

# 3D Games Programming

Interaction based Game  
Development

Gordon Johnson  
K1451760

## Table of Contents

Overview / Links.....	2
Introduction.....	3
Plot .....	3
Tasks to be Implemented .....	4
Instructions.....	6
Interfaces .....	6
Controls .....	8
Understanding the Game Objects.....	10
Techniques / Algorithms / Tools.....	12
Version Control.....	12
The Game Controller.....	12
Player Controller .....	16
NPC Jeeps .....	18
Interaction Mechanisms.....	19
Game Controls Interaction .....	19
Reward Mechanics .....	20
References.....	21
Appendix One – GitHub Commits .....	23
Appendix Two – Game Controller Data Dictionary.....	31

## Overview / Links

Buggy Boy (1985) meets Fastlane: Road to Revenge (2017)

**Name:** Buggy Boom

**Type:** Endless Driving, avoid obstacles, defeat enemies, collect rewards and level up.

**Theme:** Non-realistic, Fantasy Violence

**Audience:** 12+ Aimed at the younger age group

**Technology:** Unity3D 2017.2

**GitHub:** [https://github.com/LordGee/buggy\\_boom](https://github.com/LordGee/buggy_boom)

**Video:** <https://youtu.be/LNH09MziiIE>



Figure 1 - Inspired by: Buggy Boy (Top Left), Fastlane (Right) and Clicker Hero (Bottom Left)

## Introduction

The proposed game, will be based upon the genre of 'Endless Runner'. Taking inspiration from Fastlane, which is a 2D game created by London based company 'Space Ape'. The objective of which, is to play for as long as possible whilst avoiding obstacles and destroying enemy cars along the way, enabling a player to collect in-game cash, gems and level-up. By obtaining enough cash / gems the player can improve the performance of their car. The retro 'Buggy Boy' game, originally released for the 8-bit generation of computing, provided a behind the car view (third person view) with similar objectives as the 'Fastlane game', such as avoiding obstacles and collecting bonus points to obtain additional score. Higher points were given for the bigger flags, these had to be driven through as opposed to over. The final inspiration for this game is taken from 'Clicker Hero', while not a driving game, it utilises a Gamification feature of levelling and progression, that I will attempt to include into this game.

Using the above titles as inspiration, I propose to make a game that will portray a 3D perspective, presenting the camera behind the car like in 'Buggy Boy'. The play surface will be a flat straight surface, that will represent the 'Fastlane' concept and provide obstacles and action of a similar nature.

## Plot

Can you survive the dark, foggy night on the never-ending road? Armed with a trusty projectile cannon, players battle through the endless hordes of rival vehicles that will stop at nothing to bring your journey to an end. For each neutralised rival, you will accumulate coins that can be used to upgrade your vehicle and improve your chances for survival, every penny counts! Increasing your player level, provides an additional bonus whilst spawning more challenging rivals. Survive long enough and you may come bumper to bumper with the boss rival, who will put your driving skills to the test, by firing a volley of projectiles at you. Defeat them before they defeat you and earn extra bonuses.

## Tasks to be Implemented

The following, has been prioritised using the MoSCoW method. This will be a working document, as such priorities may change throughout the course of the project:

### Must Have:

- A surface which resembles a road. The material offset should be updated to provide the illusion that the road is moving.
- A buggy (the player pawn). This should be fixed on certain axis, to allow only left and right movement. Other constraints should apply, to ensure the player remains within the play space.
- Rival vehicles. These should be spawned in a random road lane, at the opposite end of the road surface. Updates should include a continuous movement towards the player. If a player collides with a rival, then the players' health is reduced.
- Projectiles, so the player can shoot at oncoming rivals. If a projectile collides with a rival game object, that object will have its health reduced, if the health reaches to zero or below, then that game object will be destroyed.
- Shredder. A game object should be placed directly behind the player, to catch and destroy any game objects that pass the players line of sight.
- UI Display. To provide score, health and multiplier statistics.

### Should Have:

- Road block objects, to provide an unpassable obstacle. To spawn in rows covering the road, a condition must be that there is always a safe route through even if that is just one lane. Road blocks, should not be destroyed by the player and the player colliding with the road block will trigger an instant game over.
- Additional rival vehicle, which spawn randomly and shoot projectiles back at the player.
- A layer of fog, that obscures the view of road ahead, providing a challenging field of view.
- Reward pick-ups, should be spawned when a rival is destroyed. These pick-ups, can include additional score or repairing tools.
- A game over scene, which displays the game score and the players new running total score.
- A menu system, to start the game, manage user options (such as volume levels) and display a running total of score.

**Could Have:**

- A boss vehicle, that has a large amount of health to spawn after a period of play time. The boss should have a different movement mechanic than the rival vehicles. It should travel towards the player, until a certain point is reached and then remain in front of the player, performing a side to side movement until it is destroyed.
- Additional rival vehicle that simulates a simple AI, that changes lane towards the players' lane location.
- Each rival vehicle, should have a unique material or prefab attached, so the rival type can easily be visually identified.
- Building objects spawn either side of the play space, providing scenery and depth to the game.
- The menu scene. To contain a levelling up display, as well as an option to upgrade the buggies fire power, minimum health and minimum multiplier.
- Animation of the player vehicle, for changing direction and idle positions. This should include rotation of the vehicle and wheel movement.

**Won't Have:**

- Game variables, that are passed from different scenes, should be encrypted to prevent cheating.
- Access online database to store and retrieve results. These can be used to provide an online leaderboard of results, creating a competitive game experience.

## Instructions

### Interfaces

Upon starting Buggy Boom, you will be presented with a menu screen. This screen displays several options which include:



Figure 2 - Main Menu

The options menu, provides configurable settings, that will affect in-game experiences. The music volume, changes the level of the ambient background music, within all scenes of the game. The SFX volume, changes the level of the sound effects within the game itself. Auto Fire, provides an option to auto detect when a destroyable enemy is in range and targetable. It will automatically fire projectiles at the enemy. The Accelerometer option, is available but is only compatible with accelerometer devices such as a smart phone, it allows the vehicle to be controlled by rotating the mobile device.

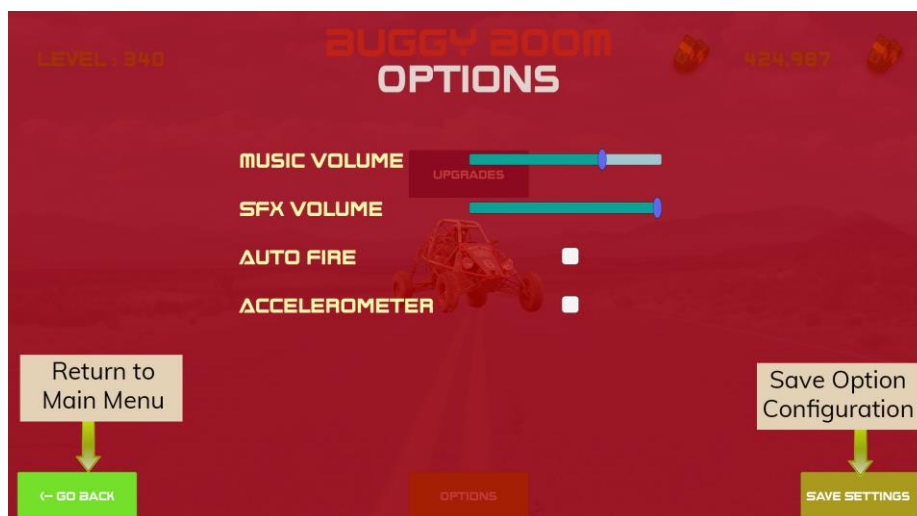


Figure 3 - Options Menu

For any changes to be committed, the ‘Save Settings’ button, must be pressed before returning to the main menu.

The Upgrades menu, allows you to upgrade your vehicle, providing you have collected sufficient money. ‘Fire power’ increases your starting fire power, by 1 point per level increase, making it easier to destroy stronger opponents as the game progresses. ‘Health’ increases your maximum health by 1 point per level increase, when taking damage, you can repair to this new higher limit. ‘Multiplier’ increases your minimum multiplier by 0.1x point per level increase, whenever the multiplier is reset during play, it will now reset to this new minimum, allowing a player to accumulate addition money.

