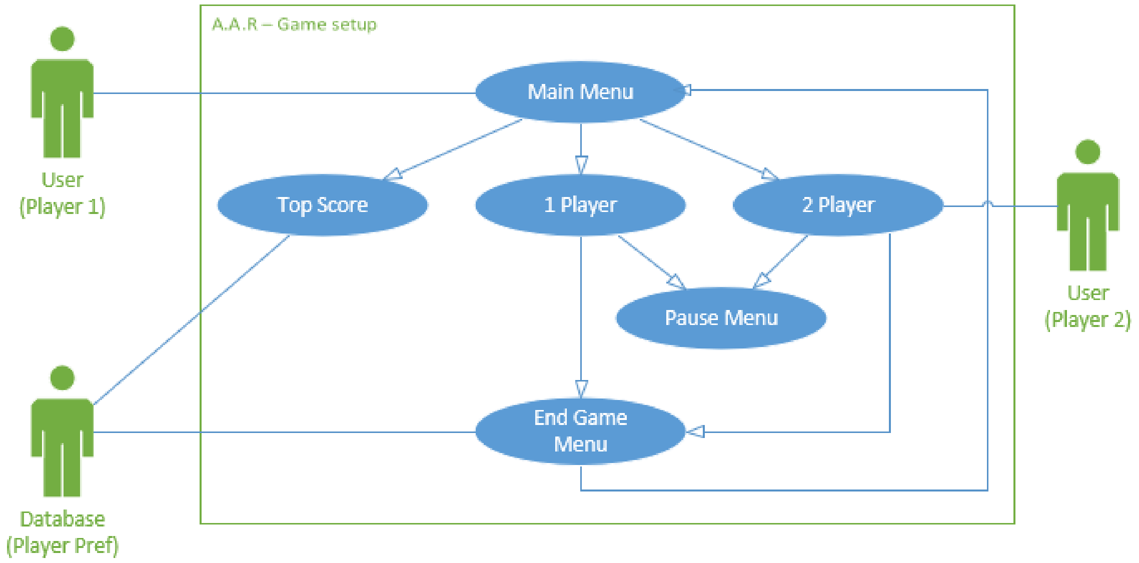
4: System Requirements Specification

Arena Assault Racing will be designed to be an easy to pick up & play game. Instructions on how to play the game will & can be kept to a minimal such as the inclusion of a map or list of the games controls inputs.

Functional Requirements

The most important functional requirement for A.A.R is to allow the user to set up, play, finish & start a new game. The user should also be able to set up multiplayer game with a split screen.

Use Case Diagram



Requirement 1 (Creating Single player game)

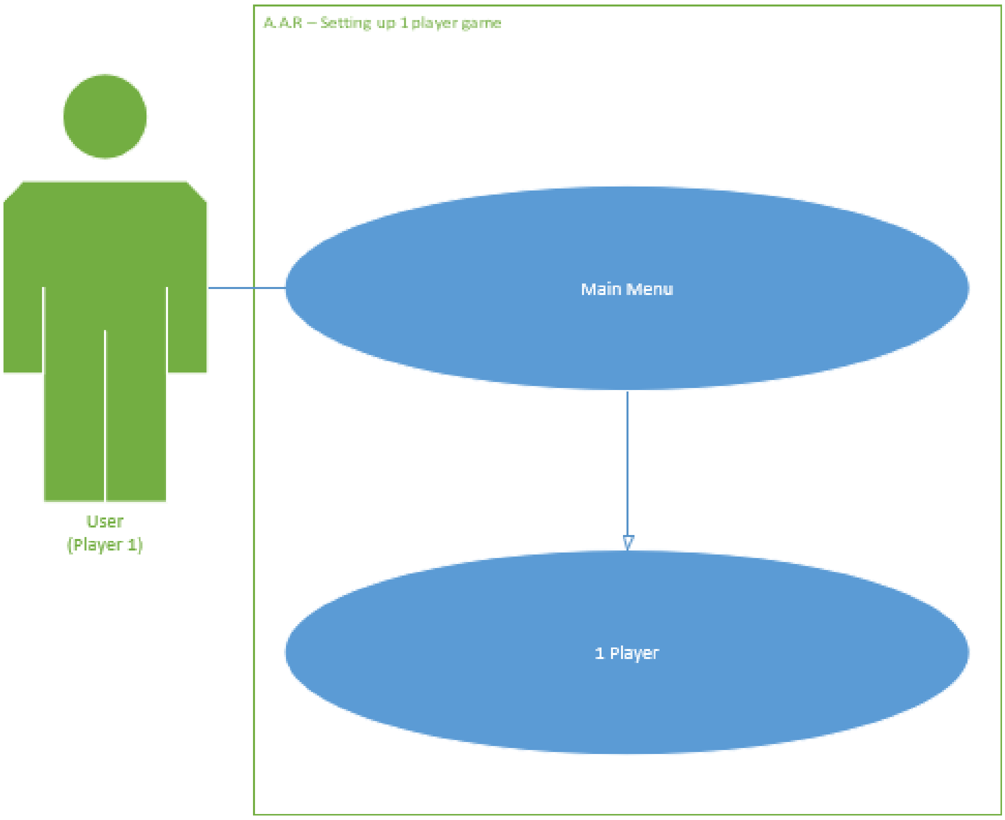
Description & Priority

Creating a single player game against AI controlled enemies will be a highly critical feature to be implement into the game. The user will need to be able to start a game from the main menu scene.

Scope

The scope of this use case is to show how a user will be able to create a single player game.

Use Case Diagram



Flow Description

Precondition

The user has starts the program & selects 1 player in the home menu. The user will be brought into the game & the round will start.

Activation

The use case starts when the user opens the game or is located in the main menu.

Main flow

The system identifies the main menu.
The <User (Player 1)> selects 1 Player button.
The system brings the user to the 1 player scene.
The <User (Player 1)> can now start to play the game in single player.
The player is able have control over there corresponding car.

Termination

The user starts to plays the game.

Post condition

The system goes into a wait state.

Requirement 2 (Creating 2 player game)

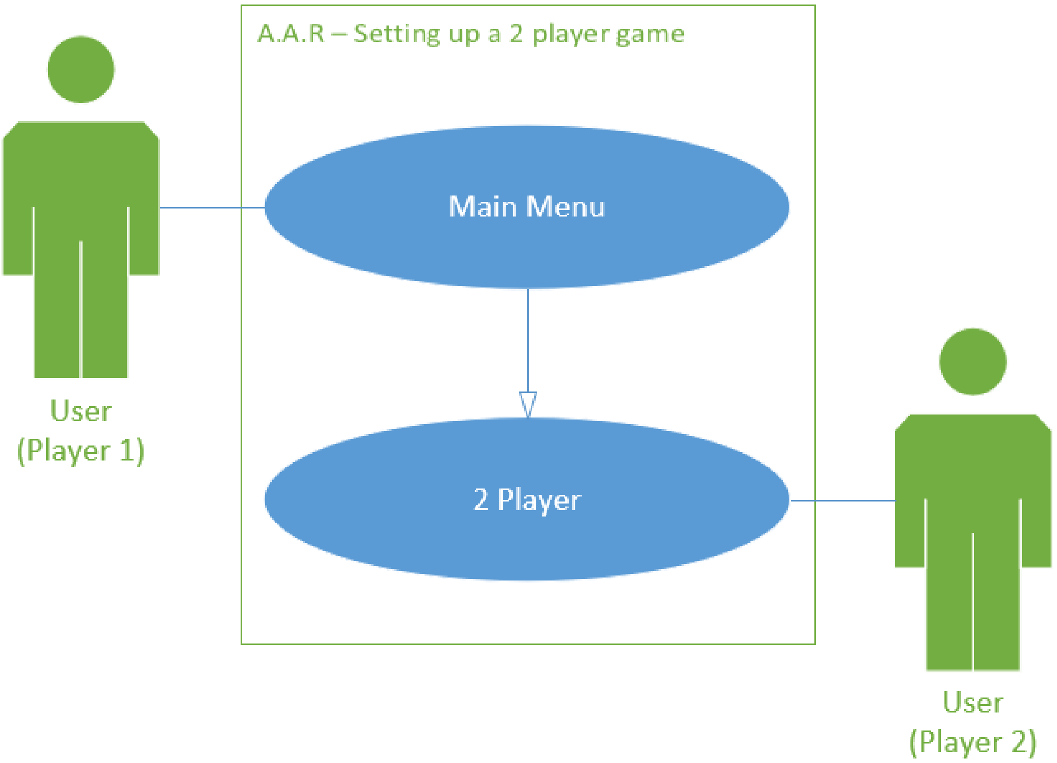
Description & Priority

Creating a 2 player game against other AI controlled enemies will be a highly critical feature to be implement into the game. The user will need to be able to start a 2 player game from the main menu scene. This requirement is to be set at a high priority as the user(s) will need to able to start a the game.

Scope

The scope of this use case is to show how a user will be able to create a multiplayer player game.

Use Case Diagram



Flow Description

Precondition

The user has starts the program & selects 2 player in the home menu. The user will be brought into the game & the round will start. Player 2 will be able to control there car from this point.

Activation

The use case starts when the user opens the game or is located in the main menu.

Main flow

The system identifies the main menu.
The <User (Player 1)> selects 2 Player button.
The system brings the user to the 2 player scene.
The <User (Player 1)> & <User (Player 2)> can now start to play the game in with a spilt screen. Each player only have control over there corresponding car.

Termination

The users starts to plays the game.

Post condition

The system goes into a wait state.

Requirement 3 (Pausing gameplay while in game)

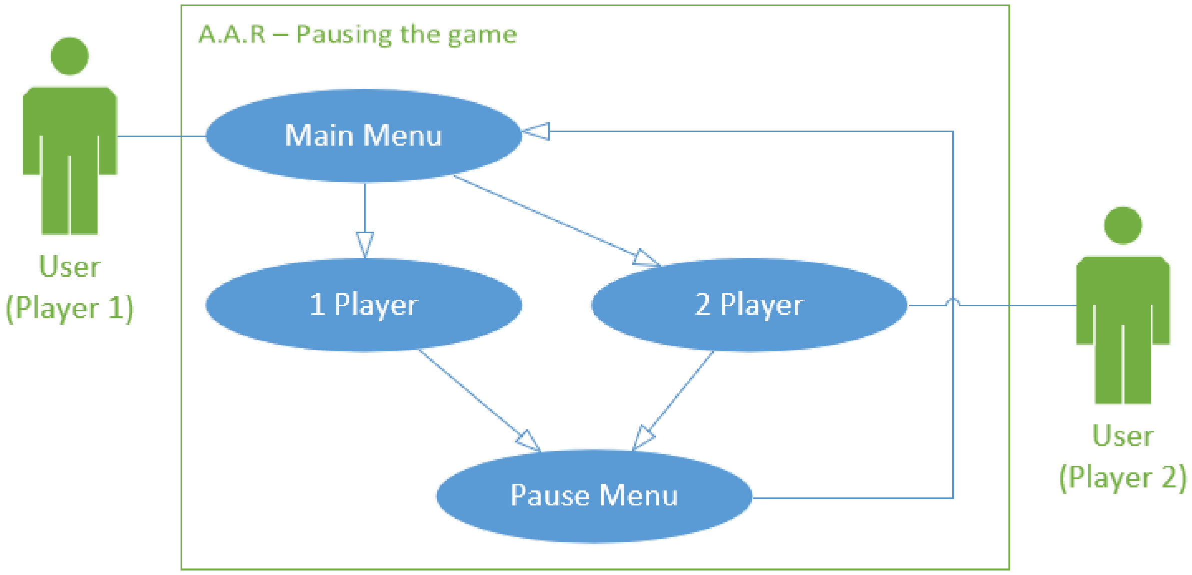
Description & Priority

This requirement to allow the user(s) to pause the gameplay of the game. This will allow the user to take needed break & stoppage during a round without impacting the games result. The pause feature will display a pause menu which can be used to navigate back into & continue the game or quit back to the menu. This requirement would be set to a low priority as it does not impact in playing the game.

Scope

The scope is to be to allow the user(s) to freeze the game, display a pause menu UI & give the user a choice to select to leave or continue the game.

Use Case Diagram



Precondition

The user has starts the program & selects to play the 1 player or 2 player mode in the home menu. The user will be brought into the game & the round will start.

Activation

The use case starts when the user opens the game or is located in the main menu. User can only pause the game when in game (1 Player or 2 Player).

Main Flow (Player 1 pauses the game & wants to resume the round in a 1 player game)

The system identifies the main menu.
The <User (Player 1)> selects 1 Player button.
The system brings the user to the 1 player scene.
The <User (Player 1)> can now start to play the game.
The player is able have control over there corresponding car.
The <User (Player 1)> wants to pause the game to take a phone call & presses the pause button (e.g. ESC key).
The system freezes gameplay, makes users control (movement & shooting) of their car in active & displays the pause menu screen.
The <User (Player 1)> user is now able to step away from the game.
The <User (Player 1)> has returned & is wanting to continue their game.
The <User (Player 1)> presses the pause button again (ESC key) or clicks the Resume button in the pause menu.
The system closes the pause menu, reactivates the users controls (movement & shooting) & unfreezes the gameplay.
The <User (Player 1)> can now continue to play the game where the left off.

Alternate flow 1 (Player 1 or 2 pauses the game & wants to resume the round in a 2 player game)
The system identifies the main menu.
The <User (Player 1)> selects 2 Player button.
The system brings the user to the 2 player scene.
The <User (Player 1)> & <User (Player 2)> can now start to play the game.
The player is able have control over there corresponding car.
The <User (Player 1)> or <User (Player 2)> wants to pause the game to take a phone call & presses the pause button (e.g. ESC key).
The system freezes gameplay, makes users control (movement & shooting) of their car in active & displays the pause menu screen.
The <User (Player 1)> or <User (Player 2)> user is now able to step away from the game.
The <User (Player 1)> or <User (Player 2)> has returned & is wanting to continue their game.
The <User (Player 1)> or <User (Player 2)> presses the pause button again (ESC key) or clicks the Resume button in the pause menu.
The system closes the pause menu, reactivates the users controls (movement & shooting) & unfreezes the gameplay.
The <User (Player 1)> or <User (Player 2)> can now continue to play the game where the left off.

Alternate flow 2 (Player 1 wants to quit the round in a 1 player game)
The system identifies the main menu.
The <User (Player 1)> selects 1 Player button.
The system brings the user to the 1 player scene.
The <User (Player 1)> can now start to play the game.
The player is able have control over there corresponding car.
The <User (Player 1)> wants to quit the round they are playing.
The <User (Player 1)> presses the pause button (e.g. ESC key).
The system freezes gameplay, makes users control (movement & shooting) of their car in active & displays the pause menu screen.
The <User (Player 1)> clicks the Menu button in the pause menu.
The system closes game & returns the user to the main menu.

