

National College of Ireland  
BSc in Computing  
Gaming & Multimedia  
2016/2017

Conor Donnelly  
x13734595  
x13734595@student.ncirl.ie

Construction Mayhem

Technical Report



# Declaration Cover Sheet for Project Submission

## SECTION 1 *Student to complete*

<b>Name:</b>   
<b>Student ID:</b>  
<b>Supervisor:</b>   

## SECTION 2 Confirmation of Authorship

*The acceptance of your work is subject to your signature on the following declaration:*

I confirm that I have read the College statement on plagiarism (summarised overleaf and printed in full in the Student Handbook) and that the work I have submitted for assessment is entirely my own work.

Signature: \_\_\_\_\_

Date: \_\_\_\_\_

NB. If it is suspected that your assignment contains the work of others falsely represented as your own, it will be referred to the College's Disciplinary Committee. Should the Committee be satisfied that plagiarism has occurred this is likely to lead to your failing the module and possibly to your being suspended or expelled from college.

# Table of Contents

Executive Summary.....	4
1 Introduction.....	5
1.1 Background.....	5
1.2 Aims.....	6
1.3 Technologies.....	6
2 System.....	8
2.1 Requirements.....	8
2.1.1 Functional requirements.....	16
2.1.2 User requirements.....	16
2.2 Design and Architecture.....	17
2.3 Implementation.....	19
2.4 Graphical User Interface (GUI) Layout.....	21
2.5 Testing.....	22
2.6 Evaluation.....	24
3 Conclusions.....	25
4 Further development or research.....	26
5 Appendix.....	27
5.1 Project Proposal.....	27
5.2 Project Plan.....	29
5.3 Monthly Journals.....	29

## **Executive Summary**

This technical report is underlining the progress I made in my project regarding technologies and logic used, and the enormous changes that occurred in my project since my Mid-Point Technical Report upload on the 11<sup>th</sup> of December 2016. In this report, I can show snippets of the finalised code and, as well as discussing the obstacles faced along the way to producing the final product. This report will highlight the architecture of the system, screenshots of GUIs that show the final product, and include testing that I have used to fine tune the project into making it more player friendly.

# 1 Introduction

Construction Mayhem is a time trial game with a destructible environment. The games aim is to gather as many bricks as possible (boosting your score) around a building site using a forklift before the time runs out, but you are also required to use your forklift to physically remove objects that are in your way of collecting bricks. Some of the obstacles are heavier than others, depending on what they are, and others have bricks hidden behind them so the main focus of the game is to use of the forklift to manoeuvre your way around the building site and collecting bricks.

## *1.1 Background*

My original plan for creating this game was to make a timeless puzzle game that required the user to drive around a building site and prevent falling bricks from hitting the ground by catching them on the back of a pickup truck. Over time, the bricks would pile up on the back of the truck and keep stacking on top of one another until either one brick hits the ground or the user flips the truck.

Many issues occurred with this original plan and almost instantly it became quite apparent that this was far too ambitious. Lining the vehicle up underneath a spawning point that would work with the assistance of an overhead camera, but over time the bricks stacks would get so high that lining it up at spawning points would make it impossible to see where the bricks are coming from. I knew then that the idea of limitless spawning was going to be too difficult, so I added a time limit and abandoned my plan to implement stacking.

My original idea didn't only change, but it downgraded. I was left with a game where you drive from point to point to collect bricks falling from the sky and over time realised that even when I began adding obstacles, I couldn't make the game any more interesting. This is when I began adding a destructible environment and got the idea of adding a forklift with a functioning lift.

With the functioning lift, you could move the items in the game I added rigid bodies onto (for the destructible environment) and this is where I got the idea of placing bricks around

the map that you had to go find before the timer ran out, as opposed to bricks spawning randomly in the sky for you to catch.

## ***1.2 Aims***

I have always aimed to jump ahead of schedule so I could begin implementing further functionalities into my game (for example multiplayer, an AI that tries to knock you off course, and hosting it online), but with the original game plan itself proving quite ambitious a lot of these further functionalities couldn't be achieved and will be discussed in further developments.

The first main aim of my finished product was to create a fully destructible environment. Not everything in the map can be moved (has a rigid body attached to it), but as planned, *almost* everything can be moved or destroyed (within reason, for example, you can't destroy the border gates to drive off the map nor can you knock down buildings that guide you to other areas on the map).

I then needed to create a surrounding environment so that the map will appear full, so that users wouldn't be looking into empty spaces surrounding the map. I achieved this by creating a surrounding mountain terrain and placing the construction overlooking the sea on the beach.