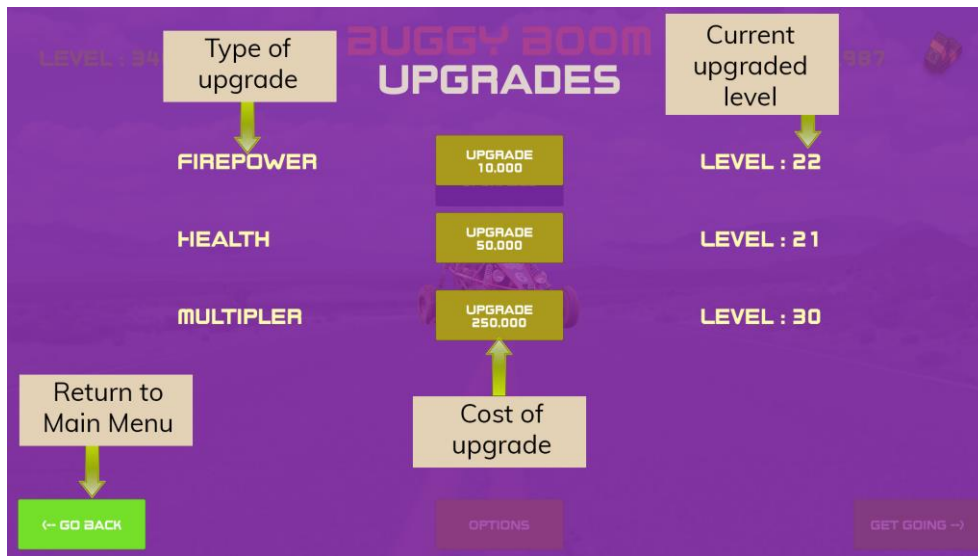


Figure 4 - Upgrades Menu

The main game view, displays the score multiplier, game score, player health and the players' buggy position.



Figure 5 - Game Interface



## Controls

There are several methods of controls that can be used to play the game:

### Keyboard Control

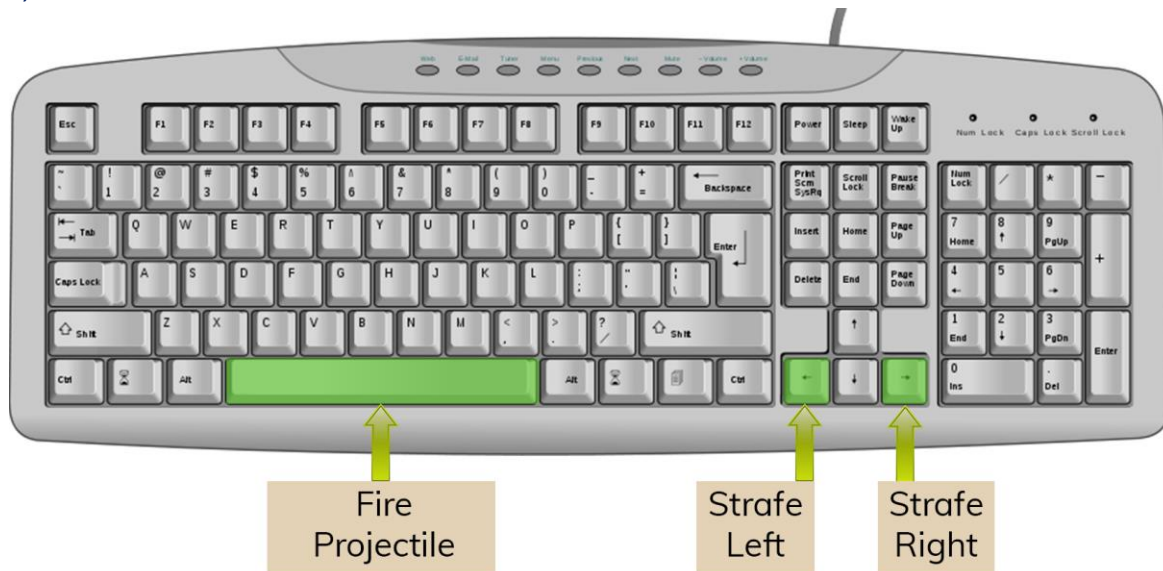


Figure 6 - Keyboard Controls

### Gamepad Control

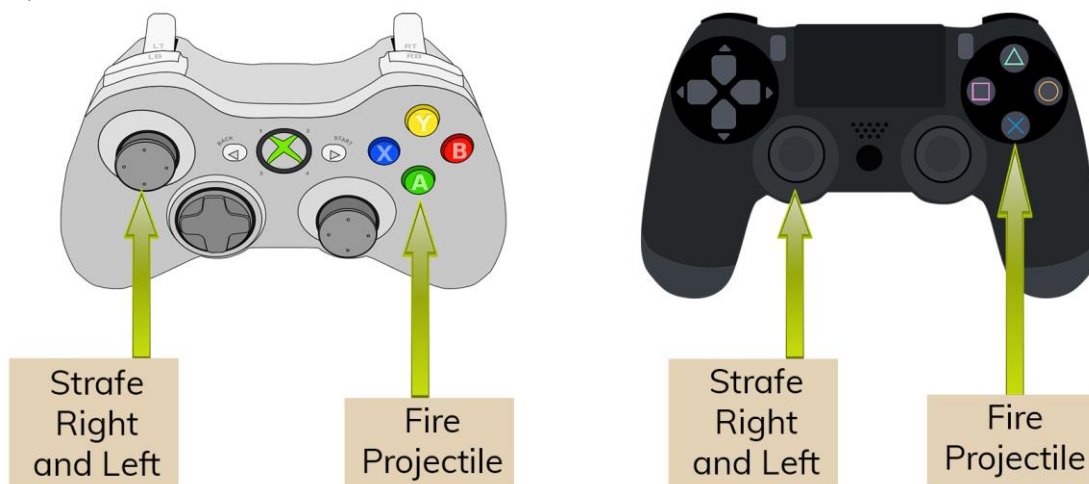


Figure 7 - Gamepad Controls

### Mobile Device Controls

By default, the mobile controls are set to touch, continuous shooting will also be enabled.

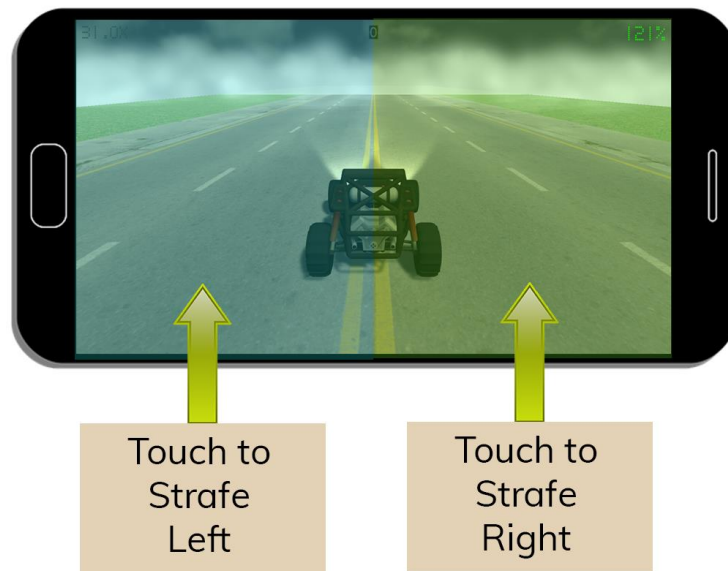


Figure 8 - Mobile Device Controls (Default)

Within the options menu, if the 'Accelerometer' is enabled, the following controls will be applied, continuous shooting will also be enabled.