Alternate flow 3 (Player 1 or 2 wants to quit the round in a 2 player game
The system identifies the main menu.
The <User (Player 1)> selects 2 Player button.
The system brings the user to the 2 player scene.
The <User (Player 1)> & <User (Player 2)> can now start to play the game.
The player is able have control over there corresponding car.
The <User (Player 1)> or <User (Player 2)> wants to quit the round they are playing.
The <User (Player 1)> or <User (Player 2)> presses the pause button (e.g. ESC key).
The system freezes gameplay, makes users control (movement & shooting) of their car in active & displays the pause menu screen.
The <User (Player 1)> or <User (Player 2)> clicks the Menu button in the pause menu.
The system closes game & returns the user(s) to the main menu.

Exceptional flow 1 (The user(s) don't pause the game)
The system identifies the main menu.
The <User (Player 1)> selects 1 Player button.
The system brings the user to the 1 player scene.
The <User (Player 1)> can now start to play the game.
No pause action was taken.

Exceptional flow 2 (The user(s) don't pause the game)
The system identifies the main menu.
The <User (Player 1)> selects 2 Player button.
The system brings the user to the 2 player scene.
The <User (Player 1)> & <User (Player 2)> can now start to play the game.
No pause action was taken.

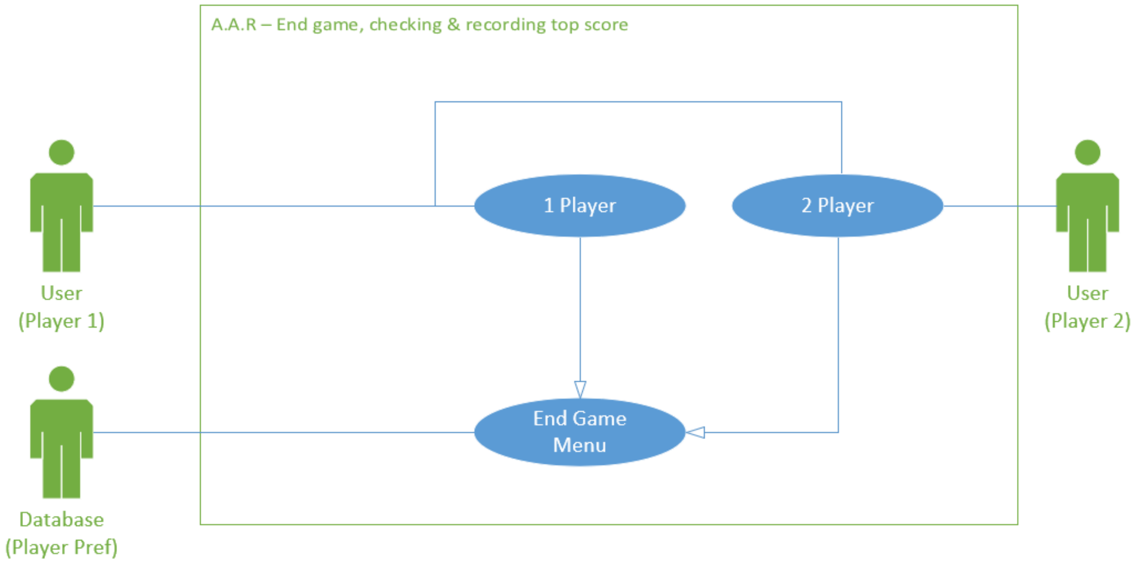
Termination
The user(s) continue playing there game or return to the main menu.

Post condition
The system goes into a wait state

Requirement 4 (Ending the game, recording & checking for new top score)
Description & Priority
In A.A.R when the user(s) wins a round they will get a score on how well they did. Using the equation $\left(\frac{1}{TIME+1}\right) \cdot HEALTH \cdot 10^2 = SCORE$ to calculate how well the player did. If the user score is greater than the current top score there score will be saved & they will be greeted with message notifying them on their achievement. This requirement should be set as a medium to high priority as it

Scope
The game will need to get the players health & the time of when the player defeats all there other cars & win.

Flow Description



Precondition
The user starts the application & selects to play a 1 or 2 player game.

Activation
This use case starts when <User (Player 1)> starts 1 or 2 player a game.

Main flow (Player 1 wins a 1 player game without top score)
The system brings the user to the 1 player scene.
The <User (Player 1)> can now start to play the game.
The player is able have control over there corresponding car.
The <User (Player 1)> defeats all AI controlled cars.
The system gets the players health, the time it took to complete at the end of the round & calculates the players score for this round.
The system checks score to the top score saved in the database (Player Pref), score is less than top score.
The system freezes gameplay, makes users control (movement & shooting) of their car in active & displays the end game menu screen with the players score.

Alternate flow 1 (Player 1 wins a 1 player game with top score)
The system brings the user to the 1 player scene.
The <User (Player 1)> can now start to play the game.
The player is able have control over there corresponding car.
The <User (Player 1)> defeats all AI controlled cars.
The system gets the players health, the time it took to complete at the end of the round & calculates the players score for this round.
The system checks score to the top score saved in the database (Player Pref), score is greater than top score.
The system replaces the old top score the players score for this round.
The system freezes gameplay, makes users control (movement & shooting) of their car in active & displays the end game menu screen notifying the user of the new top score achievement & the players score.

Alternate flow 3 (Player 1 wins & player 2 dies in a 2 player game without top score)
The system brings the user to the 2 player scene.
The <User (Player 1)> & <User (Player 2)> can now start to play the game.
The players are able have control over their corresponding car.
The <User (Player 1)> defeats all AI & <User (Player 2)> controlled cars.
The system gets the <User (Player 1)> health, the time it took to complete at the end of the round & calculates the players score for this round.
The system checks score to the top score saved in the database (Player Pref), score is less than top score.
The system freezes gameplay, makes users control (movement & shooting) of their car in active, displays the end game menu screen displaying playing player 1 wins & their score.

Alternate flow 4 (Player 1 wins player 2 dies in a 2 player game with top score)
The system brings the user to the 2 player scene.
The <User (Player 1)> & <User (Player 2)> can now start to play the game.
The players are able have control over there corresponding car.
The <User (Player 1)> defeats all AI & <User (Player 2)>controlled cars.
The system gets the <User (Player 1)> health, the time it took to complete at the end of the round & calculates the players score for this round.
The system checks score to the top score saved in the database (Player Pref), score is greater than top score.
The system replaces the old top score the players score for this round.
The system freezes gameplay, makes users control (movement & shooting) of their car in active, displays the end game menu screen notifying the player 1 wins with a new top score achievement & the players score.

Alternate flow 5 (Player 2 wins & player 1 dies in a 2 player game without top score)
The system brings the user to the 2 player scene.
The <User (Player 1)> & <User (Player 2)> can now start to play the game.
The players are able have control over their corresponding car.
The <User (Player 2)> defeats all AI & <User (Player 1)> controlled cars.
The system gets the <User (Player 2)> health, the time it took to complete at the end of the round & calculates the players score for this round.
The system checks score to the top score saved in the database (Player Pref), score is less than top score.
The system freezes gameplay, makes users control (movement & shooting) of their car in active, displays the end game menu screen displaying playing player 2 wins & their score.

Alternate flow 6 (Player 1 wins player 2 dies in a 2 player game with top score)
The system brings the user to the 2 player scene.
The <User (Player 1)> & <User (Player 2)> can now start to play the game.
The players are able have control over there corresponding car.
The <User (Player 2)> defeats all AI & <User (Player 1)>controlled cars.
The system gets the <User (Player 2)> health, the time it took to complete at the end of the round & calculates the players score for this round.
The system checks score to the top score saved in the database (Player Pref), score is greater than top score.
The system replaces the old top score the players score for this round.
The system freezes gameplay, makes users control (movement & shooting) of their car in active, displays the end game menu screen notifying the player 2 wins with a new top score achievement & the players score.

Exceptional flow (Player 1 & 2 both die & lose in a 2 player game)
The system brings the user to the 2 player scene.
The <User (Player 1)> can now start to play the game.
The players are able have control over there corresponding car.
The <User (Player 1)> & <User (Player 2)> are both defeated by the AI controlled cars.
The system freezes gameplay, makes users control (movement & shooting) of their car in active & displays the end game menu screen displaying a message that both players have died.

Termination
The class ends when the user gets the end game screen. The user can click the menu button to return to the main menu.

Post condition
The system goes into a wait state

Requirement 5 (Displaying & resetting the current top score)

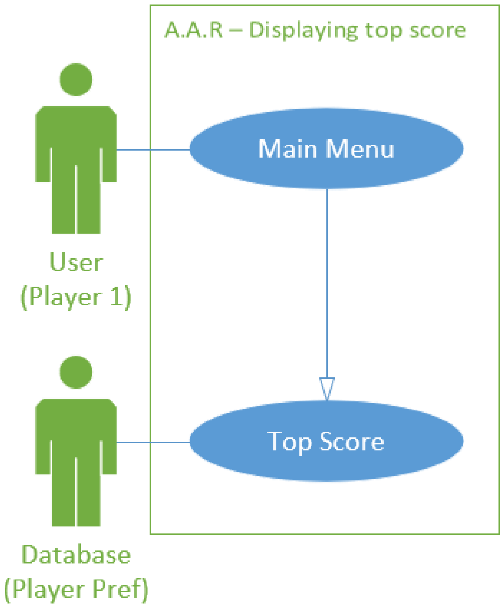
Description & Priority

A.A.R will allow users to see what their currently top score. The user can reset this score at any time if they wish. This requirement is to be set at a low priority as does not affect the player from playing the game.

Scope

Displaying the top score current top score

Flow Description



Precondition

The user is at the main menu screen. A top score may or may not have been recorded before this time.

Activation

The user is at the main menu screen.

Main flow (checking the top score)

The system identifies the main menu.
The <User (Player 1)> selects Top Scores button.
The system brings the user to the Top Scores scene & loads the current to score from the database (Player Pref).
The <User (Player 1)> can now see the current top score is 1000.
The <User (Player 1)> selects Back button.
The system returns user back to the main menu

Alternate flow (Deleting the top score)

The system identifies the main menu.
The <User (Player 1)> selects Top Scores button.
The system brings the user to the Top Scores scene & loads the current to score from the database (Player Pref).
The <User (Player 1)> can now see the current top score is 1000.
The <User (Player 1)> selects Reset button.
The system deletes the top score from the database (Player Pref) & the top score is reset to 0.
The <User (Player 1)> can now see the current top score is 0.
The <User (Player 1)> selects Back button.
The system returns user back to the main men

Requirement 6 (Movement & controlling)
Description & Priority
In A.A.R a major gameplay requirement is to be able to move your car & shoot your enemies with the on board weaponry. The controls for both Player 1 & Player 2 must be different but have to be work the same. Players controls are as follows:

INPUT	PLAYER 1	PLAYER 2 (NUM PAD)
Accelerate	W	8
Reverse/brake	S	5
Steer right	D	6
Steer left	A	4
Hand brake	E	9
Shoot	Q	7
Pause	ESC	

Scope
The scope of this use case is to show how the user & other AI controlled players interact with each other.

Precondition
The player(s) must in a 1 or 2 player game be able to control their car.

Activation
The player(s) must in a 1 or 2 player game mode.

Main flow (Player moves forward)
The system starts the game.
The <User (Player 1) or (Player 2)> taps their accelerate key.
The system moves the car forward slowly.
The <User (Player 1) or (Player 2)> presses & holds their accelerate key.
The system moves the car forward faster.

Alternate flow 1 (Player moves left)
The system starts the game.
The <User (Player 1) or (Player 2)> taps their steer left & accelerate key.
The system moves the car forward slowly to the left.
The <User (Player 1) or (Player 2)> presses & holds their steer left accelerate key.
The system moves the car forward faster to the left.

Alternate flow 2 (Player moves right)
The system starts the game.
The <User (Player 1) or (Player 2)> taps their steer right & accelerate key.
The system moves the car forward slowly to the right.
The <User (Player 1) or (Player 2)> presses & holds their steer right accelerate key.
The system moves the car forward faster to the right.

Alternate flow 3 (Player hand brakes & moves left)
The system starts the game.
The <User (Player 1) or (Player 2)> taps their hand brake, steer left & accelerate key.
The system moves the car forward slowly to the left.
The <User (Player 1) or (Player 2)> taps their hand brake, presses & holds their steer left accelerate key.
The system moves the car forward faster to turns the left quicker.

Alternate flow 4 (Player hand brakes & moves right)
The system starts the game.
The <User (Player 1) or (Player 2)> taps their hand brake, steer right & accelerate key.
The system moves the car forward slowly to the right.
The <User (Player 1) or (Player 2)> taps their hand brake, presses & holds their steer right accelerate key.
The system moves the car forward faster to turns the right quicker.

Alternate flow 5 (Player use's brakes)
The system starts the game.
The <User (Player 1) or (Player 2)> presses & holds their accelerate key.
The system moves the car forward fast.
The <User (Player 1) or (Player 2)> presses & holds their reverse/brake key.
The system moves the car a stop.

Alternate flow 6 (Player use's reverse)
The system starts the game.
<User (Player 1) or (Player 2)> is stationary.
The <User (Player 1) or (Player 2)> tabs their reverse/brake key.
The system moves the car backwards slowly.
The <User (Player 1) or (Player 2)> presses & holds their reverse/brake key.
The system moves the car backwards faster.

Alternate flow 7 (Player shoots)
The system starts the game.
<User (Player 1) or (Player 2)> is stationary.
The <User (Player 1) or (Player 2)> tabs their shoot key.
The system fires a single shot.
The <User (Player 1) or (Player 2)> presses & holds their shoot key.
The system fires a multiple of shots.

Exceptional flow (Player presses nothing)
The system starts the game.
<User (Player 1) or (Player 2)> is stationary.

Post condition
The system goes into a wait state

Requirement 7 (Shooting, damage & deaths)
Description & Priority
In A.A.R a major gameplay requirement is to be able to attack & kill the other player in the arena. With the aim of the game is to be last man standing, when a user to AI car gets destroyed the is no respond need. When a car takes enough damage the will get destroyed/die. Bullet shots to will do 15 damage. When playing 2 player mode when one player dies & the other is still going the game should allow the living player to continue in the round.

Scope
The scope of this use case is to show how the players & AI's cars can take damage & be destroyed.