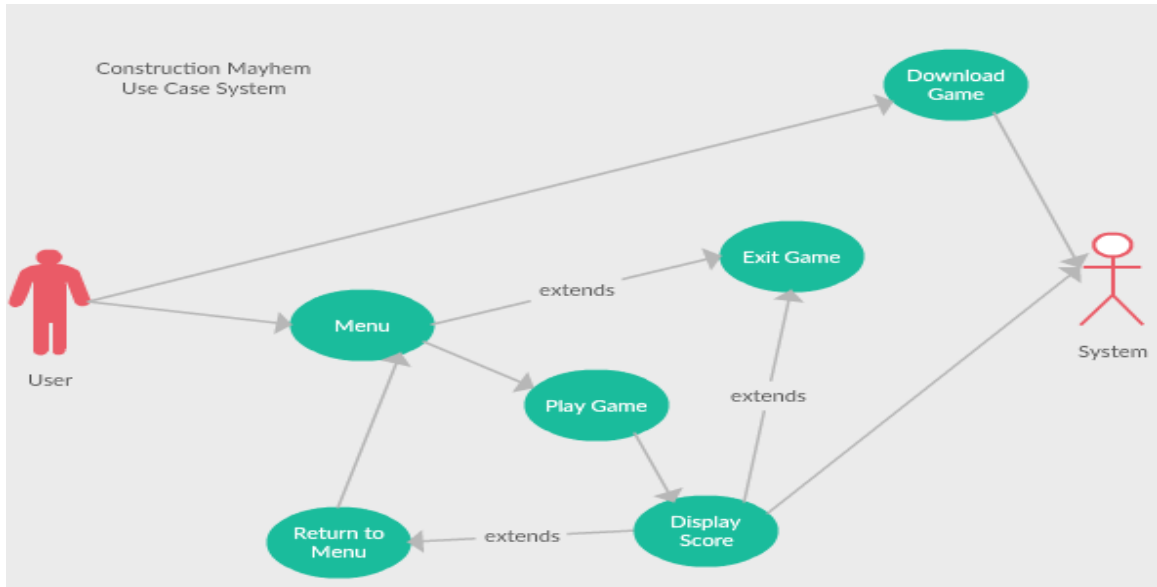
My last aim was to improve the functionality of the fork lift. The lift could be used to move objects but when I ran over bricks they added to the score board straight away. My aim was to have the bricks collectable only when the lift itself touched them, and I achieved this by adding tags onto the forklifts colliders that only the bricks could be collected when it touched them, making it more challenging and time consuming when collecting bricks.

## ***1.3 Technologies***

I used Maya for designing everything, from the map to the vehicle, and the surrounding environment including every building, structure and moveable object. The only thing on the map that I didn't personally design and texture was the water as this couldn't be imported from Maya, so I had to use the Water asset that comes with Unity itself. Unity is

the program I used for building the game itself, in Unity I could add functionality to the game by implementing code (C#), which was coded using Visual Studios. Here I coded the functionality for the vehicle, the timer and score counter, and changing the scenes from the menu to the game itself.

## 2 System



### 2.1 Requirements

- [Requirement 1 <Download Game>](#)

#### Description & Priority

This is when the <User> chooses to <Download Game> onto the <System> that they will be playing the game on.

#### Use Case – 001 Download Game

##### Scope

<Download Game> onto <System>.

##### Flow Description

##### Precondition

<User> <System> is running.

##### Activation

Use Case 001 starts when the <User> begins download.



## Main Flow

1. <User> selects .exe file to download the game.
2. <User> successfully downloads the game onto the <System>.

## Alternate Flow

### A1

1. <User> selects .exe file to download the game.
2. Not enough memory on <System>.
3. <User> makes space in the <System>.
4. <User> successfully downloads the game.

## Exceptional Flow

### E1

1. <User> selects .exe file to download the game.
2. Unknown error occurs.

## Termination

Game successfully downloads.

## Post Condition

<System> goes into a waiting state.

- [Requirement 2 <Play Game>](#)

## Description & Priority

This is how the <User> gets to access the content of the game and make use of its functionalities.

## Use Case – 002 Play Game

### Scope

Select the <Play Game> option in the <Menu>.

## Flow Description

### **Precondition**

The <Menu> is open.

### **Activation**

Use Case 002 starts when the <User> selects the <Play Game> button in the <Menu>.

### **Main Flow**

1. <User> selects the <Play Game> button in the <Menu>.
2. <User> begins to play the game.
3. When timer runs out, the game will <Display Score> and gives the <User> an option to <Return to Menu>.

### **Alternate Flow**

#### **A1**

1. <User> selects the <Play Game> button in the <Menu>.
2. <User> chooses to <Exit Game>.