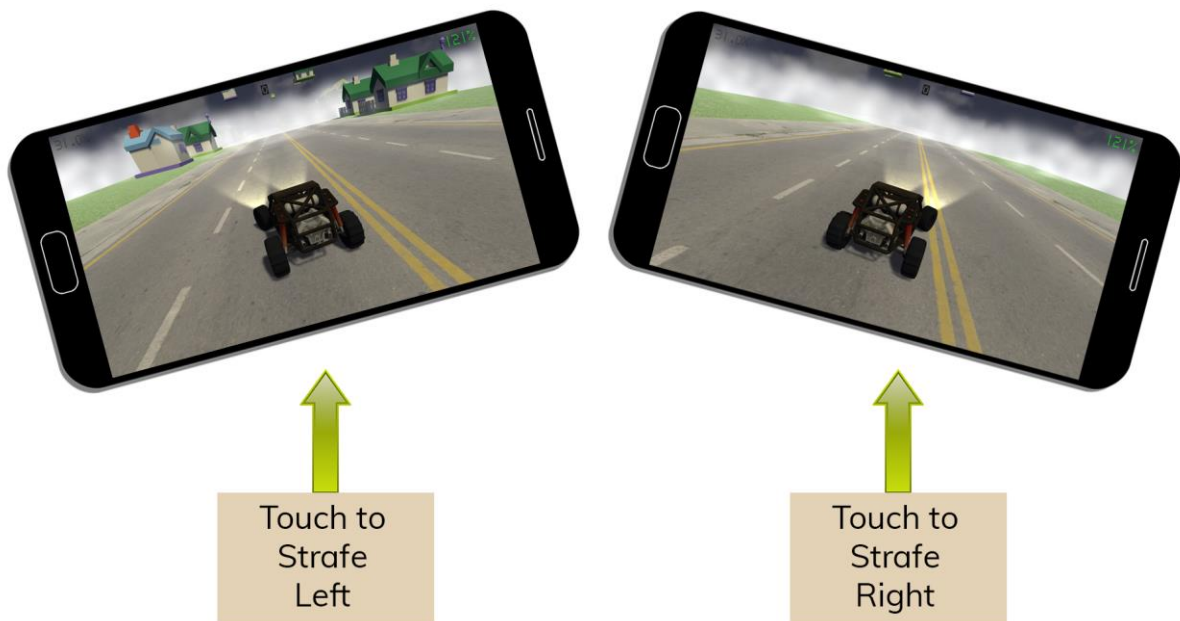





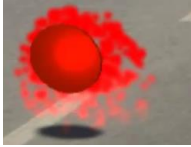
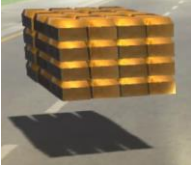



Figure 9 - Mobile Device Controls (Accelerometer)

## Understanding the Game Objects

The following, provides a persona for each object that can be found within the game:

Who it is	What it looks like
<p><b>The Green Jeep</b></p> <p>Drives only in the lane that it was spawned.            Slower speed then player.            Colliding reduces player health.            Destroy for potential bonus.</p>	
<p><b>The Orange Jeep</b></p> <p>Drives only in lane that it was spawned.            Randomly, shoots projectiles towards player.            Projectiles, reduce players' health.            Slower speed then player.            Colliding reduces player health.            Destroy for potential bonus.</p>	
<p><b>The Blue Jeep</b></p> <p>Moves towards the players' current position,            changing one lane at a time.            Slower speed then player.            Colliding reduces player health x2.            Destroy for potential bonus.</p>	
<p><b>The Road Block</b></p> <p>Can block up to 5 out of 6 lanes.            Are stationary.            Can't be destroyed.            Colliding will remove all players' health.</p>	

Who it is	What it looks like
<p><b>The Boss</b></p> <p>Stays in middle of road until close to player.            At set position, will strafe from right to left.            Randomly, shoots projectiles towards player.            Projectiles reduce players' health.            Will not get close enough to collide with.            Destroy for potential large bonus.</p>	
<p><b>Your Projectiles</b></p> <p>Green in colour.            Always fire in a forward facing direction.</p>	
<p><b>Rival Projectiles</b></p> <p>Red in colour.            Always fires backwards, in the direction of the players' position at the time of instantiation.</p>	
<p><b>Coins</b></p> <p>Collect 'coins' for an additional 20 points.</p>	
<p><b>Gold Bars</b></p> <p>Collect 'gold bars' for an addition 100 points.</p>	
<p><b>Tool Kit</b></p> <p>Collect 'tool kits', to repair your buggy health by 5 health points. If the buggy health is already at maximum or close to it, the health points are converted into game points.</p>	

## Techniques / Algorithms / Tools

### Version Control

For this project, I chose to use GitHub to manage all project files. This tool provides a road map of events, that transpired throughout the implementation of this game. It provides a solid back up of all relevant game files and changes. This came in handy when an animation broke the program, as I was able to use the information stored on GitHub, to quickly revert back to a previous state. A complete record of commits and comments can be found in (Appendix One).

### The Game Controller

The ‘Game Controller’ script, forms the heart of the gameplay logic; it manages all general variables that are required at the start of the game. As a central point, it manages general functions that have been made public, allowing them to be quickly called from other scripts, which in turn will affect the remainder of the gameplay.

A form of a data dictionary, of all the variables managed by the ‘Game Controller’, can be found in (Appendix Two) towards the back of this report.

The ‘Game Controller’, starts its life in a new game, by declaring set variables. Some of the initial calculations, require access to the player preferences script, in order to obtain the players upgrade variables. It then proceeds to the ‘Update’ function, where a switch statement is used to determine the current game state. As seen in (figure 10), in order to perform the appropriate operations for that game state, the behaviour branches off in three directions.

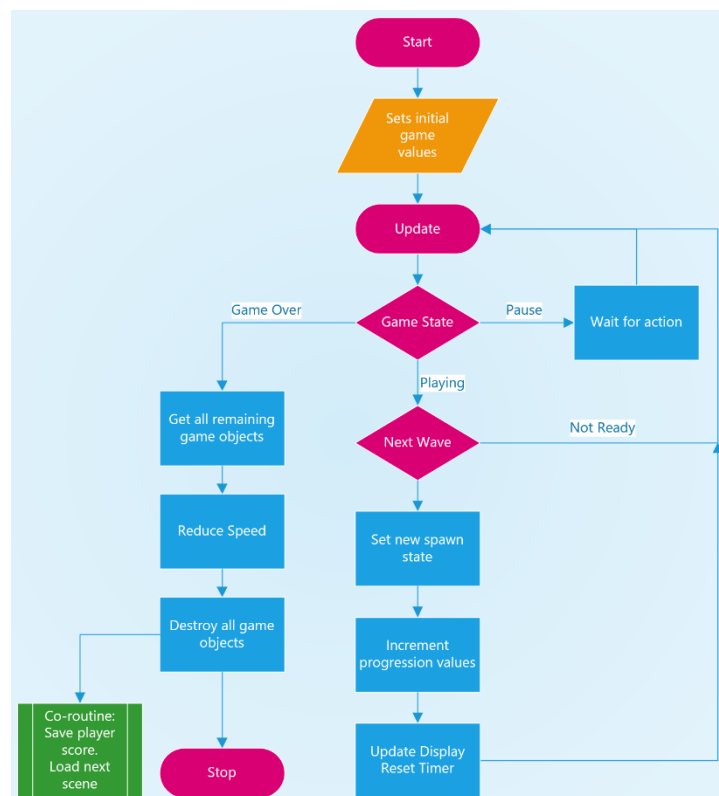


Figure 10 - Game Controller Behaviour

The ‘Game Controller’, also contains other functions as described below. The public functions can be called upon by other scripts, variables within the game controller will be updated as required.

#### *Damage Player*

The ‘Damage Player’ function, takes in one argument, this defines the amount of damage the player receives. It checks if the player is currently invincible, if not then the players’ health is reduced by the amount of damage passed into the function, the multiplier is reset to its minimum value (depending upon upgrades). The invincible Boolean is set to true and a co-routine is started, this changes back to false after a fifth of a second. The function then checks if the players’ health has been reduced to zero or less, if this is true then the players’ health is set to zero for cosmetic purposes and the game state changes to ‘Game Over’. Finally, the game canvas text is updated with the new values.

```
public void DamagePlayer(float _dmg)
{
    if (!invincible)
    {
        playerHealth -= _dmg;
        playerMultiplier = 1 + minMultiplier;
        invincible = !invincible;
        StartCoroutine(PlayerHit());
    }
    if (playerHealth <= 0)
    {
        playerHealth = 0;
        currentyGameState = GAME_STATE.GameOver;
    }
    UpdateHUD();
}

private IEnumerator PlayerHit()
{
    yield return new WaitForSeconds(0.2f);
    invincible = false;
}
```

This function is called upon by the ‘Player Controller’ and the ‘NPC Obstacle’ scripts.