Flow Description
Precondition
To for cars to be destroyed, the user(s) has to be in the 1 or 2 player modes.

Activation
This use case starts when a <user (Player 1)> starts a 1 or 2 player game.
Main flow (Player 1 shoots a AI with health > the shoot damage)
The system starts the game.
The <user (Player 1)> press the shoot key.
The system plays the gun fire sound.
The system sends out a Raycast which hits AICar1.
The system plays the impact sound.
The system takes the damage (15) away from AICar1's health.

Alternate flow (Player 1 Shoots at AI with health =< the shoot damage)
The system starts the game.
The <user (Player 1)> press the shoot key.
The system plays the gun fire sound.
The system sends out a Raycast which hits AICar1.
The system plays the impact sound.
The system takes the damage (15) away from AICar1's health.
The AICar1 health is now less than 0, AICar1 will now become inactive (dies).
The <user (Player 1)> is able to continue their round if there is still other enemies alive, if else last one standing the game follows Requirement 4, Main flow & Alternate flow 1

Alternate flow 2 (Player 1 shoots Player 2 with health > the shoot damage)
The system starts the game.
The <user (Player 1)> press the shoot key.
The system plays the gun fire sound.
The system sends out a Raycast which hits <user (Player 2)>.
The system plays the impact sound.
The system takes the damage (15) away from <user (Player 2)> health.
The system will update the <user (Player 2)> health stat on their UI.

Alternate flow 3 (Player 1 Shoots Player 2 with health =< the shoot damage)
The system starts the game.
The <user (Player 1)> press the shoot key.
The system plays the gun fire sound.
The system sends out a Raycast which hits <user (Player 2)>.
The system plays the impact sound.
The system takes the damage (15) away from <user (Player 2)> health.
The <user (Player 2)> health is now less than 0, <user (Player 2)> will now become inactive (dies).
The <user (Player 2)> side of the screen will turn black with a message saying "You died".
The <user (Player 1)> is able to continue there round if there is still other enemies alive, if else last one standing the game follows Requirement 4, Alternate flow 5 or 6.

Exceptional flow (Player 1 Shoots but misses)
The system starts the game.
The <user (Player 1)> press the shoot key.
The system plays the gun fire sound.
The system sends out a Raycast which hits nothing or a wall.

Exceptional flow 2 (Player 1 Shoots but misses)
The system starts the game.
The AICar1 Shoots.
The system plays the gun fire sound.
The system sends out a Raycast which hits nothing or a wall.

Termination
In the 1 Player mode the game will end when <user (Player 1)> is last man standing or dies.
In the 2 Player mode the game will end when <user (Player 1)> or <user (Player 2)> are last man standing or if both are dead.

Post condition
The system goes into a wait state

Data Requirements
A.A.R will require data to stored. The games data will include the players top score. This will be hosted on the users local machine. This data will be very small & will not take up much space on the users machine. The user will be able to reset this data at any point they wish.

Environment Requirements

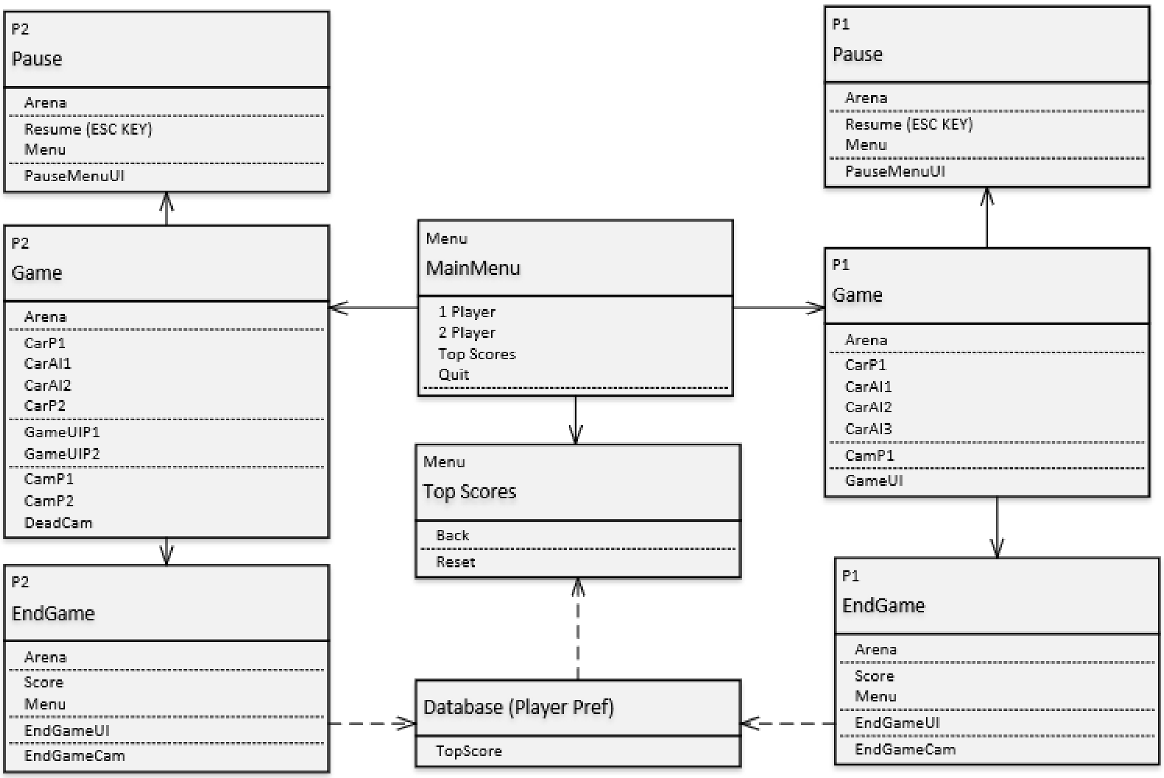
This software will be built on the game engine Unity version 2017.2. With the required programs needed to build this game there is a minimum requirements need with hardware to allow development, testing & playing of the game. The follow stating the requirement needed for: Running Unity for development of the game:

OS:	Windows 7 SP+1, 8, 10, 64-bit versions only, Mac OS X 10.9+.
CPU:	SSE2 instruction set support.
GPU:	Graphics card with DX9 (shader model 3.0) or DX11 with feature level 9.3 capabilities.

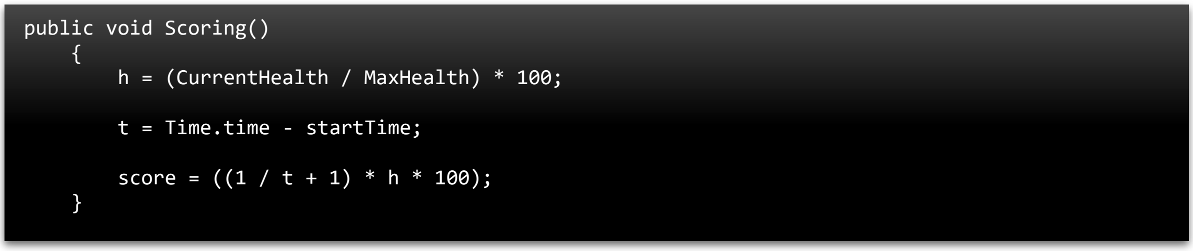
https://unity3d.com/unity/system-requirements

Design & Architecture

Architecture Diagram



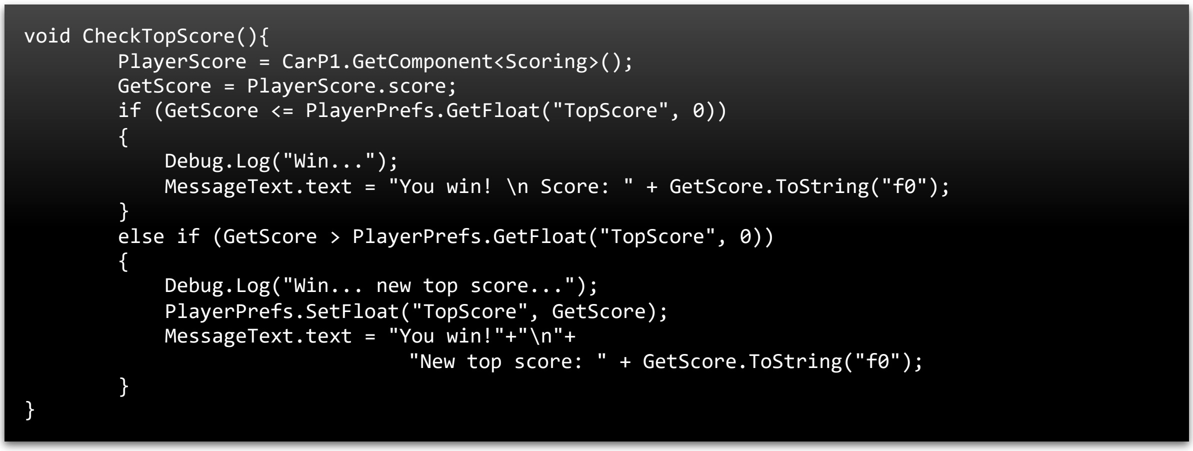
Scoring algorithms



```
public void Scoring()
    {
        h = (CurrentHealth / MaxHealth) * 100;

        t = Time.time - startTime;

        score = ((1 / t + 1) * h * 100);
    }
```

Calculating the players score is done with the equation $\left(\frac{1}{TIME+1}\right) \cdot (HEALTH) \cdot 10^2 = SCORE$. The C# code above shows this equation implemented into the project. The part of the equation $\left(\frac{1}{TIME+1}\right)$ was used so the long the game when on the lower the this will equal. If user achieved a time of 0 this would cause issues to the equation I will use TIME +1 to counteract this..

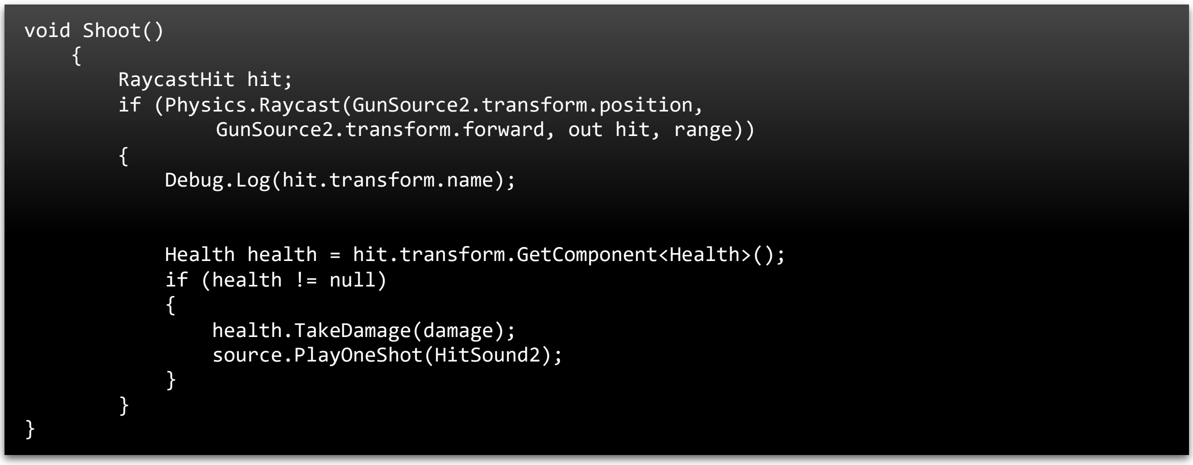
Top Scores



```
void CheckTopScore(){
        PlayerScore = CarP1.GetComponent<Scoring>();
        GetScore = PlayerScore.score;
        if (GetScore <= PlayerPrefs.GetFloat("TopScore", 0))
        {
            Debug.Log("Win...");
            MessageText.text = "You win! \n Score: " + GetScore.ToString("f0");
        }
        else if (GetScore > PlayerPrefs.GetFloat("TopScore", 0))
        {
            Debug.Log("Win... new top score...");
            PlayerPrefs.SetFloat("TopScore", GetScore);
            MessageText.text = "You win!"+"\n"+
                                 "New top score: " + GetScore.ToString("f0");
        }
}
```

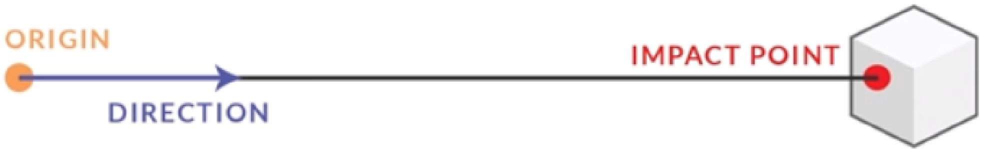
Checking & saving a new top scoring is done in the CheckTopScore() method. This method is called when a player wins a round. The method gets the score created in the Scoring class reference it as GetScore. GetScore is checked against the current stored TopScore. If the score is greater then the TopScore it is replaced & the user is notified achievement & there score. If the score is less then the rounds score is only returned to the user.
https://www.youtube.com/watch?v=vZU51tbgMXk&t=324s

Shooting



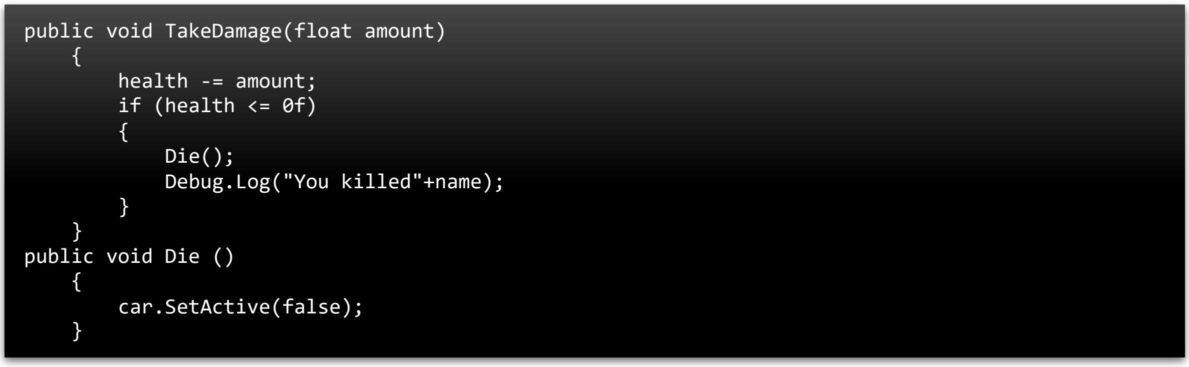
```
void Shoot()
    {
        RaycastHit hit;
        if (Physics.Raycast(GunSource2.transform.position,
                GunSource2.transform.forward, out hit, range))
        {
            Debug.Log(hit.transform.name);


            Health health = hit.transform.GetComponent<Health>();
            if (health != null)
            {
                health.TakeDamage(damage);
                source.PlayOneShot(HitSound2);
            }
        }
}
```