*Damage NPC*

The ‘Damage NPC’ function, takes in four arguments, the game object itself, the amount of points, the amount of damage associated with this specific game object and finally a reference to the game objects health variable. If necessary, the game object can be destroyed from here. The function first removed the amount of damage from the NPC’s health, because this is a referenced variable, it will affect the value in the script that called this function. It then checks if the health is zero or less, if this is true the points are awarded to the players score. Next, the script checks if the damaged object is a boss (in hindsight, this condition statement can be improved by using the `_obj.tag` to check this). If it is a boss, the ‘Spawn Collectables’ function is executed multiple times, to provide a larger reward compared to defeating a regular rival, a random range is passed in to minimise objects spawning on top of each other. In order to increase the challenge, variables are then set for the boss timer and level up. The current spawn state, is changed back to jeep and an audio clip is set. Outside this statement, the audio clip is played, an explosion effect is spawned at the objects position and the game object is destroyed.

```
public void DamageNPC(GameObject _obj, float _pts, float _dmg, ref float _hea)
{
    _hea -= _dmg;
    if (_hea <= 0)
    {
        AddPoints(_pts);
        if (currentSpawn == SPAWN_NPC.Boss && GameObject.FindWithTag("NPCBoss"))
        {
            for (int i = 0; i < Mathf.Floor(playerMultiplier * 5); i++)
            {
                SpawnCollectable(new Vector3(_obj.transform.position.x +
                    Random.Range(-2.0f, 2.0f),
                    _obj.transform.position.y,
                    _obj.transform.position.z + Random.Range(-2.0f, 2.0f)));
            }
            bossTimer = Time.timeSinceLevelLoad;
            LevelUp();
            progressionTimer = Time.timeSinceLevelLoad;
            currentSpawn = SPAWN_NPC.Jeep;
            audio[0].clip = bigExplosion;
        }
        else
        {
            SpawnCollectable(_obj.transform.position);
            audio[0].clip = smallExplosion;
        }
        audio[0].transform.position = _obj.transform.position;
        audio[0].volume = playerPrefs.GetSfxVolume();
        audio[0].Play();
        NpcDeathEffect(_obj.transform.position);
        Destroy(_obj);
        UpdateHUD();
    }
}
```

This function, is called upon by the ‘NPC Obstacle’ and the ‘Boss NPC’ scripts.

### *Spawn Collectables*

The 'Spawn Collectables' function, takes in one argument, the vector position of the soon to be destroyed game object. This small function, plays a game of chance, where each game object (which are added to the array in desirability order) has a chance to spawn. On the first iteration for the coins, if the random number equals 1, then that gets added as the spawn value. On the second iteration for the gold bars, if the random number equals 1, then the gold bars replace the coins in the spawn value. However, if the random number is anything else, then the spawn value is locked with the coins and no other checks matter. By the end of the iterations, whichever object the spawn value indicates to, will get created. Due to this, it is possible for the spawn value to remain at -1, which will not spawn anything.

```
private void SpawnCollectable(Vector3 pos)
{
    int spawnValue = -1;
    bool next = true;
    for (int i = 0; i < collectables.Length; i++)
    {
        if (Random.Range(0, 2) == 1 && next)
        {
            spawnValue = i;
        }
        else
        {
            next = false;
        }
    }
    if (spawnValue != -1)
    {
        Instantiate(collectables[spawnValue], pos, Quaternion.identity);
    }
}
```



## Player Controller

The ‘Player Controller’ script, manages all interaction between the player and the game. Its primary function, is to determine the correct control schema to use and to execute the appropriate methods for that schema. Through searching (Unity Docs, 2017), there was an interesting way to identify the build platform of the project, this allowed statements to be set based on the build, and to define which appropriate control mechanics to use. So if the build was compiled for a mobile device, then during the ‘Update’, the correct control function for a mobile device would be executed.

```
void Update ()
{
#if UNITY_EDITOR
    StandardBuilds();
#endif

#if UNITY_PS4
    PS4Builds();
#endif

#if UNITY_IOS
    MobileBuilds();
#endif

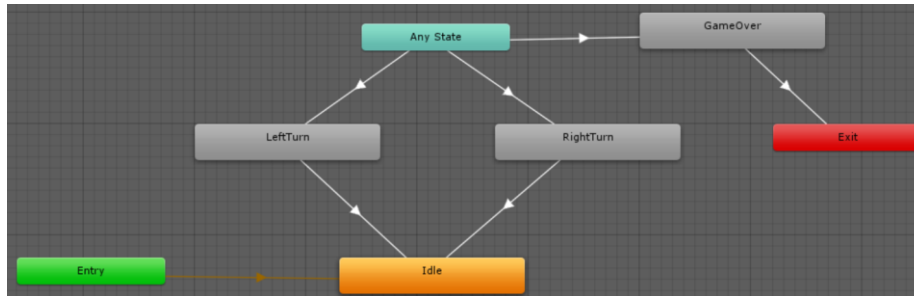
#if UNITY_ANDROID
    MobileBuilds();
#endif

#if UNITY_WEBGL
    StandardBuilds();
#endif

#if UNITY_STANDALONE_WIN
    StandardBuilds();
#endif
}
```

*Check Buggy Rotation*

This function, originally began with the idea that when the player moves the buggy left or right, the buggy itself would rotate. This later changed to being managed by the animator, this provided much smoother and more realistic movement of the buggy. It was set up, to ensure all changing states could be actioned from any position, returning to idle when no longer necessary. The entry for the three states, is determined by parameter Booleans. In hindsight, for the left and right animation, these could have been set as a float, based on the horizontal user input.



The script that handles this parameter, simply determines if the input value is a positive or negative value and sets the Booleans accordingly. After this is done, it checks for any constraints on the movement, such as if the player is going out of bounds and if so constricts this movement.

```

private void CheckBuggyRotation(float _direction)
{
    if (_direction != 0)
    {
        if (_direction > 0)
        {
            // transform.Rotate(0f, GetTranslatedPosition(buggySpeedRot), 0f);
            anim.SetBool("TurnRight", true);
            anim.SetBool("TurnLeft", false);
        }
        else if (_direction < 0)
        {
            anim.SetBool("TurnRight", false);
            anim.SetBool("TurnLeft", true);
        }
    }
    else
    {
        // transform.Rotate(0f, GetTranslatedPosition(transform.rotation.y * -1,
        buggySpeedRot * 20f), 0f);
        anim.SetBool("TurnRight", false);
        anim.SetBool("TurnLeft", false);
    }
    CheckPositioningConstraints();
}

```

## NPC Jeeps

The ‘Jeep’ script deals with all three types of Jeeps. A randomly generated value, determines which Jeep is created. Once decided, variables are configured for this type of Jeep, allocated its unique attributes. With the exception of the standard green Jeep, which just does a standard movement, the update checks for an action timer. If the time has reached a set frequency, then the script will check if the Jeep is a shooter. If it is, it will shoot a projectile in the direction of the players’ position, at the time of the projectile being instantiated.

```

if (Time.timeSinceLevelLoad - actionTimer > actionFreq)
{
    if (shooter)
    {
        ShootProjectile();
        audio.Play();
    }
    else if (changer)
    {
        currentIndex = FindCurrentIndex(transform.position.x, minMaxPos);
        playerIndex = FindCurrentIndex(player.gameObject.transform.position.x,
                                        minMaxPos);

        if (playerIndex < currentIndex)
        {
            changerMoveCount = -2f;
        }
        else if (playerIndex > currentIndex)
        {
            changerMoveCount = 2f;
        }
    }
    actionTimer = Time.timeSinceLevelLoad;
    actionFreq = Random.Range(minAction, maxAction);
}
if (changerMoveCount < -0.1f)
{
    moverScript.ChangeLane(Vector3.left.x * 0.1f);
    changerMoveCount += 0.1f;
}
else if (changerMoveCount > 0.1f)
{
    moverScript.ChangeLane(Vector3.right.x * 0.1f);
    changerMoveCount -= 0.1f;
}
else
{
    moverScript.ChangeLane(0f);
}
}

```

To simulate an artificial intelligence within the game, at random intervals the Changer Jeep will change its position, to a lane one closer to the players’ position. This is done by calculating the player and Jeep positions, within the lane array returning the index value. From this, the Jeep can work out which direction to travel in next. The distance between each lane is 2 points on the X axis. I used a counter, to gradually spoon feed the script that controls the Jeep movement, an incremental amount until the counter returned to zero. This allowed the Jeep to finish moving, once 2 points in total had been passed through.

## Interaction Mechanisms

### Game Controls Interaction

For this game, I wanted to try and include as many appropriate control interfaces, as reasonably possible. Initially starting with the keyboard mechanics, which just required three keys and the mouse, for navigating the menu systems. Having tested the game with a gamepad, I found that the menu system was inactive and unable to start the game. In the Inspector of the 'Event System', I found an option called 'First Selected' that takes in a game object, I declared this as the 'Start' button. This presented the start button as already highlighted and was able to start the game. However, navigating was not intuitive and only some buttons were able to be

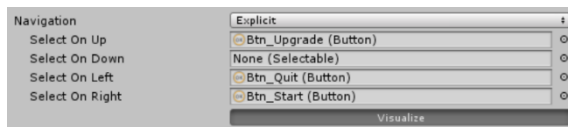


Figure 11 - Explicit navigation options

menu options.

When tested on the PS4 development kit, the navigation failed again, with no starting button highlighted at all. After some searching, I found another option in the Inspector, for the 'Event System' object, this was displayed as 'Force Module Active'. After checking this box, the navigation seemed to come alive again. However, this led to another problem, if the PS4 controllers track pad was accidentally touched, then the selected button became lost and no longer retrievable without restarting the game. After spending some time trying to remedy this issue, I settled for creating a 'Reset' button, so if this was to happen, pressing the circle button on the controller would reset the menu to its default state.

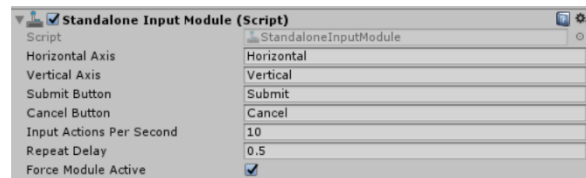


Figure 12 - Force Module Active was required to be checked

When testing on a mobile device, the touch mechanics appeared to work well. With only some minor code changes required, to ensure projectiles are constantly firing. The game on the mobile device, was very jumpy and not very responsive, but I believe that was due to the very old android device I was using. Testing the 'Accelerometer', on this device proved very difficult. The vehicle appeared to move at a very slow pace compared to when using other control methods. It appears that the movement speed of the vehicle, would need to be doubled to make this an effective control method. Unfortunately, I have been unable to get a connection with this device since, so have been unable to test it again.

### Reward Mechanics

As stated previously, I wanted to include a reward scheme in order to improve the chance of repeat play. I implemented the 'Game Score', as a currency that can be accumulated every time the person plays. Also stored is the overall accumulation, regardless of spend, it is this value that determines the player game level. Currently this is calculated for every 10,000 accumulated increases, the players level up by 1 point. For future improvement, this calculation is incremented for each level that's passed, so 20,000 = Level 2, 40,000 = Level 3, 80,000 = Level 4, and so on. The level number, will play an intricate role in how challenging the game is. In order to make the game achievable, the player also needs some benefit to offset the greater challenge.

The 'Upgrades' menu system, enables the player to upgrade certain attributes of the vehicle and gameplay. For 10,000 the player can upgrade the 'Fire Power', this increases the damage dealt by the player, by 1 point. 50,000 upgrades the players' maximum health by 1 point. With 250,000 upgrading the player minimum multiplier by 0.1 point.

This provides the player with achievable goals to aim for, whilst still finding the game more challenging as they progress, with upgrading of the vehicle to offset some of the difficulty.

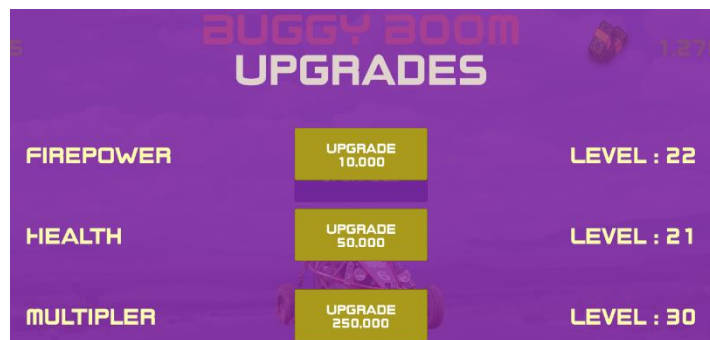


Figure 13 - Upgrade menu system

As a future improvement, I would like to look into a more appropriate method of storing these variables, without utilising the 'Player Preferences', as these can be easily accessed and manipulated. Although having the ability to manipulate these variables was beneficial during the testing period.

## References

### *Audio Assets*

Pogmog (2017) *Money Collect 2*. Available at:

<http://freesound.org/people/Pogmog/sounds/393911/> (Accessed: 21 October 2017)

Guest (2016) *Cha Ching.wav*. Available at: <http://freesound.org/people/guest/sounds/351304/> (Accessed: 21 October 2017)

Creek23 (2013) *cha ching.wav*. Available at:

<http://freesound.org/people/creek23/sounds/75235/> (Accessed: 21 October 2017)

Fins (2012) *laser*. Available at: <http://freesound.org/people/fins/sounds/146725/> (Accessed: 21 October 2017)

KenRT (2015) *Rachet.wav*. Available at: <http://freesound.org/people/KenRT/sounds/319996/> (Accessed: 21 October 2017)

Cyberlogical (2017) *Music Starter Kit*. Available at: <http://u3d.as/tJj> (Accessed: 21 October 2017)

Moonflower Carnivore (2016) *Fantasy SFX for Particle Distort Texture Effect Library*. Available at: <http://u3d.as/i8W> (Accessed: 21 October 2017)

Glitch (2012) *Glitch*. Available at: <http://www.glitchthegame.com> (Accessed: 22 October 2017)

### *3D Model Assets*

15/10

UltimateArcade (2014) *Toon Sports Car*. Available at: <http://u3d.as/6Dw> (Accessed: 07 October 2017)

Cactuscreatives Pvt. Ltd. (2014) *Car*. Available at: <http://u3d.as/8Er> (Accessed: 07 October 2017)

br3n065 (2016) *Sandrail (Dune Buggy)*. Available at: <http://u3d.as/5K3> (Accessed: 14 October 2017)

RK.Team (2017) *Modular Road Block [FREE]*. Available at: <http://u3d.as/Nt5> (Accessed: 15 October 2017)

Mister Necturus (2016) *Medieval Gold*. Available at: <http://u3d.as/6ce> (Accessed: 17 October 2017)

Ilias Kap (2017) *Workplace Tools*. Available at: <http://u3d.as/MvD> (Accessed: 18 October 2017)

ZugZug Art (2017) *Cartoon Monster Truck*. Available at: <http://u3d.as/j91> (Accessed: 21 October 2017)

LowlyPoly (2017) *Simple Houses Lite*. Available at: <http://u3d.as/Fny> (Accessed: 22 October 2017)

#### *Texture / Materials / Skybox/ 2D Assets*

Dmitriy Chugai (2013) *Seamless road texture*. Available at: [http://texturelib.com/texture/?path=/Textures/road/road/road\\_road\\_0021](http://texturelib.com/texture/?path=/Textures/road/road/road_road_0021) (Accessed: 07 October 2017)

GameWarming (2016) *Autumn Mountain*. Available at: <http://u3d.as/mZv> (Accessed: 07 October 2017)

Arctic\_Foto (2015) *Off-road-rally-buggy-race-offroad*. Available at: <https://pixabay.com/en/off-road-rally-buggy-race-offroad-1917039/> (Accessed: 22 October 2017)

Francisco Morais (2016) *Black Top Road Under Clear Blue Cloudy Sky*. Available at: <https://www.pexels.com/photo/dessert-roadtrip-nevada-united-states-90633/> (Accessed: 22 October 2017)

OpenClipart-Vectors (2013) *Banknotes-bankroll-bill-money*. Available at: <https://pixabay.com/en/banknotes-bankroll-bill-money-159085/> (Accessed: 24 October 2017)

Qianyuez (2016) *Free Night Sky*. Available at: <http://u3d.as/G26> (Accessed: 25 October 2017)

#### *Fonts*

Bou Fonts (2009) *Score Board*. Available at: <https://www.dafont.com/score-board.font> (Accessed: 14 October 2017)

Woodcutter (2015) *Scoreboard*. Available at: <https://www.dafont.com/scoreboard.font> (Accessed: 14 October 2017)

Jonathan Paterson (2016) *Boom Box*. Available at: <https://www.dafont.com/boom-box.font> (Accessed: 22 October 2017)