The user press the shoot button it calls apon the Shoot() method. This method is used for both players. Shooting is doing with use of Raycasts. When raycast is fired from the players gun (Origin) in a direction & it can detects what it impacted such as walls & players. When a raycast hits something it an requests that the objects health takes damage.
https://www.youtube.com/watch?v=THnivyG0Mvo&t=217s



https://www.youtube.com/watch?v=THnivyG0Mvo&t=217s

Damaging health & deaths

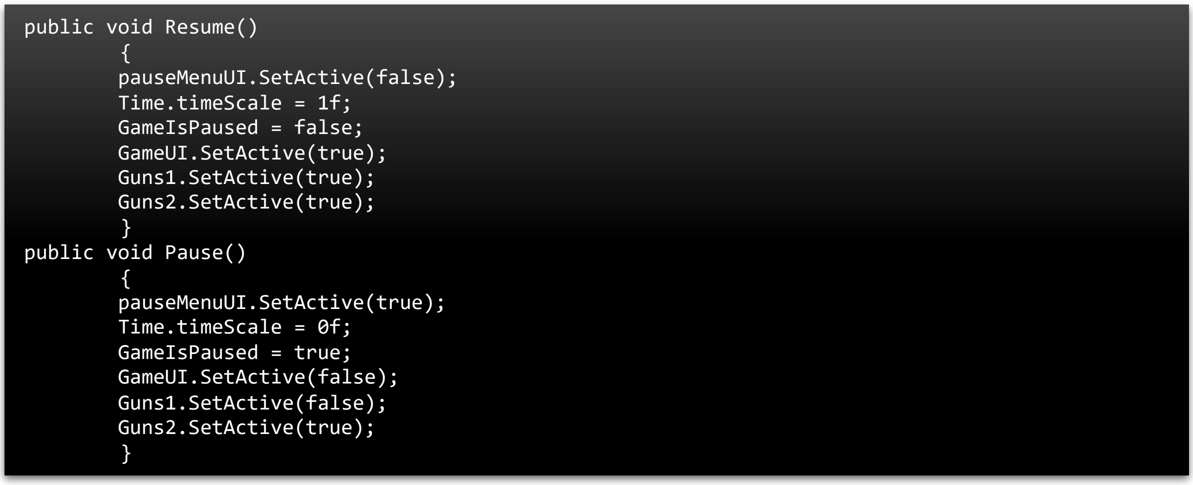


```
public void TakeDamage(float amount)
    {
        health -= amount;
        if (health <= 0f)
        {
            Die();
            Debug.Log("You killed"+name);
        }
    }
public void Die ()
    {
        car.SetActive(false);
    }
```

https://www.youtube.com/watch?v=THnivyG0Mvo&t=217s

When a player or AI car gets hit by a raycast they take damage. Damage is take away from the cars health with the TakeDamage() method. This method takes away the amount of damage from the current players health. If the player health reaches or goes bellow 0 the Dies() method runs & makes the car inactive that kills off the player.

Pausing the game

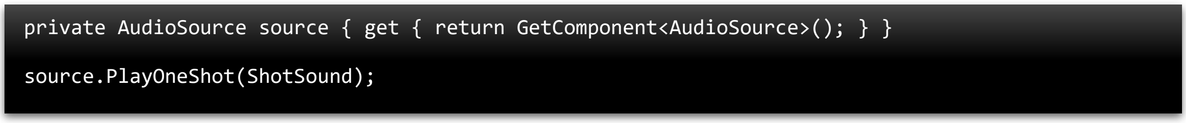


```
public void Resume()
        {
        pauseMenuUI.SetActive(false);
        Time.timeScale = 1f;
        GameIsPaused = false;
        GameUI.SetActive(true);
        Guns1.SetActive(true);
        Guns2.SetActive(true);
        }
public void Pause()
        {
        pauseMenuUI.SetActive(true);
        Time.timeScale = 0f;
        GameIsPaused = true;
        GameUI.SetActive(false);
        Guns1.SetActive(false);
        Guns2.SetActive(true);
        }
```

https://www.youtube.com/watch?v=JivuXdrIHK0

The pause menu is controlled by two methods Pause() & Resume(). The Pause() method calls the games time to be set to 0f which freezes the game, GameUI & car controls are also disabled. The Resume() method does the oppsite of the Pause() method. The method calls the games time to be set to normal speed of the game, GameUI & car controls are also enabled.

Sound



```
private AudioSource source { get { return GetComponent<AudioSource>(); } }

source.PlayOneShot(ShotSound);
```

https://www.youtube.com/watch?v=6OT43pvUyfY

Sound the A.A.R is used for effects & in game music. Sound is implemented by code the same way for sound effects & music, loops & delays can be added to the sound in the Unity inspector menu. The code above gets the sound file as with GetComponent<AudioSource>();. For example when a shot is fired source.PlayOneShot(ShotSound); is used to call & play the gun shot sound.

Non-Functional Requirements

Response time requirement
The game should have quick response time when in game as split second can be the difference between getting your shot of & missing. Latency or glitching will cause the gamer to become frustrated.

Availability requirement
Everyone is able to play A.A.R who have a supported device to run it. The game will be able to download from a website once it has been completed.

Reliability requirement
Transferring from menu to game screen in application should be quick & seamless to reduce any possible user frustrations. In game hit boxes should be tight & responsive to allow positive feedback to the user.

Maintainability requirement
The user will be able to reset there top score via the top score menu. This is the only data that the game will store.

Portability requirement
The application is planned to be playable on Windows PC.

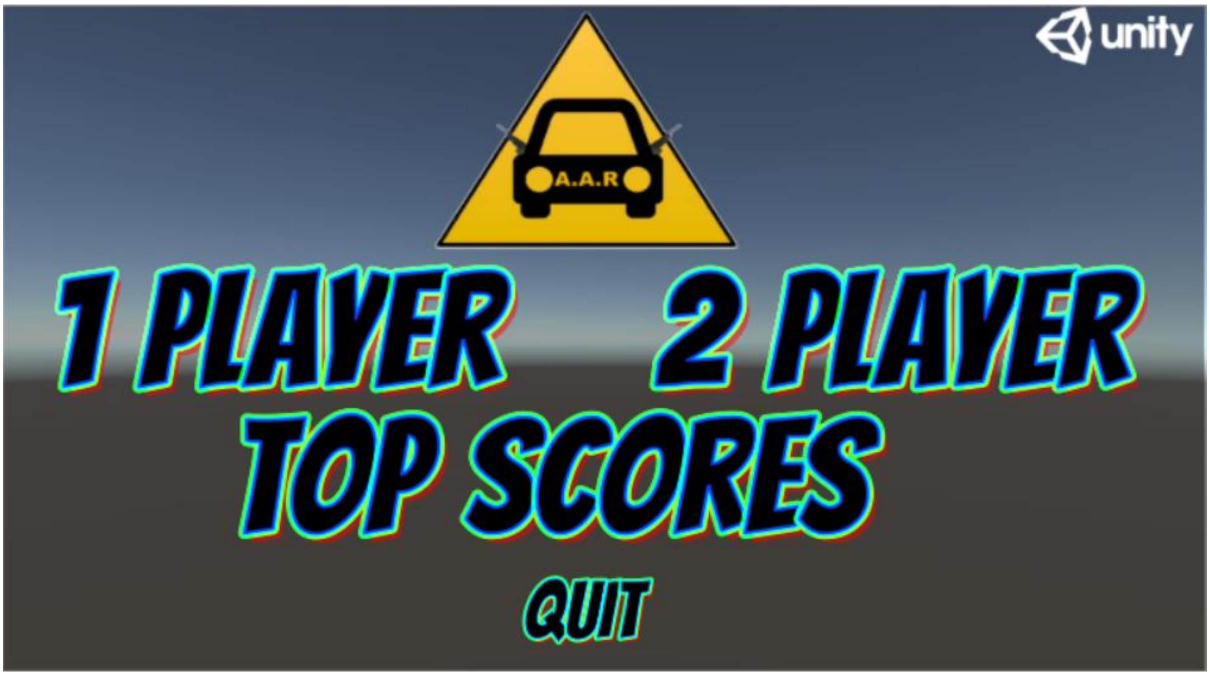
Extendibility requirement
Updates to the application such as new content can be added with future software updates.

Reusability requirement
The use of local multiplayer & top score intis gamers to come back to beat their friends & record a new top score.

Resource utilization requirement
Unity, Blender, Audacity, Visual Studios & GIMP are free to access.

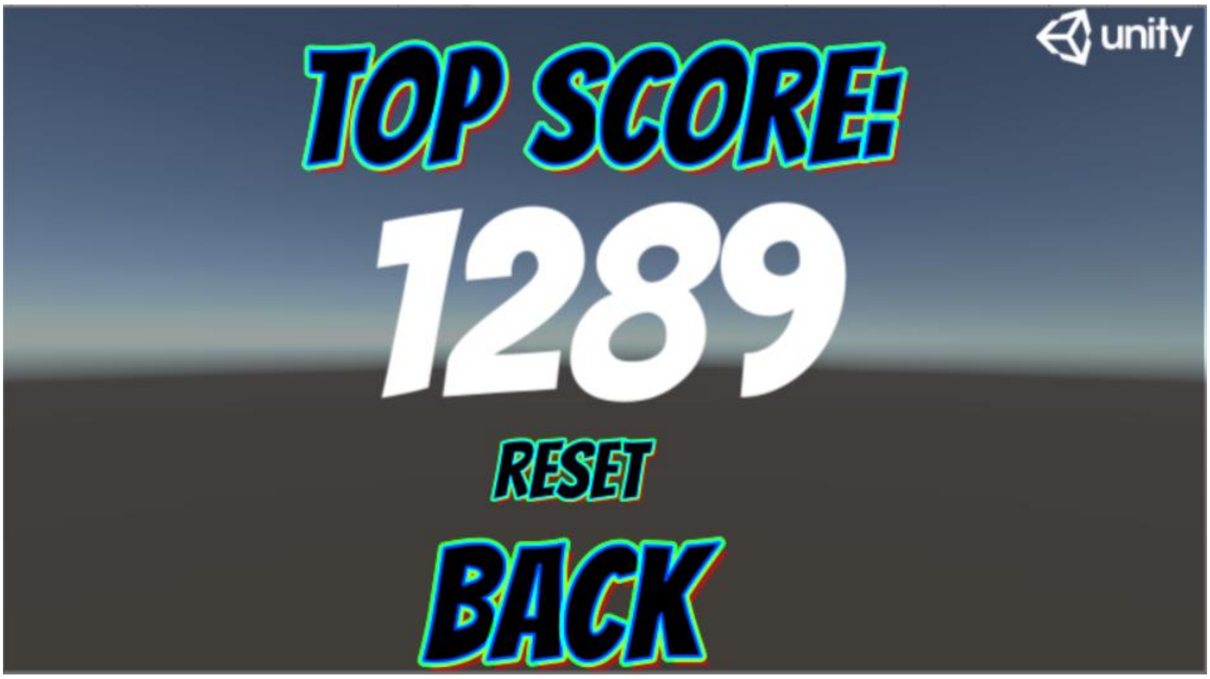
Graphical User Interface (GUI) Layout

Main Menu UI



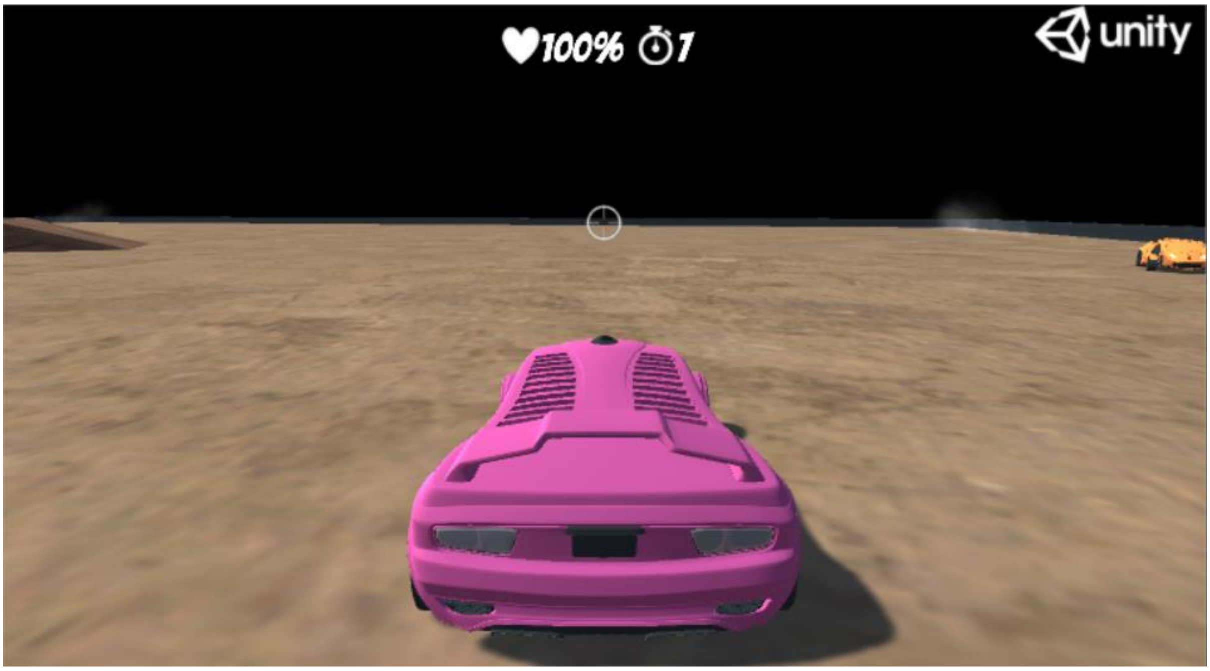
This is the main menu, this is what the player is greeted to when the start the game. The main menu will allow the user to select between 1 & 2 player modes, navigate to be able to check the top score or to quit & close the game completely.

Top Score Menu UI



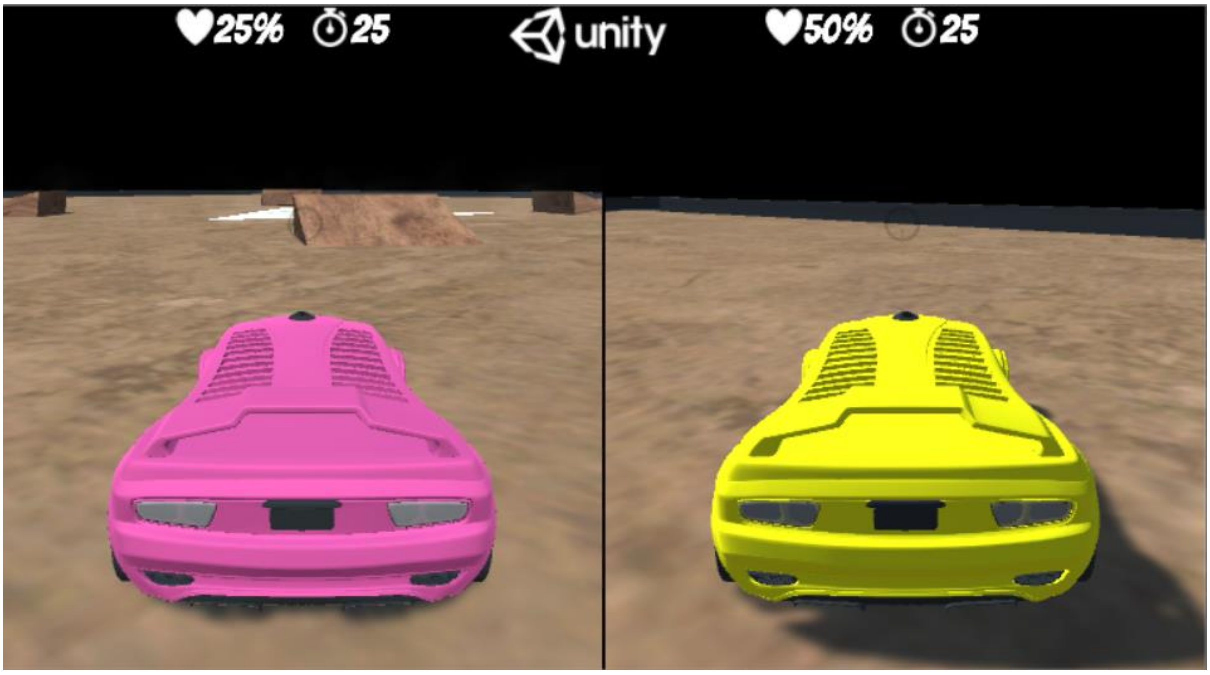
The top score page will display the players current top score. The top score can be reset back to 0 using the reset button. The user can return back to the main menu clicking the back button.

Player 1 UI



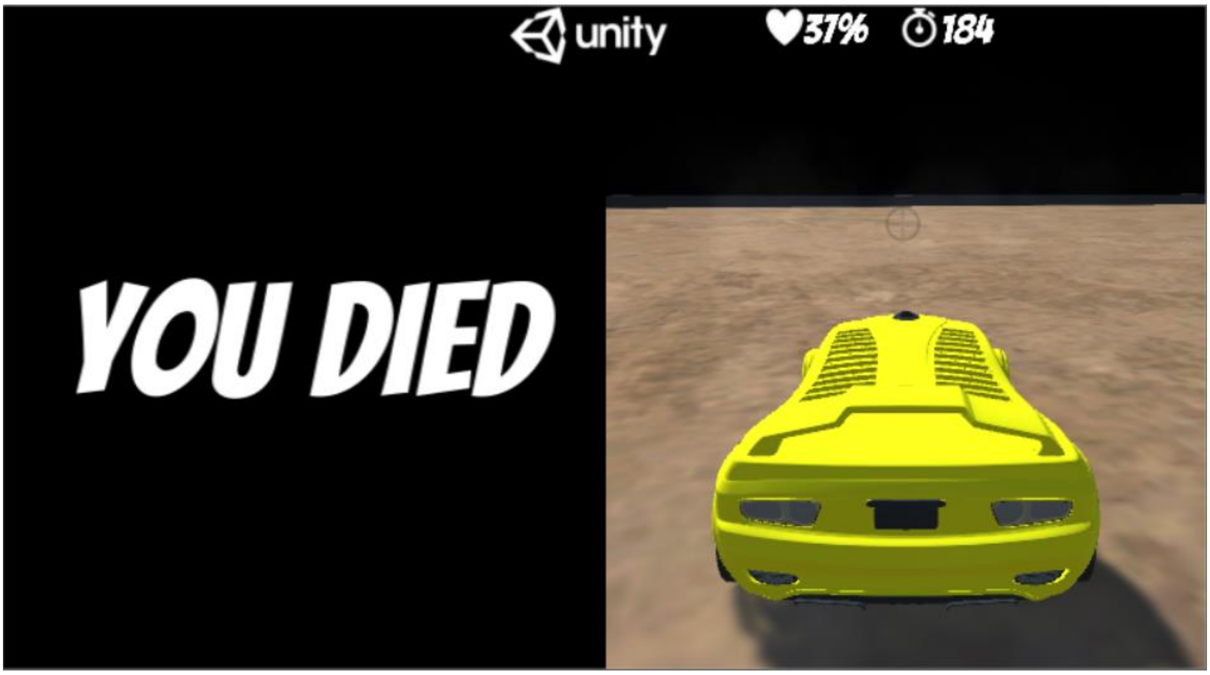
This shows screen shot displays the UI & gameplay for the 1 player mode. The UI shows the users car directly in front in a 3rd person view. The UI also shows the players health next to the hearth icon & the round running time next to the clock icon. A target spirt was added to aid the user aim when shooting.

2 Player UI



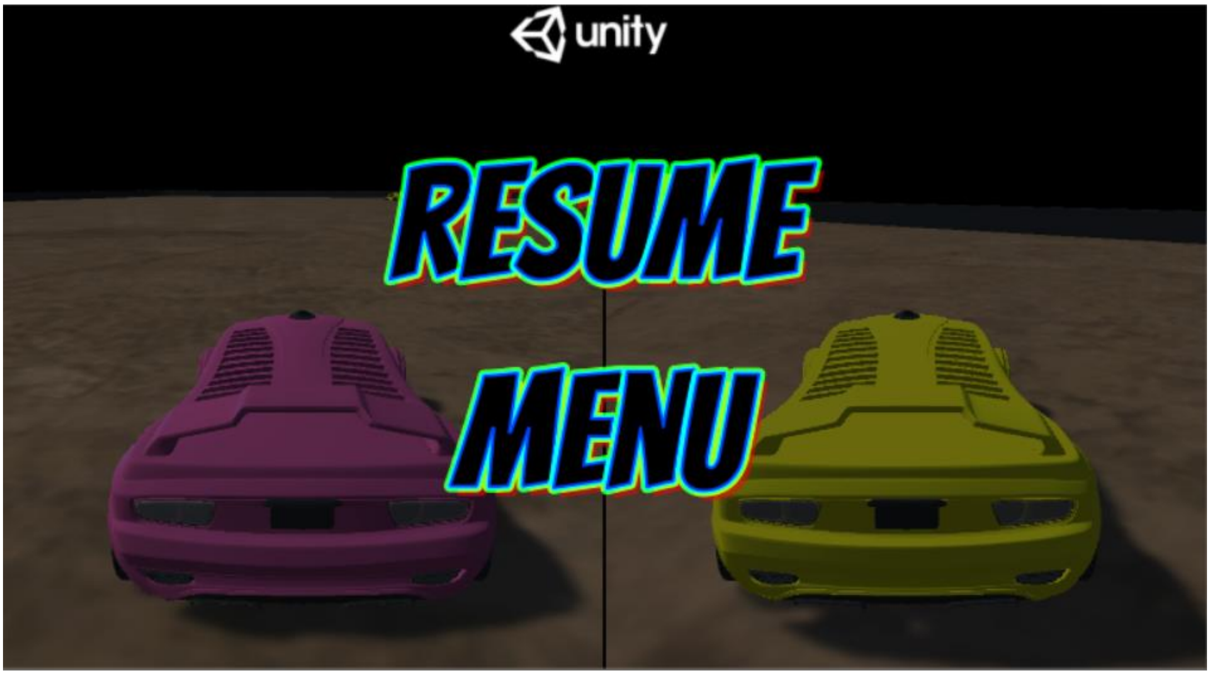
This shows screen shot displays the UI & gameplay for the 2 player mode. The UI shows the users' cars directly in front of them in a 3rd person view. The UI also shows the players health & time just like in the 1 Player mode. A target spirt was added to aid the user aim when shooting. Player 1's car is on display on the left & player 2's screen is on the right.

2 Player, when one player has died UI



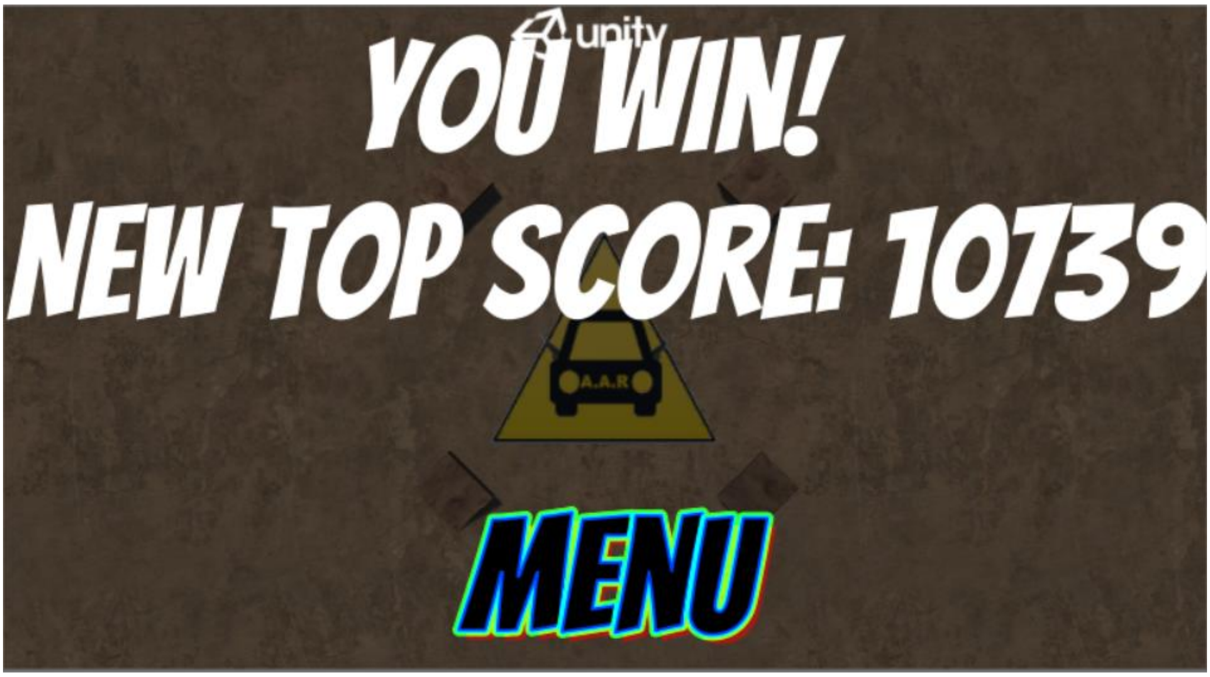
In a 2 player game when a user dies while the other player & other AI controlled cars are still alive the dead user’s screen will go black displaying the message “YOU DIED”. The player that is still alive can continue to play as their side of the screen wont be effected.

Pause Menu UI



When a player pauses the game with the ESC key the pause menu will appear. This menu will freeze the game to allow the user(s) to take a break & resume the round when ready or the they can quit the game to return to the main menu.

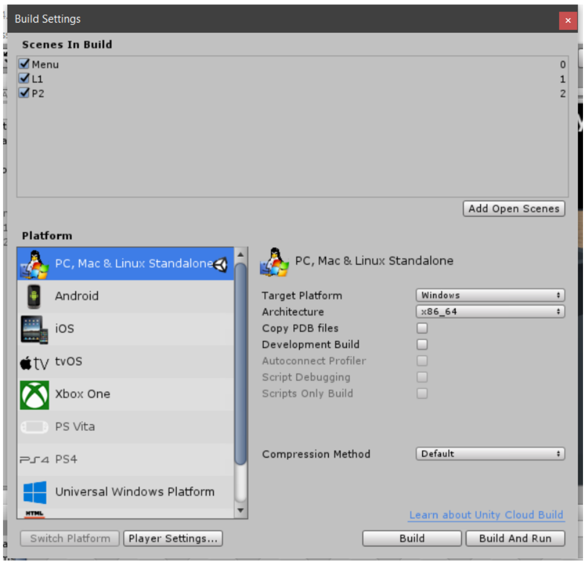
End Game UI



When a 1 or 2 Player game ends with a win or lost the end game screen will load with the result of the match. In the screen shot approve shows when a player wins a 1 player game with a top score. The user can click the menu button to return to the main menu.

Implementation

Unity is the selected game engine used to build A.A.R. Unity allows users to export the program you have create via the build settings. Unity is able to build many devices such mobile devices & games consoles but for this project I will only be porting the game to PC. The screen shot attached bellow shows how the project can be sent to built for PC.