#### *Coding References*

Unity Docs (2017) *Random.ColorHSV*. Available at: <https://docs.unity3d.com/ScriptReference/Random.ColorHSV.html> (Accessed: 07 October 2017)

Unity Docs (2017) *Platform dependent compilation*. Available at: <https://docs.unity3d.com/Manual/PlatformDependentCompilation.html> (Accessed: 07 October 2017)

## Appendix One – GitHub Commits

Commits on Oct 25, 2017

**Finish Push** ...

LordGee committed 3 days ago

- + Updates where required to the event system due to the PS4 not enforcing certain components. Fortunatley the module itself had a tick box to force itss use.
- + Updated the menu script due to the PS4 controller touch pad, if you accidently touch this pad within a UI experience the selected items is lost and navigation is no longer posible, unfortunatley I couldn't just enforce this if the selection was lost... for some weird reason! Instead created a short script that reseted the menu system with the push of a Fire2 button.
- + Updated the results scene for the same reason as above

+ PS4, PC, Android and Web builds have been completed off this push,

fcff155

↔

**UI Controller Support** ...

LordGee committed 3 days ago

- + Added Controller support for the user interface... otherwise we won't be able to start the game on a PS4 lol

Now I'm going to uni

603828b

↔

**Control Schemes** ...

LordGee committed 3 days ago

- + Added support for standard, mobile and PS4 control schemes. Tested on Stardard and mobile devices. Will be heading to University to test the PS4 build on the dev kits.

NOTE: Hopefully this will be the final push / commit for this project as GitHub appears to struggle after 1gb of data... probably my fault!

1f47ea3

↔

**Finishing touches, 1 task left** ...

LordGee committed 4 days ago

- + Fixed issue with the Material offset starting at the wrong speed on a second go. Switch out Time.time for timesSinceLevelLoad.
- + The third jeep now transitions between lane, also at a much smother rate.
- + Changed skybox to make it nighttime, and added headlights to the buggy.
- + Added new SkyBox. Credit to <http://u3d.as/G26>

ced78ab

↔

Commits on Oct 24, 2017

**Moving Jeep** ...

LordGee committed 4 days ago

- + Attempting to swap lanes with third jeep
- + Currently moves in the right direction, but does still crash into the side wall

TODO: Need to move one lane at a time

cd1f23b

↔

**Quite a bit, read details** ...

LordGee committed 5 days ago

- + Added picture of Bank Notes. Credit to : <https://pixabay.com/en/banknotes-bankroll-bill-money-159085/>
- + Completed the results screen, variables are stored and displayed.
- + Updated main menu to display players level and money.
- + Completed upgrade canvas and three items are upgradable.
- + Updated all game scripts to enforce the new changes

e0e098a

↔

Commits on Oct 23, 2017

**Results Scene Layout** ...

LordGee committed 5 days ago

- + Added layout to results scene
- + Added variables in the player prefs scrip to manage score storage

388e690






















↔

GORDON JOHNSON


23




Commits on Oct 22, 2017

 <b>Options Menu and Much More</b> ... LordGee committed 6 days ago + Added an options screen to manage, music and sfx volumes and indicate whether the player would like to use auto fire or accelerometer features. + Updated all other scripts to reflect any option changes + Fixed reward pickups to only be collected by the player + Added a larger delay to the Boss and speed up all enemy projectiles + Added trigger event to destroy particle effects that do not get destroyed + Updated projectile damage from jeep to make them more powerful + Added menu control script to manage all main menu interaction + Added a Player Prefs script to manage all stored variables, getting and setting + Updated repair pickup and reduced repair amount from 20 to 5	 ea15b54 
 <b>Fixed Bug</b> ... LordGee committed 6 days ago + Fixed issue where if a house moves out into the road will destroy the player but not end the game, with Game Over.	 c554ae8 
 <b>Implementing the Main Menu Screen</b> LordGee committed 6 days ago	 30dd892 
 <b>Level Management</b> ... LordGee committed 6 days ago + Added a level controller as a prefab to be included in every scene + Added a function to load the next scene automatically after a period of time + Added a function to quit out the game completely	 e7202ee 
 <b>Created simple splash screen</b> ... LordGee committed 6 days ago + Added splash screen sound. Credit to : <a href="http://www.glitchthegame.com">http://www.glitchthegame.com</a> + Added background image. Credit to : <a href="https://www.pexels.com/photo/dessert-roadtrip-nevada-united-states-90633/">https://www.pexels.com/photo/dessert-roadtrip-nevada-united-states-90633/</a> + Added heading font. Credit to <a href="https://www.dafont.com/boom-box.font">https://www.dafont.com/boom-box.font</a> + Added picture of buggy with modification. Credit to : <a href="https://pixabay.com/en/off-road-rally-buggy-race-offroad-1917039/">https://pixabay.com/en/off-road-rally-buggy-race-offroad-1917039/</a>	 9b39be2 
 <b>Houses</b> ... LordGee committed 6 days ago + Added low poly houses, credit to: <a href="http://u3d.as/Fny">http://u3d.as/Fny</a> + Updated scripts to spawn the low poly houses instead of primitive cubes + Refactored Game Controller script a little more when dealing with the game over sequence.	 8042852 
 <b>Game Over</b> ... LordGee committed 7 days ago + Added game over animation + Implement a game sequence that takes the player to the results scene + Stop all objects from moving + Slows other objects and offsets to minimum  BUG: 50/50 whether the animation sequence gets activated	 e5efc92 


Commits on Oct 21, 2017

 **Music Controller** ...  
 LordGee committed 7 days ago 📄 8e1c2ba <>

- + Added music controller to play correct music depending on Scene
- + Updated progression incremter to make the game more challenging
- + Added a slightly bigger gap between roadblocks, to prevent impossible situations
- + Reduced the Max Speed of Jeeps and Blockers, as they ended up moving faster then their projectiles.
- + Added a hit SFX to the player when hit by a projectile.

 **Sound Effect and Music** ...  
 LordGee committed 7 days ago 📄 e270d4d <>

- + Added Coins SFX reference: <http://freesound.org/people/Pogmog/sounds/393911/>
- + Added Gold SFX reference: <http://freesound.org/people/guest/sounds/351304/>
- + Added Gold2 SFX reference: <http://freesound.org/people/creek23/sounds/75235/> (CC Non Commercial)
- + Added Lazer SFX reference: <http://freesound.org/people/fins/sounds/146725/>
- + Added Repair SFX reference: <http://freesound.org/people/KenRT/sounds/319996/>
- + Added Music Loops for ambience sound reference : <http://u3d.as/tj3>
- + Added Explosion SFX reference: <http://u3d.as/i8W>
- + Added all sound effects to various areas within the game


 **Refactoring + Comments** ...  
 LordGee committed 7 days ago 📄 0470a66 <>

- + Revisited every line of code in all scripts, refactored and improved where necessary.
- + Added comments to functions that are not initially obvious.
- + Added Tooltips to public variables to provide context within the Unity Inspector

Commits on Oct 20, 2017

 **Refactoring code**  
 LordGee committed 8 days ago 📄 11c44f1 <>


Commits on Oct 19, 2017

 **Boss Attacks** ...  
 LordGee committed 9 days ago 📄 645e678 <>


- + Added boss functionality that spawns after a period of time
- + Boss moves slowly into a set X position, the transforms from right to left in front of the player. Shooting is more aggressive and random.
- + Added enums to define what object is to be spawned next
- + improve enemy projectile spawn accuracy
- + Updated sscript to pass in the correct damage for type of vehicle
- + Update object spawner script to refleat the new game object boss to minimise conflicts.

TODO: Shorten the length of time the player is invinsible after damage taken

Commits on Oct 18, 2017

 **Minor improvements** ... LordGee committed 10 days ago d1da77e <>


- + Added Monster Truck asset credit to <http://u3d.as/j91>
- + Added the material for the Jepp to be dependent on whether the enemy is a shooter
- + Added back in the Sphere Cast detection for previously shot projectiles
- + Added some variables into the game controller for the boss

 **Repair Pickup, Enemy Projectiles + Improved Controls** ... LordGee committed 10 days ago 3f995bf <>

- + Added a couple of tool assets to represent a repair power up for the buggy. Credit to <http://u3d.as/MvD>
- + Added repair function that will increase the players chance of survival. If the repair exceeds the maximum the remainder is turned into gold.
- + Added enemy projectile that shoots towards the player from a standard jeep, not all jeeps shoot and are selected at random whether they are shooters or not.
- + Added constraint so jeep can only shoot a maximum of twice per spawn.
- + Updated the Sphere Cast to only detect the Jeep Enemies
- + Improved the animation for the buggy to turn this was done by removing the fixed time of the animation, also removed the rotation code to leave this effect down to the animator.


TODO: use the same random selection to define the material, this way shooters will always have the same material.  
 TODO: Going to add back in the Sphere Cast for a projectile

Commits on Oct 17, 2017


 **Money or Collectables** ... LordGee committed 11 days ago 806ead6 <>

- + Added gold coins and gold bars objects as pick-ups throughout the game. Credit to <http://u3d.as/6ce>
- + Added collectable Game Objects as an array to the game control script
- + When an NPC is destroyed a Spawn Collectable function is called, passing in the NPC position
- + The Spawn Collectable function plays a game of bingo, by calling a random number either 0 or 1, if the first number = 1 the process is repeated. Each iteration will spawn a more desirable collectable then the previous, but only if extra lucky.
- + Removed obsolete variables that have not been used


Commits on Oct 16, 2017

 **Road Block Spawn Improvements** ... LordGee committed 12 days ago 93f6ed9 <>


- + Added enum and rewrote game controller to manage the different spawn objects a bit clearer
- + Updated the progression calculator to increment as opposed to multiply
- + Improved object spawner to retrieve three random lane numbers in which to spawn several road blocks in a line, Making the game more challenging
- + Updated the Sphere Cast to ignore Road Blocks, if I remember next time I should also include to ignore the Road itself .

 **Minor cleanup** ... LordGee committed 12 days ago bef864 <>

Commits on Oct 15, 2017













 **Road Blocks** ... LordGee committed 13 days ago b5c9046 <>

- + Added road block asset from the store, credit to <http://u3d.as/NT5>
- + Refactored code, to be more generic when generating obstacles
- + Put more entherisis on the controller deciding which variables to pass back;
- + unique scripts to define the jeep or road block



 **Removal of project for testing** ... LordGee committed 13 days ago 2e1d614 <>

- + This should'nt have been pushed to the repo, the project was used for testing assets before implementing them into buggy boom


Commits on Oct 14, 2017

 <b>Animation</b> ... LordGee committed 14 days ago + Added animation to idle state of buggy, all four wheel turn, plus a clog on the back + Added animation for left and right turns  TODO: Issue with front wheels when turning, they look like they are about to fall off!	 <span style="border: 1px solid #ccc; padding: 2px;">22cf715</span> <span style="border: 1px solid #ccc; padding: 2px; margin-left: 10px;">&lt;&gt;</span>
 <b>Changed the Player Object</b> ... LordGee committed 14 days ago + Added new vehicle credit to <a href="http://u3d.as/5K3">http://u3d.as/5K3</a> + Removed many of the obsolete components to ensure it operated similar to the previous model  // TODO: Must provide animation components as it's more apparent now that the object is not moving	 <span style="border: 1px solid #ccc; padding: 2px;">33194ce</span> <span style="border: 1px solid #ccc; padding: 2px; margin-left: 10px;">&lt;&gt;</span>
 <b>RayCast to SphereCast</b> ... LordGee committed 14 days ago + Updated the RayCast to a SphereCast to that projectiles will start firing before player being on target	 <span style="border: 1px solid #ccc; padding: 2px;">7b809fd</span> <span style="border: 1px solid #ccc; padding: 2px; margin-left: 10px;">&lt;&gt;</span>
 <b>Auto Shooting</b> ... LordGee committed 14 days ago + Added a raycast for detecting NPC cars, which initiates an auto shooting + Updated the conditions to ensure what the raycast was detecting was not the walls or the previously instantiated projectile + Added a new arra of materials to provide a selection of different colours for the cars.	 <span style="border: 1px solid #ccc; padding: 2px;">6dbaeb9</span> <span style="border: 1px solid #ccc; padding: 2px; margin-left: 10px;">&lt;&gt;</span>
 <b>HUD Canvas</b> ... LordGee committed 14 days ago + Added Canvas to display the main game variables + Added game score, font credit <a href="https://www.dafont.com/scoreboard.font">https://www.dafont.com/scoreboard.font</a> + Added Multiplier and Health, font credit <a href="https://www.dafont.com/scoreboard.font">https://www.dafont.com/scoreboard.font</a> + Updated health text to display colour depending on amount left. + Fixed issue where projectiles pass through NPC when NPC requires more then one hit.	 <span style="border: 1px solid #ccc; padding: 2px;">99ceb6e</span> <span style="border: 1px solid #ccc; padding: 2px; margin-left: 10px;">&lt;&gt;</span>
 <b>Updated to version 2017.2.0 + minor collision changes</b> ... LordGee committed 14 days ago + Now projectiles are destroyed on collision with NPC + All objects except for the flooar are destroyed upon contact with the wall. Mini exlosion toi provide effects	 <span style="border: 1px solid #ccc; padding: 2px;">929a0e7</span> <span style="border: 1px solid #ccc; padding: 2px; margin-left: 10px;">&lt;&gt;</span>

Commits on Oct 13, 2017



 <b>Projectiles now Destroy NPC</b> ... LordGee committed 15 days ago + When damaging NPC the player points calc is also passed through. + Added NPC Death Effects function, to minmise repeated code + Updated the projectile spawn script, due to collision with player object causing unwanted side effects, projectile now spaawns +1f on the z + Added projectile life span to auto destroy shortly after entering the fog layer.	 <span style="border: 1px solid #ccc; padding: 2px;">f76b018</span> <span style="border: 1px solid #ccc; padding: 2px; margin-left: 10px;">&lt;&gt;</span>
--	---

Commits on Oct 11, 2017


 **Game Controller** ...

LordGee committed 17 days ago

- + Added game controller to manage major variables within the game
- + Getters and setters for all attributes
- + Added progression element that increases the games difficulty over a period of time, but at the same time increases the score multiplier.
- + Speed maxes out to ensure possible gameplay
- + Added damage to player and npc function that will apply the correct solution and provide a central location for this to take place.
- + Update NPC Jeep spawning script to grab the relevent uptodate values from the game controller and apply to newly created NPC objects



 `dc14084` 

---

 **Projectiles** ...

LordGee committed 17 days ago

- + Added Protile game object, lighting effect and particle system
- + Instantiate on fire or jump
- + Directed determined on orginal facing direction of player
- + Reduce rotation of player to 20 degrees each direction

 `7b2e462` 


Commits on Oct 11, 2017

 **Updated Project to 2017.1.2**

LordGee committed 17 days ago

 `0777df3` 



Commits on Oct 8, 2017

 **Destroy NPC Cars** ...


LordGee committed 20 days ago

- + Added two particle systems (Smoke and Explosion)
- + Added collision detection on the NPC obstacle, which will, if the collision is by the player perform the destroy object function. Prior to the object being destroyed the particle effect is instantiated and destroyed 2 seconds later.
- + Added script to move the particle effect at spawn and half rate of the object.
- + Reduced timings before first spawn of objects

NOTE: Added a 0.1f delay to the destruction of the NPC added a much nice transition



 `564a51a` 

---


 **Contained Rotation** ...

LordGee committed 20 days ago

- + Added constraint to the rotation of the player when changing direction

 `7d6d5db` 



---

 **Car Rotation** ...


LordGee committed 20 days ago

- + Added rotation translate to the player when moving horizontally
- + Added a return to start function so when there is no movement the Y rotation will gradually return to zero streaightening the car.
- + Added addition constraint to prevent the Z position being changed while rotating. As the function was similar just overloadee the previous function with the additional variable that was required.
- + Added additional scenes for later purpose, (Splash, Menu, Results)

TODO : Add addition rotational constraint to prevent the player turning the car 360 degrees



 `6496f27` 

---


 **Finish Fog / Added Camera Arm** ...

LordGee committed 20 days ago

- + Added additional fog partical system to add additional depth in front of the main fog screen.
- + Added a Camera Arm game object which parents the main camera.
- + Added script to the player arm which follows the player position, keeping the camera directley behind the player