Build Settings

Scenes In Build

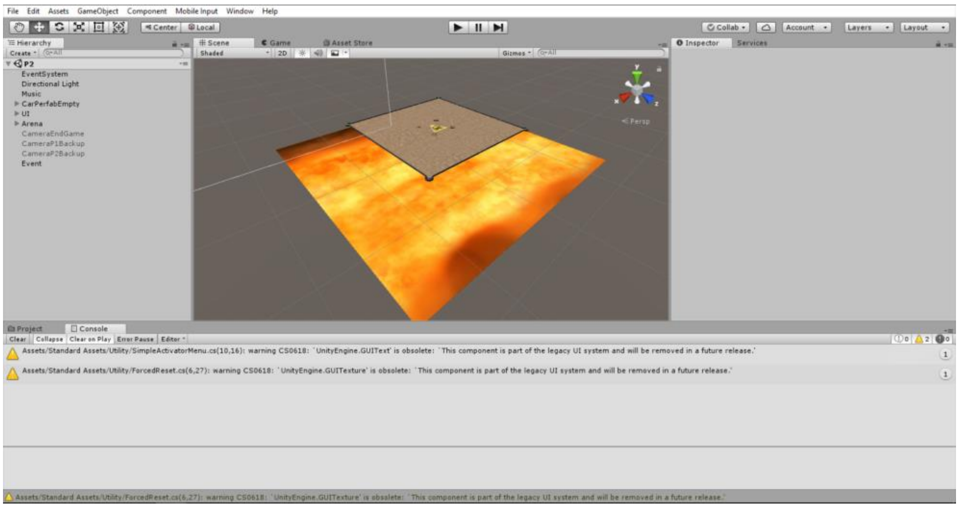
☑ Menu 0
☑ L1 1
☑ P2 2

Platform

PC, Mac & Linux Standalone

Target Platform	Windows
Architecture	x86_64
Copy PDB files	□
Development Build	□
Autoconnect Profiler	□
Script Debugging	□
Scripts Only Build	□
Compression Method	Default

Unity's UI allows users to create 2D or 3D scenes. The scenes are what makes up the menu's & the game modes. Game objects such as UI items, models, camera's & lights can be added to the scene these can compined together to create what is need for the game. These game objects can be modified by using the Unity inspector. Componds can be added into the inspector such as conliders, scripts & attaching in sound files.



```
Assets/Standard Assets/Utility/SimpleActivatorMenu.cs(10,16): warning CS0618: `UnityEngine.GUIText' is obsolete: `This component is part of the legacy UI system and will be removed in a future release.'
Assets/Standard Assets/Utility/ForcedReset.cs(6,27): warning CS0618: `UnityEngine.GUITexture' is obsolete: `This component is part of the legacy UI system and will be removed in a future release.'
```

Assests can be purchassed or download from the Unity Asset Store this has come in hand when implementing models (e.g. cars & guns), code (e.g. controllors & AI) & textures (e.g. gound & lava). Assest can also imported from other applications such as models from Blender, sound files from Audicity & images from GIMP. With the use of Blender I create an unique plateform to located in the corners. This platoform was used to be used to start the players off. GIMP was used to help create & edit UI items such as the health & time icons. Audicity was used to edit sound effects & music so they can become a better fit for the game. Sound files where optained from https://freesound.org/home/.

5: Testing

Customer Testing

Customer testing was done with use of surveys to be given to the testers after they finish testing. The testers used are members of the NCI Gaming & eSport Societies.

Link to customer testing survey https://goo.gl/forms/Wgez8E1omU9HSU0y1. (Survey results attracted under "Customer Testing" in Appendix)

As can be seen from the surveys 80% would recommend this game to a friend or colleague. This is great as it will get other people notified about the game by word of mouth & would like to play it. 60% of the testers provided a positive response to the game while 40% where unsure. This may have been in result to the lack of content available to the players. The feedback in terms of the quality of the game returned was very mixed as 60% where unsure, 20% thought the game was low quality & 20% felt the game was high quality. This may have been due to the some of the ongoing bugs in the game & low graphic quality as many gamers are used to HD standard graphics. The lack of other game modes & replicability to the game may have been a factor in why 40% found it unlikely & 20% feeling unsure about coming back to play the game again. This could be fixed by implementing more game modes such as a deathmatch or capture the flag. In regards to the comments left about the game users responded very well towards the layout & controls of the game. The use of a common input of method of the WASD keys as in many other main stream games allowed user transition to the game very easily. This is great as it allows for great platform to build on.

This survey was very helpful as it allows to grasp an understanding what parts of the game works well & where the game needs to be improved apron into the future.

Usability Testing

For a usability testing a think aloud test was done with three people in order to test my game & players ability on what options they would choose to progress. Link to usability testing survey https://goo.gl/forms/hMtKMnyxoifhBFmJ2 (Survey results attracted under "Usability Testing" in Appendix) & to the think aloud video <link to YouTube video>.

Person 1

This person is the one who's think aloud was recorded & out of the three people. This user users task was to start a 1 player game. This user took the XXX seconds to complete the testing of the game. The user was very exited to try out a new game. Within the first few moments she was impressed how quick & easy it was to start a game. When the tester started the 1 player mode & was very well able to use the games controls. While playing she stated that the in game music was very upbeat, this was very much intended as it may intense players to play an fast & attack minded game. The continued to go on to win the round by killing of the other AI controlled cars. The user seemed to have enjoyed their time playing the game. The results from the SUS given to this person was XX out of 100.

Person 2

This person test was linked in with person 3 as they played a 2 player game. Person 2 played as player 1. This user took 60 seconds to complete the testing of the game. The user was very eager to play & beat there friend at the game. The person started the game & selected 2 player game. This users screen was located on the left. The user was quickly able to grasp the control layout. The player sadly did not last long as they were killed off by one of the AI controlled cars. When the player died & was greeted the “YOU DIED” screen, knew they had just sit back & watch how person 3 (Player 2) finished off. The results from the SUS given to this person was XX out of 100.

Person 3

This person test was linked in with person 2 as they played a 2 player game. Person 3 played as player 2. This user took 90 seconds to complete the testing of the game. The user was very focused as he wanted beat his opponent at the game. The person 2 started the game & selected 2 player game. This users screen was located on the right. Was a little bit put off for a moment as they were using the number pad as controls but quickly adapted. As for the issue for the users control input the user took a little bit of damage when taking on their first AI controlled car that approached him but was able to turn it around & defeat it. The player followed up killing off the AI controlled car that person 2 (player 1) was defeated by & winning the round with a top score. Some joking between person 2 & 3 started as of result of the game, this great to see as this interaction between the players would encourage them to have another game. The results from the SUS given to this person was XX out of 100.

Conclusion

With all the analysing while doing the think aloud it shows that the testers enjoyed the playing the game. They were amazed with how easy it was to pick up the game as the controls where easy to understand, use & where layout very well. With the test there was some technical bugs that will need fixing such as menu button sound effects, find such errors in the think aloud will provide me resources to quality assure the game. Out of the 2 rounds player both 2 out of the 3 player go wins & the other player was killed by an AI controlled car, this shows that the game is bet able but can also provide challenge. During the testing I notice players found somewhat hard to quickly find & locate the AI cars nearer the end of the round, this may need to be looked into as searching for enemas provides a challenge in itself but implementing mini map will allow user know where they need to go quicker. The overall score from the SUS combined was XX out of 100 which is a good number at as it proves the game is good but will need some improvements & polishing to improve the quality of the game. Finally, through all of this analysis & from the testing it is clear that the game is quite appealing & will gain some traction.

https://usabilitygeek.com/how-to-use-the-system-usability-scale-sus-to-evaluate-the-usability-of-your-website/

Evaluation

Over time the game has been evaluated by the use of testers & playtests done. Once a new car/game mode/arena have been completed it would then be evaluated with playtests to make sure that everything is running as it should be. The game still works find on current low end PC’s/laptops with as Unity provide a configuration menu when starting the .exe file to change the games screen resolution, graphics & if the game should run full screen or in a window.

6: Conclusions

From the researching & creation of this document you can see that this game has surely evolved over time. From all this work it has been beneficial as it has helped in the discovery/learning of game development. With the testing it is clear that this game will be attractive towards the target audience, as feedback has been positive across the board. Despite all the positive feedback about the project there are is a lack of different game modes (such as deathmatch & capture the flag) & content (such as more choice of cars & arenas). Keeping in mind the positive response it is clear that there will be opportunities with the game may it be in regards to expansion as people love find the gameplay easy to use & the current game mode of a last man standing is very entertaining in both 1 & 2 player games. Finally, it ca be concluded that all of the document combined have both laid the ground work & the growth of the project to continue forward.

7: Further Development

As A.A.R stands as a game it is very mentalistic in terms of a game. Saying this game has a strong base to build apron into something bigger with additional resources such as a development team to work on different aspects game.

The game could incorporate a more game modes such deathmatches & capture the flag. Introduction of these game modes would increase the replicability of the game. More content may get added to the game such as different vehicles & arenas. The games multiplayer function can also develop with extra local players (3 or 4 player game) & have matches hosted online to allow players play against others around the world. These extra features & content can be added to the game via free or paid DLC which in turn can create additional revenue for the game.

8: References

- Official PlayStation™Store Ireland | Home of PlayStation games, PS4, PS3, PSVita . 2017. *Official PlayStation™Store Ireland | Home of PlayStation games, PS4, PS3, PSVita* . [ONLINE] Available at: https://store.playstation.com/en-ie/home/games#!/en-nz/games/ctr-crash-team-racing/cid=EP9000-NPEE00026_00-GCRASHTEAM000001. [Accessed 29 November 2017].
- Mike Fahey. 2017. Mario Kart 8 Switch Makes Battle Mode So Much Better. [ONLINE] Available at: https://kotaku.com/mario-kart-8-switch-makes-battle-mode-so-much-better-1794448420. [Accessed 29 November 2017].
- VentureBeat. 2017. John Riccitiello sets out to identify the engine of growth for Unity Technologies (interview) | GamesBeat. [ONLINE] Available at: https://venturebeat.com/2014/10/23/john-riccitiello-sets-out-to-identify-the-engine-of-growth-for-unity-technologies-interview/. [Accessed 29 November 2017].
- GIMP. 2017. GIMP - GNU Image Manipulation Program. [ONLINE] Available at: https://www.gimp.org/. [Accessed 29 November 2017].
- Unity. 2017. Unity - System Requirements. [ONLINE] Available at: https://unity3d.com/unity/system-requirements. [Accessed 29 November 2017].
- Audacity® | Free, open source, cross-platform audio software for multi-track recording and editing.. 2018. Audacity® | Free, open source, cross-platform audio software for multi-track recording and editing.. [ONLINE] Available at: https://www.audacityteam.org/. [Accessed 02 May 2018].
- Blender Foundation. 2018. blender.org - Home of the Blender project - Free and Open 3D Creation Software. [ONLINE] Available at: https://www.blender.org/. [Accessed 02 May 2018].
- Visual Studio. 2018. Visual Studio IDE, Code Editor, VSTS, & App Center. [ONLINE] Available at: https://www.visualstudio.com. [Accessed 02 May 2018].
- YouTube. 2018. How to make a HIGH SCORE in Unity - YouTube. [ONLINE] Available at: https://www.youtube.com/watch?v=vZU51tbgMXk&t=324s. [Accessed 03 May 2018].
- YouTube. 2018. Shooting with Raycasts - Unity Tutorial - YouTube. [ONLINE] Available at: https://www.youtube.com/watch?v=THnivyG0Mvo&t=217s. [Accessed 03 May 2018].
- YouTube. 2018. PAUSE MENU in Unity - YouTube. [ONLINE] Available at: https://www.youtube.com/watch?v=JivuXdrIHK0. [Accessed 03 May 2018].
- YouTube. 2018. Introduction to AUDIO in Unity - YouTube. [ONLINE] Available at: https://www.youtube.com/watch?v=6OT43pvUyfY. [Accessed 03 May 2018].
- Freesound - login. 2018. Freesound - login. [ONLINE] Available at: https://freesound.org/home/. [Accessed 03 May 2018].
- usabilitygeek.com. 2018. No page title. [ONLINE] Available at: https://usabilitygeek.com/how-to-use-the-system-usability-scale-sus-to-evaluate-the-usability-of-your-website/. [Accessed 05 May 2018].

9: Appendix

Project Proposal

Introduction
Background

Arena Assault Racing (A.A.R for short) will be a game that allows the user to battle in weaponized vehicles in game modes such as survival & multiplayer. The game will have a wide range of different enemy's. The graphics will be top down view with retro 8-bit 2D graphics. The game will be published to android mobile devices & as well on Windows PC.

The user will fight against human & AI controlled bots to get to the top of the leather board. The user will be able to gain points on how well they do in each level to allow them by new cars & items to improve its performance.

Drive Alive will fit into the driving, action/shooter & survival genres. The driving genre is a massive & very diverse within the gaming. The different types of driving/racing games come in the likes of simulators such as Forza Motorsport & Grand Turismo series & causal racers such as Need for Speed & Mario Kart. With the large selection of drive games on the market this it shows that it is one of the most popular genres. Action/shooting games are very popular from adventure style with the likes of the GTA & Saints Row series & war themed shooters likes of Call of Duty & Battlefield series. These games are very popular as GTA & Call of Duty games be the most sold & highly rated games when they its released.

Objectives
There are several objectives involved in creating this project, these include:

- Create a simple & easy to play gameplay system that will allow any one at any age, level or gaming experience to be able to start playing game.
- Create a simple to UI design to help the user navigate through the game's menus.
- Create & design several different arenas' that will be diverse with their own layouts & challenges to master.
- Artwork & animations for the user's cars, enemies, props, weapons & arenas.
- Create & implement music & sound effects into the game to help make the game more immersive.
- Create the different AI strategies for the games different enemies.
- Create an achievement & unlockable system that will award the user for passing levels & for playing the game in different ways.
- Get the game work on both android phones & tablets.
- Build a local & server based database to allow users scores & progress to be stored.
- With the use of cloud services from Amazon & Google to store the games data & host the application to allow online multiplayer.
- Test a games prototype with a number hard-core gamers & people who would be technology literate.

Target Group
As I plan to publish the game to Android & Google Play Store, I would be targeting people who play games on their phones & tablets. This game will may be suitable for users & will be a family friendly game to play. The style of the game that allows users drive, shoot enemies & other players will make this fun to play with friends.

Technical Approach

Research

- Research into other similar games such as Rocket League, Crash Team Racing & Mario Kart on how these games run & are played. Playing & researching test games will help give me ideas to implement to my game as well as how to improve this.
- Once the game reaches a viable & playable product, I would use my position & contacts within the NCI Gaming Soc to get people to try play & test the gameplay & give me feedback to help improve the created level & future levels to be made.

Literature review

My approach for research is:

- Play & study how the best drive & shoot style games feel to play & how they work.
- I have I also planned to research into driving & attacking AI's strategies
- Consider game theories to help make my game interesting & intriguing to keep playing.

Special Resources Required

This game will require the studying & practice of many different software applications. These applications will also need a computer which need to meet the specified minimum hardware requirements to run these applications. With the required software & hardware to develop the game, there is also required hardware & software for testing, debugging & deployment. The plan of the game been hosted on the Google Play Store & following list includes an early list of:

- Unity 5:
  - Operating System: Windows 7 SP1+ or newer.
  - CPU: SSE2 instruction set support.
  - GPU: graphics card with DX9 or DX11 with feature level 9.3 capabilities.

  (The rest mostly depends on the complexity of the project.)