 `f68c09b` 

---



















 **Testing Particle System** ...













LordGee committed 21 days ago

- + Playing with the idea of introducing fog / clouds with the Particle System

 `5d7b60e` 

Commits on Oct 7, 2017

 <b>Spawn Enemy Cars</b> ... LordGee committed 21 days ago + Import asset 3D model car, credit to <a href="http://u3d.as/8Er">http://u3d.as/8Er</a> + Added a spawner to generate game in random lanes + After a few attempts, pre-defined the lanes by grid reference in an array + Added rigidbody and coliders to all cars inc player + Import skybox for background, credit to <a href="http://u3d.as/mZv">http://u3d.as/mZv</a> + Changed camera position  TODO: - Attach camera to car to follow X movement, - Implement projectiles to clear path through other cars, - add alternative obstacles	 <span style="border: 1px solid #ccc; padding: 2px;">1ee5db6</span> 
 <b>Push Irrelevant</b> LordGee committed 21 days ago	 <span style="border: 1px solid #ccc; padding: 2px;">1a5e524</span> 
 <b>Random Colours to objects</b> ... LordGee committed 21 days ago + Added random colour to the generating of the walls. Obtained through unity manual <a href="https://docs.unity3d.com/ScriptReference/Random.ColorHSV.html">https://docs.unity3d.com/ScriptReference/Random.ColorHSV.html</a>	 <span style="border: 1px solid #ccc; padding: 2px;">c21c863</span> 
 <b>Add delay to Walls</b> ... LordGee committed 21 days ago + Added a delay until the first spawn object  <pre>// this reduces the amount of slipping objects</pre>	 <span style="border: 1px solid #ccc; padding: 2px;">f06e3a9</span> 
 <b>Side floors</b> ... LordGee committed 21 days ago + Added texture material for the side location (Out of Bounds area) + Added Offset movement script to mirror the road e.g. simulate the acceleration  TODO: Need to control the movment of the building to intatially simulate the acceleration. At the moment they are starting too quick, which looks like they are sliding across the ground until the rest of the materials reach full acceleration threshold.	 <span style="border: 1px solid #ccc; padding: 2px;">6af4b23</span> 
 <b>Generating Walls</b> ... LordGee committed 21 days ago + Added Wall Spawner left and right to generate random building + Added script the generate at specific intervals and child object to the spawner + Added script to create wall with random size and add translation + Added a shredder object behind the camera to destroy any ojects that go out of view.  <pre>// todo: add random material to the creation script</pre>	 <span style="border: 1px solid #ccc; padding: 2px;">1e10b52</span> 

 <b>Buggy Player</b> ... LordGee committed 21 days ago + Added Toon Sports Car as Temporary prop to simulate the Buggy, credit to <a href="http://u3d.as/6Dw">http://u3d.as/6Dw</a> + Added simple control script + Constraints added to script to prevent the buggy driving off the road	 <b>ff6a3f5</b> 
 <b>Terrain - Main Road</b> ... LordGee committed 21 days ago + Obtained road surface and side walk images (Credit to texturelib.com for free use of the textures) + Edited Image to to include side path on both sides + Created cude for terrain + Added new image as material and set tiling aspects + Created a MaterialOffset script to provide acceleration of the road which maintains a certain speed at threshold.	 <b>647470b</b> 
 <b>New Project</b> ... LordGee committed 21 days ago + Created New Project + Saved Initial Scene	 <b>cee178d</b> 
 <b>Initial commit</b> LordGee committed 21 days ago	 <b>8b41868</b> 

## Appendix Two – Game Controller Data Dictionary

*Public Variables*

Name	Data Type	Start Value	Description
<b>playerHealth</b>	Float	100	Stores and manages the players overall health.
<b>playerDamage</b>	Float	10	Defines how much damage a player will do to a rival vehicle.
<b>npcGameObjects</b>	Game Object Array	Set in Unity inspector	Stores game objects that are used by the spawn script by calling the Get function.
<b>npcJeepHealth</b>	Float	10	Health that the Jeep starts with.
<b>npcJeepDamage</b>	Float	4	Damage caused by Jeep either collision or projectile damage.
<b>npcJeepSpeed</b>	Float	10	Speed that the jeep translates its new position.
<b>npcJeepPoints</b>	Float	100	How many points are rewarded for defeating a Jeep.
<b>npcBlockHealth</b>	Float	999999	Health that the Road Block starts with.
<b>npcBlockDamage</b>	Float	999999	Damage caused by Road Block though collision damage.
<b>npcBlockSpeed</b>	Float	12	Speed that the jeep translates its new position.
<b>npcBlockPoints</b>	Float	0	How many points are rewarded for defeating a Road Block, which should never happen anyway.
<b>npcMonsterHealth</b>	Float	1000	Health that the Boss starts with.
<b>npcMonsterDamage</b>	Float	5	Damage caused by Boss through projectile damage.



<b>npcMonsterSpeed</b>	Float	5	Speed that the Boss translates its new position.
<b>npcMonsterPoints</b>	Float	1000	How many points are rewarded for defeating a boss.
<b>SPAWN_NPC</b>	Enumeration	Jeep, Block, Boss	Sets the values of the various Spawn States.
<b>currentSpawn</b>	SPAWN_NPC	Jeep	Defines the current spawn state during play.
<b>GAME_STATE</b>	Enumeration	Playing, Paused, GameOver	Sets the values of the various Game States.
<b>currentGameState</b>	GAME_STATE	Playing	Defines the current game state during play.
<b>collectables</b>	Game Object Array	Set in Unity inspector	Game object prefabs, such as coins bars and repair tools are stored here.
<b>explodeEffect</b>	Game Object	Set in Unity inspector	Game object prefabs, to simulate an explosion effect using the particle system is stored here.
<b>smallExplosion</b>	Audio Clip	Set in Unity inspector	Audio file that will play a small explosion sound when required.
<b>bigExplosion</b>	Audio Clip	Set in Unity inspector	Audio file that will play a large explosion sound when required.

#### Private Variables

Name	Data Type	Start Value	Description
<b>playerScore</b>	Float	0	Stores the current score for this game.
<b>playerMultiplier</b>	Float	1	Stores the current multiplier and is used to calculate points added to the players score.
<b>playerPoints</b>	Float	100	Obsolete. Originally a default score value, this is

			now defined in specific game objects.
<b>invincible</b>	Boolean	False	After colliding with a damaging object, there is a short time where the player is invincible.
<b>doThisOnce</b>	Boolean	False	In the game over sequence, there is a set of variables that need to be defined only once. This Boolean enforces that constraint.
<b>playerController</b>	Player Controller		Provides quick access to the player controller class.
<b>maxHealth</b>	Float	Value retrieved from player preferences	Stores the upgraded value that is stored in the player preferences.
<b>minMultiplier</b>	Float	Value retrieved from player preferences	Stores the upgraded value that is stored in the player preferences.
<b>upgradedDamage</b>	Float	Value retrieved from player preferences	Stores the upgraded value that is stored in the player preferences.
<b>bossTimer</b>	Float	0	Defines the last time the boss was defeated or the start of the game.
<b>bossCountdown</b>	Float	120	Interval from the start of the time until the spawn state changes to indicate that the next spawn should be Boss.
<b>roadLaneArray</b>	Integer Array	-3, -1, 1, 3	Defines the X position of each road lane, excluding pavements.
<b>fullLaneArray</b>	Integer Array	-5, -3, -1, 1, 3, 5	Defines the X position of each lane, including pavements.
<b>bossLaneArray</b>	Integer Array	0	This array only has one value, but is kept as an array so it can still use of an already existing function.
<b>progressionTimer</b>	Float	0	Defines the last time the next wave update was executed or the start of the game.
<b>progressionCountdown</b>	Float	10	Interval from the start of the timer until the next wave update is executed.
<b>progressionIncrementer</b>	Float	0.8	Each next wave update increments game values by this amount.

<b>maxSpeed</b>	Constant Float	20	Defines that maximum NPC speed.
<b>scoreDisplay</b>	Text		Manipulates the text on the canvas when an update is required.
<b>multiDisplay</b>	Text		Manipulates the text on the canvas when an update is required.
<b>healthDisplay</b>	Text		Manipulates the text on the canvas when an update is required.
<b>audio</b>	Audio Source Array	Gets all attached audio source components	Several audio sources where used to prevent cutting other game sounds off.
<b>playerPrefs</b>	Player Prefs Control Script		Provides quick access to the player preferences class and all the get and set functions.