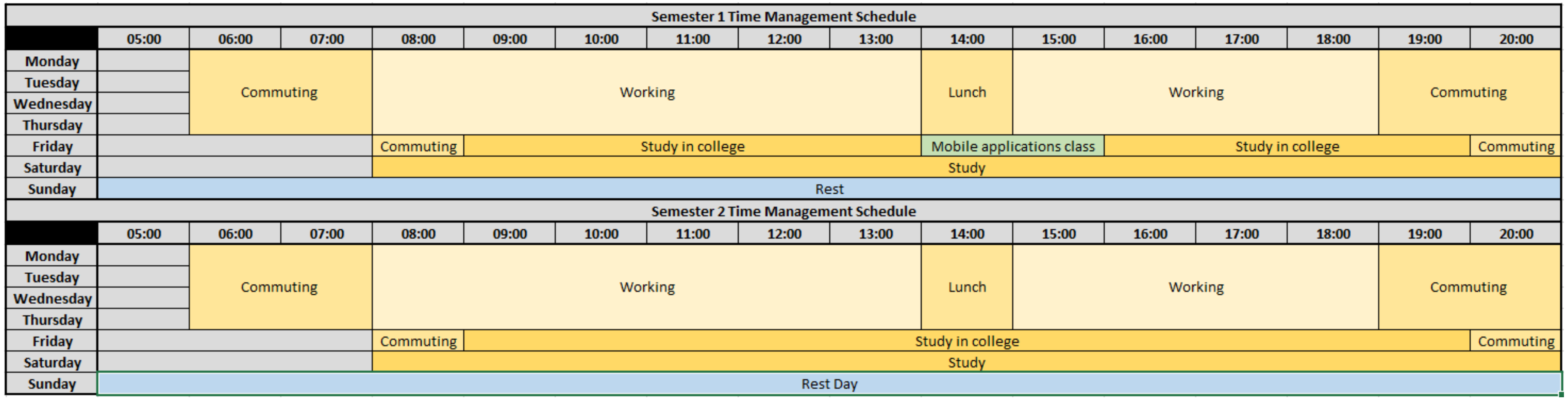
- Android Support:
  - Android SDK & JDK
  - Tablet or Phone
  - Android OS 2.2 or above.
  - Tablet or mobile device
  - Android 4.1 or newer.
  - ARMv7 CPU with NEON Support or Atom CPU.
  - OpenGL ES 2.0 or later.

- Universal Windows Platform (UWP) Support:
  - Windows 8.1 (64-bit)
  - Visual Studio 2015
  - Windows 10 SDK
- Other software & hardware needed:
  - Photo editing software.
  - Sound editing software.
  - Game pad.

Evaluation

Once the game is complete I plan on hosting the game online & the Googles Play store. I would also use my position & contacts in the NCI Gaming Soc & Irish Game Dev Community to get users to play & test the gameplay & give me feedback to help improve the games features.

Project Time Management Schedule



Semester 1 Time Management Schedule

	05:00	06:00	07:00	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00
Monday		Commuting		Working						Lunch	Working				Commuting	
Tuesday																
Wednesday																
Thursday																
Friday				Commuting	Study in college					Mobile applications class		Study in college				Commuting
Saturday				Study												
Sunday	Rest															

Semester 2 Time Management Schedule

	05:00	06:00	07:00	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00
Monday		Commuting		Working						Lunch	Working				Commuting	
Tuesday																
Wednesday																
Thursday																
Friday				Commuting	Study in college											Commuting
Saturday				Study												
Sunday	Rest Day															

Monthly Journals

Student name: Stephen Lowry | x12356291
Programme: BSc in Computing
Month: September

My Achievements

- Contacted Eamon Nolan to ask if I can get Anu Sahin as my supervisor for my project.
- I brainstormed possible projects I could build.
- I decided to go with an automobile style survival game.
- I have started research on what software can be used to build this game.
- Have narrowed down the game engine to build my game on will be Unity or Game Maker Studios.
- I have considered what platforms this game can be published to such as windows, mobile or a HTML web application.
- I have started the project proposal

My Reflection

- I have been given Anu Sahin as my project supervisor which I am very happy with. I plan to create a meeting with her soon.
- Brainstorming of types of projects, games, platforms & even technologies to be used was very helpful.
- Started & create a brief of the project proposal.
- The project/game needs a name.

Intended Changes

- I will have to meet up with my supervisor to help discuss problems I may face in my project.
- I need to create a weekly planner to help balance my full-time job at HP Enterprise, this projects tasks & other subjects.
- Finish project proposal & submit it for week 6.
- Start & create a brief for the software requirements specification of the project.
- Research differentness, pro & cons of Unity & Game Maker.
- Build & test basic functionality of the game in Unity & Game Maker.

Supervisor Meetings

Date of Meeting: N/A
Items discussed: N/A
Action Items: N/A

Student name: Stephen Lowry | x12356291
Programme: BSc in Computing
Month: October

My Achievements

- Got in contact with Anu Sahin to set up meetings every second week.
- I brainstormed what I could build into my game.
- Have narrowed down the game engine to build my game on will be Unity.
- I have looked more into what platforms this game should be published on such as a windows application, mobile or as a web application.
- I have completed & uploaded the project proposal.
- I have started the software requirements.
- Build basic gameplay into Unity & Game Maker Studios.

My Reflection

- I have completed a draft version of my project proposal, but I will need to revisit this to add a project plan into it.
- Brainstorming of types of game modes & content.
- Finished off & uploaded proposal.
- Started & created software requirements specifications.
- The game still needs a name.
- I feel Unity will be best software to use as it will allow me to implement more features & look more impressive using an industry standard software.

Intended Changes

- I will have to meet up with my supervisor to help discuss problems I may face in my project.
- I need to improve my weekly planner to help balance my full-time job, this projects tasks & other subjects.
- Finish software requirements specification & submit it for week 10.
- Improve upon the games graphics & art work, such as its UI, sprites & arena.

Supervisor Meetings

Date of Meeting: N/A
Items discussed: N/A
Action Items: N/A

Student name: Stephen Lowry

Program BSc in Computing

Month: November

My Achievements

- Found name for the project, Arena Assault Racing.
- Started my prototype for A.A.R.
- Created some art concepts for the game such as sprits, arenas & UI designs.
- Uploaded a draft version the software requirements specifications on 24/11.
- Created & completed the necessary heading in the tech report. Uploaded to Moodle on 30/11.
- Edited & completed the software requirements for the mid-point tech report.
- Touched up my project proposal & inserted it into the tech report.
- Researched Google Cloud Platform.
- Started software prototype.
- Create controlled movement functionality with the player's sprite & in simple environment.

My Reflection

- I am pleased to have created a name for my project. This has made me keener towards working my project now.
- Starting to gather large amount of art work for the project. Some may not feature in the game, but it is helpful to get ideas on to papers for future judgement.
- Majority of the documentation done has been completed. This helped outline my goals & the challenges of the project in hand.
- After considering cloud services this has given me the opportunity to attempt to implement online multiplayer mode but needs more researching.
- The software side of the development for Arena Assault Racing has been has finally broke ground. Hope to start user/consumer testing once I have playable game.

Intended Changes

- Create slides for the mid-point presentation.
- Implement basic functional enemies & shooting into the prototype.
- Create the UI structure & design.
- Get practice with my presentation with a mock with a supervisor.
- Purchase & research more into Google Cloud Platform.

Supervisor Meetings

Date of Meeting: 17/11
Items discussed:

- What the game was about?
- Status of what work has been done.
- What I am currently working on?
- What needs to be done?

Action Items:

- Complete requirements specifications for next week to be checked before upload.
- Build basic game with simplistic functionality such as the car moving & shooting in an environment.

Date of Meeting: 29/11, (With Dominic Car)
Items discussed:

- How many requirements needed for the requirement specification?
- What titles are needed to be filed out for mid-point tech report?
- Asked what the set up for the midpoint presentation is with my Anu unable to attend.
- What should be featured in my presentation?
- Set up a mock presentation before Tuesdays (5/12).

Action Items:

- Complete the necessary heading for the mid-point tech report & upload it to Moodle.
- Create a presentation with the key points about my project.

Student name: Stephen Lowry

Program BSc in Computing

Month: January

My Achievements

- Changed my game from top-down 2D to a 3D style game as this felt easier to build & allow me to implement more arena features such as ramps.
- Build an arena level to help build the game on.

My Reflection

- The change from 2D to 3D style of game was a massive change for the project, but I feel that this will help create a better looking style of game.
- I have completed the first arena which includes 4 ramps & a drop off at the edge of the map. These features have been easier to implement with the change of the games dimensions.

Intended Changes

- Implement user & AI models.
- Implement UI menus to allow user to navigate & start the game.

Supervisor Meetings

Date of Meeting: N/A
Items discussed: N/A
Action Items: N/A

Student name: Stephen Lowry

Program BSc in Computing

Month: February

My Achievements

- Created a controlling car for the user. No weapons have enabled, only driving controls.
- Implemented UI menus to allow the user to simply navigate & start the game.
- Started looking into Unity Assets that allow for a simpler solution to implementing AI into the game.

My Reflection

- Implemented the driving controls of the car into the game but need to look into implanting gun & firing controls to allow the user attack.
- The game UI was implemented with the help of the Unity asset Text Mesh Pro. This made it possible to have a unique style of text for my menu's.

Intended Changes

- Implement AI controls for the games enemies.
- Implement gun & attacking controls.
- Implement player 2 controls.

Supervisor Meetings

Date of Meeting: N/A
Items discussed: N/A
Action Items: N/A

Student name: Stephen Lowry

Program BSc in Computing

Month: March

My Achievements

- Implemented the two player controls car
- Implement AI controls for the games enemies to approach the user(s) but unable are unable to attack.

My Reflection

- No weapons have enabled, only driving controls still.
- Implemented the driving controls for 2 players into the game.
- Implemented in game sounds & music to the game.

Intended Changes

- Implement the guns firing controls.
- Create scoring system & end game leader board.
- In game pickups.
- Complete tech report.
- Create poster.

Supervisor Meetings

Date of Meeting: N/A
Items discussed: N/A
Action Items: N/A

Student name: Stephen Lowry

Program BSc in Computing

Month: April

My Achievements

- The user is now able to fire their gun in 1 & 2 player mode.
- Added to the AI to allow them to shoot.
- Fixed up the tech report from the mid-point.
- Started & completed poster.

My Reflection

- AI is found to be very aggressive but can be easily out smarted or exploited.
- User is unable to finish game once died or when has kill the other players.
- No scoring system has been impended.
- Removed the idea of game pick up, this could be added at the end if time permits it.

Intended Changes

- Add an end game for when the player wins or loses.
- Add a leader board.

Supervisor Meetings

Date of Meeting: 15/4/18
Items discussed:

- The build & progression of the game
- Showed off menu's.
- Showed gameplay for Player 1 & 2.

Action Items:

- Create a leader board
- Create a scoring system
- Create an end game.

Student name: Stephen Lowry

Program BSc in Computing

Month: May

My Achievements

- Found a way to keep track of what players are dead or alive.
- This as aloud me to determean if the round has ended in a defeat or win for the player(s).
- Created a scoring system with the use of the players health & the length of time the round lasted.
- Created & implemented a top scorer system.
- Completed the tech report.

My Reflection

- There appears to be a couple of minor bugs in the game such as sound effects & buttons not working correctly.
- A top score works very well for the game.
- With the software side of the project completed I feel what the game I have created can provide great entertainment & extra content could be added to it in the future.

Intended Changes

- Submit project for the 11/05/2018

Supervisor Meetings

Date of Meeting: 1/5/2018.
Items discussed:

- Final edit for the A.A.R game.
- Showed off menu's.
- Showed gameplay for Player 1 & Player 2.
- Shows off the end game menu for when the player loses or wins a round.
- Displayed games ability to record & save a top score.

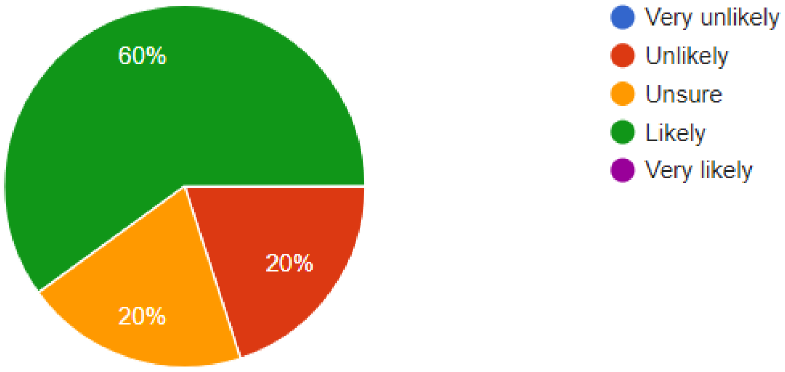
Action Items:

- Turn all attention to the tech report & return to the game if time permits it.

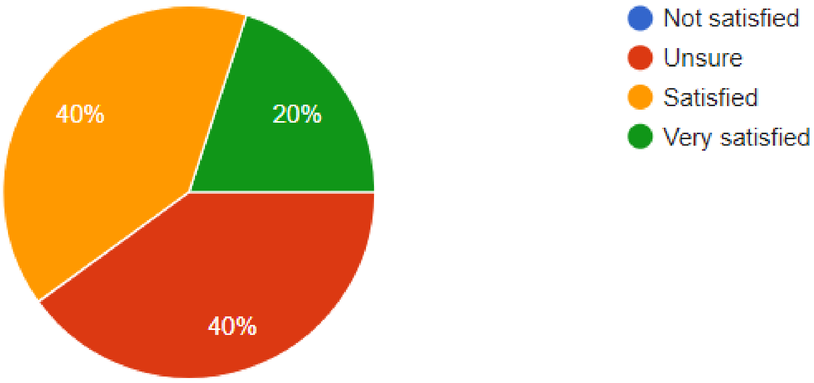
Testing & Survey results

Customer Testing Survey results

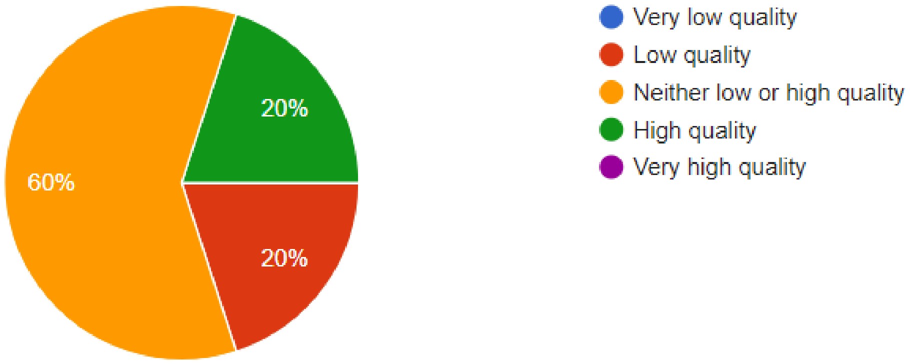
How likely is it that you would recommend game to a friend or colleague?



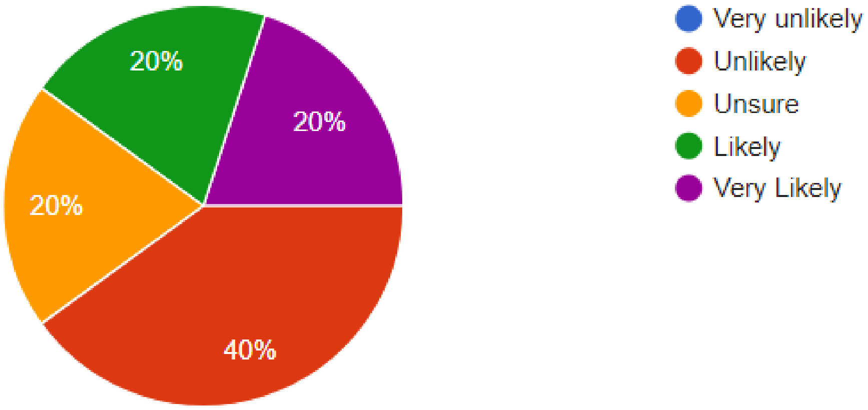
Overall, how satisfied or dissatisfied are you with the game?



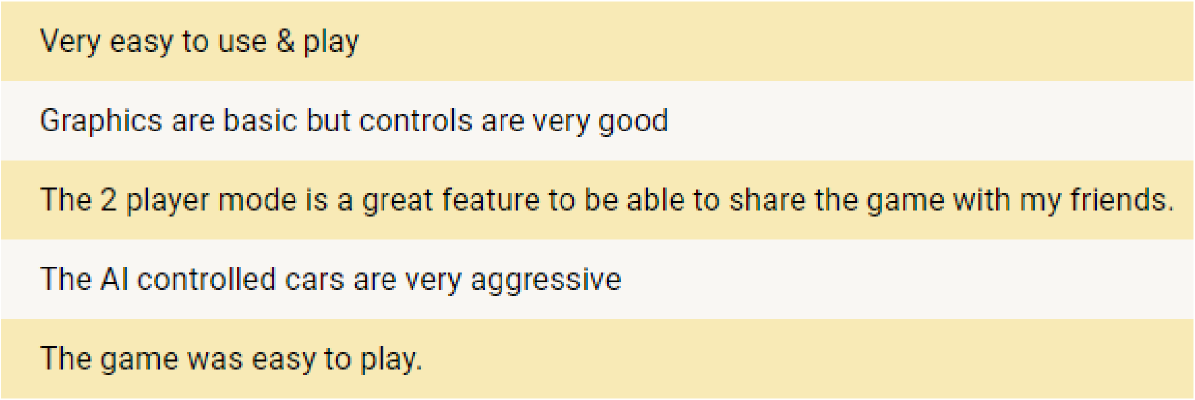
How would you rate the quality of the game?



How likely would you play it again?



Do you have any comments regarding the game?



Very easy to use & play

Graphics are basic but controls are very good

The 2 player mode is a great feature to be able to share the game with my friends.

The AI controlled cars are very aggressive

The game was easy to play.

Think Aloud Testing Task Sheet (Person 1)

Hello, my name is Stephen and I'm going to be walking you through this session today. But before we begin I will go over the reason why you are here, what you will be doing today in this session is that you will be playing my game which has been set up for you. A point I do want to make clear is that this is a test of the game and not you. During this test of the game I would like you to speak out loud in regarding what you are currently are and going to do, as well as your opinion. Feel free to speak your mind regarding your opinion.

These are the tasks that you shall be doing:

1. Load up the game from the available exe file.
2. Start a 1 player game.
3. Play a round in the 1 player mode.

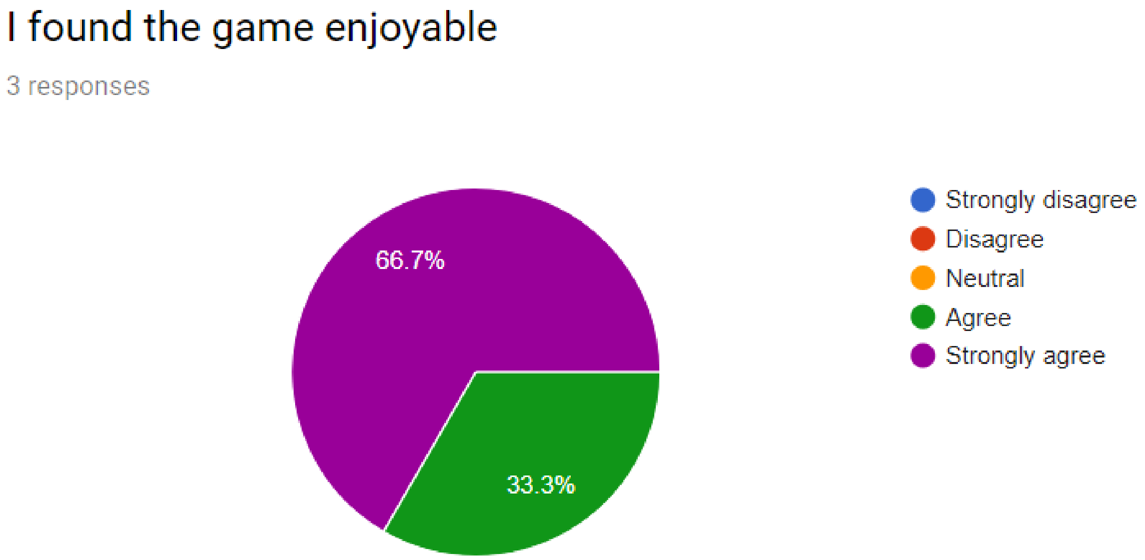
Think Aloud Testing Task Sheet (Person 2 & 3)

Hello, my name is Stephen and I'm going to be walking you through this session today. But before we begin I will go over the reason why you are here, what you will be doing today in this session is that you will be playing my game which has been set up for you. A point I do want to make clear is that this is a test of the game and not you. During this test of the game I would like you to speak out loud in regarding what you are currently are and going to do, as well as your opinion. Feel free to speak your mind regarding your opinion.

These are the tasks that you shall be doing:

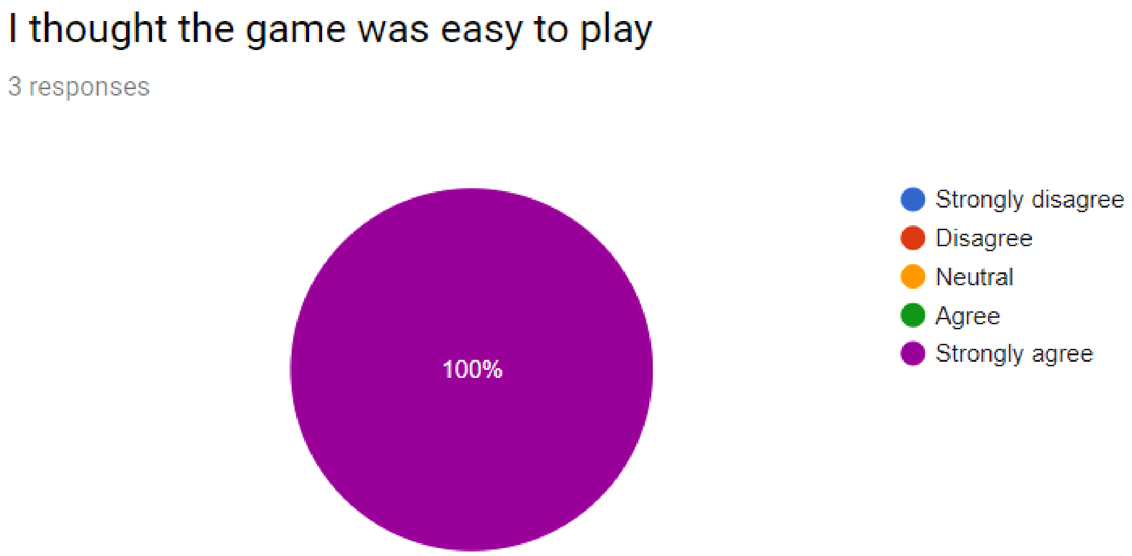
1. Player 1 loads up the game from the available exe file.
2. Player 1 starts a 2 player game.
3. Player 2 can now join in.
4. Play out the round against each other in the 2 player mode.

Usability Testing SUS results



I found the game enjoyable

3 responses

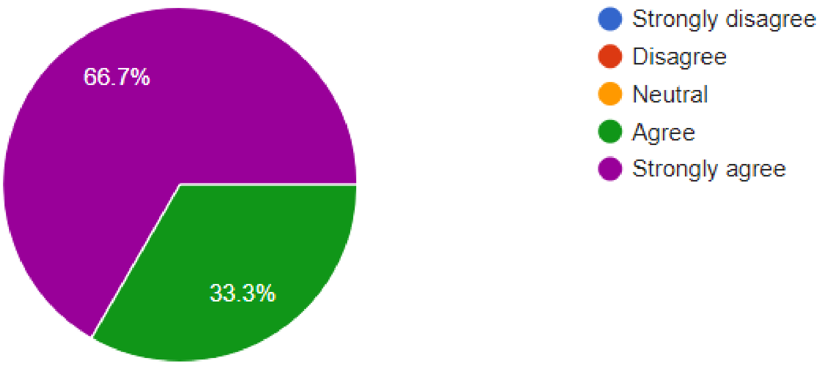


I thought the game was easy to play

3 responses

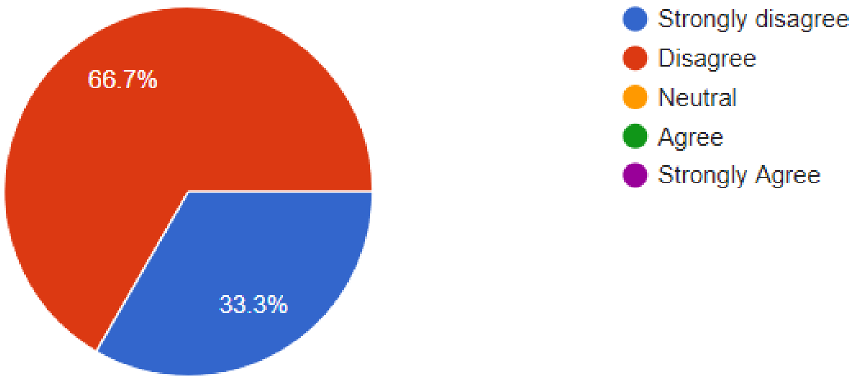
I found the game features well integrated

3 responses



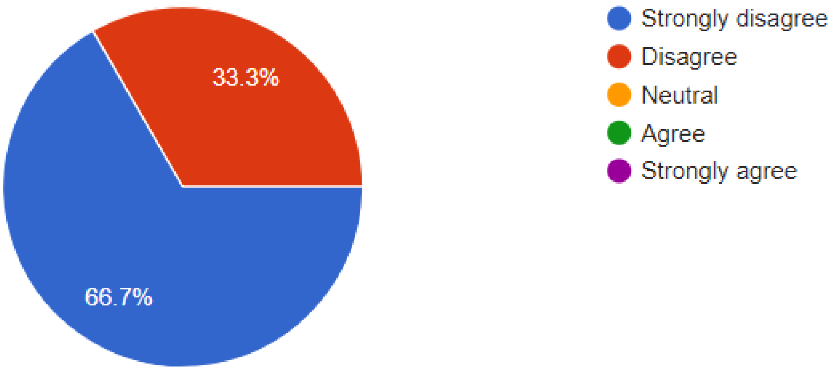
I found sections game confusing while playing it

3 responses



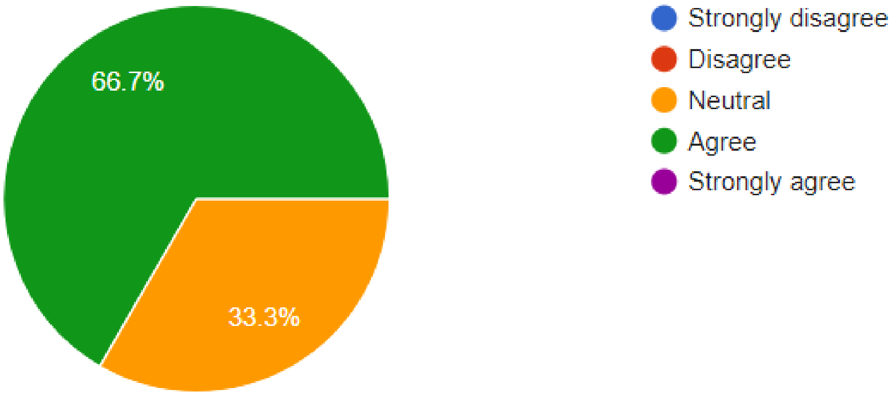
I needed to learn a lot of things in order to play the game effectively

3 responses



I like the games style

3 responses



Other Material Used

- O'Reilly | Safari. 2017. The Indie Game Developer Handbook [Book]. [ONLINE] Available at: https://www.safaribooksonline.com/library/view/the-indie-game/9781138828421/. [Accessed 29 November 2017].

- O'Reilly | Safari. 2017. Learning Path: Unity: Easy Game Development with Unity 5 [Video]. [ONLINE] Available at: https://www.safaribooksonline.com/library/view/learning-path-unity/9781788470278/. [Accessed 29 November 2017].

Unity Used

- Unity Asset Store. 2018. Standard Assets for Unity 4.6 - Asset Store. [ONLINE] Available at: https://assetstore.unity.com/packages/essentials/asset-packs/standard-assets-for-unity-4-6-21064. [Accessed 03 May 2018].

- Unity Asset Store. 2018. Lava Flowing Shader - Asset Store. [ONLINE] Available at: https://assetstore.unity.com/packages/vfx/shaders/lava-flowing-shader-33635. [Accessed 03 May 2018].

- Unity Asset Store. 2018. TextMesh Pro - Asset Store. [ONLINE] Available at: https://assetstore.unity.com/packages/essentials/beta-projects/textmesh-pro-84126. [Accessed 03 May 2018].

- Unity Asset Store. 2018. Machine Guns - Asset Store. [ONLINE] Available at: https://assetstore.unity.com/packages/3d/props/guns/machine-guns-20611. [Accessed 03 May 2018].

BLANK

PAGE

STEPHEN LOWRY
X12356291