### **Exceptional Flow**

#### **E1**

1. <User> selects the <Play Game> button in the <Menu>.
2. Unknown error occurs.

#### **E2**

1. <User> didn't <Download Game>.

### **Termination**

Countdown timer finishes, <Display Score> appears and an option to <Return to Menu>.

### **Post Condition**

<Menu> with option to <Play Game>.

- **Requirement 3 <Exit Game>**

### **Description & Priority**

This is how the <User> will <Exit Game> and return to the <System> they are playing it on.

### **Use Case – 003 Exit Game**

#### **Scope**

<Exit Game> and return to <System>.

#### **Flow Description**

#### **Precondition**

The <Menu> or <Play Game> is open.

#### **Activation**

Use Case 003 starts when the <User> chooses to <Exit Game> to return to the <System> they are playing on.

#### **Main Flow**

1. <User> chooses to <Exit Game>.
2. <User> returns to the <System>.

#### **Alternate Flow**

##### **A1**

1. <User> chooses to <Exit Game>.
2. The game freezes.
3. <User> returns to the <System>.

#### **Exceptional Flow**

##### **E1**

1. <User> chooses to <Exit Game>.
2. Unknown error occurs.

## **E2**

1. Game hasn't been opened.

### **Termination**

Game returns to the <System>.

### **Post Condition**

<System> goes into a waiting state.

- **Requirement 4 <Display Score>**

### **Description & Priority**

This is how the <User> sees their final score, after they <Play Game> and let the level end for the game to <Display Score>.

### **Use Case – 004 Display Score**

#### **Scope**

<Play Game> ends and game will <Display Score>.

#### **Flow Description**

#### **Precondition**

<Play Game> has been selected, and game is played until the timer has fully counted down.

#### **Activation**

Use Case 004 starts after the <Play Game> Use Case is selected, and the countdown timer has reached zero. Then the game will <Display Score>.

#### **Main Flow**

1. <User> selects the <Play Game> button in the <Menu>.
2. <User> begins to play the game.
3. When timer runs out, the game will <Display Score> and gives the <User> an option to <Return to Menu>.

## Alternate Flow

### A1

1. <User> selects the <Play Game> button in the <Menu>.
2. <User> begins to play game.
3. <User> chooses to <Exit Game>

### A2

1. <User> is in the <Menu>.
2. <User> chooses to <Exit Game>

## Exceptional Flow

### E1

1. <User> selects the <Play Game> button in the <Menu>.
2. Unknown error occurs.

### E2

1. <User> selects the <Play Game> button in the <Menu>.
2. <User> begins to play game.
3. Unknown error occurs.

### E3

1. <User> didn't <Download Game>

### E4

1. <User> hasn't opened the game yet.

## Termination

Game will <Display Score> and <User> will either <Return to Menu> or <Exit Game>.

## Post Condition

Game<Menu> or the <System> goes into a waiting state.

- **Requirement 5 <Return to Menu>**

### **Description & Priority**

This is how the <User>, after they <Play Game> and let the level end for the game to <Display Score>, select the option to <Return to Menu>

### **Use Case – 005 Return to Menu**

#### **Scope**

<Play Game> until the level is over.

#### **Flow Description**

#### **Precondition**

<User> must complete level.

#### **Activation**

Use Case 005 starts when the <User> selects <Return to Menu> after completing the level.

#### **Main Flow**

1. <User> selects the <Play Game> button in the <Menu>.
2. <User> begins to play the game.
3. When timer runs out, the game will <Display Score> and gives the <User> an option to <Return to Menu>.
4. <User> chooses to <Return to Menu>.
5. <User> has returned to the <Menu>.

#### **Alternate Flow**

##### **A1**

1. <User> selects the <Play Game> button in the <Menu>.
2. <User> begins to play the game.
3. When timer runs out, the game will <Display Score> and gives the <User> an option to <Return to Menu>.
4. Game freezes and <User> chooses to <Exit Game>, returning to <System>.

## Exceptional Flow

### E1

1. <User> selects the <Play Game> button in the <Menu>.
2. <User> begins to play the game.
3. When timer runs out, the game will <Display Score> and gives the <User> an option to <Return to Menu>.
4. Unknown error occurs.

### E2

1. <User> selects the <Play Game> button in the <Menu>.
2. Unknown error occurs.

### E3

1. <User> selects the <Play Game> button in the <Menu>.
2. <User> begins to play the game
3. Unknown error occurs.

### E4

1. <User> has not downloaded the game.

### E5

1. <User> hasn't opened the game.

## Termination

<User> chose to <Return to Menu> and is currently in the <Menu>

## Post Condition

Game is in <Menu> in a waiting state.

### **2.1.1 Functional requirements**

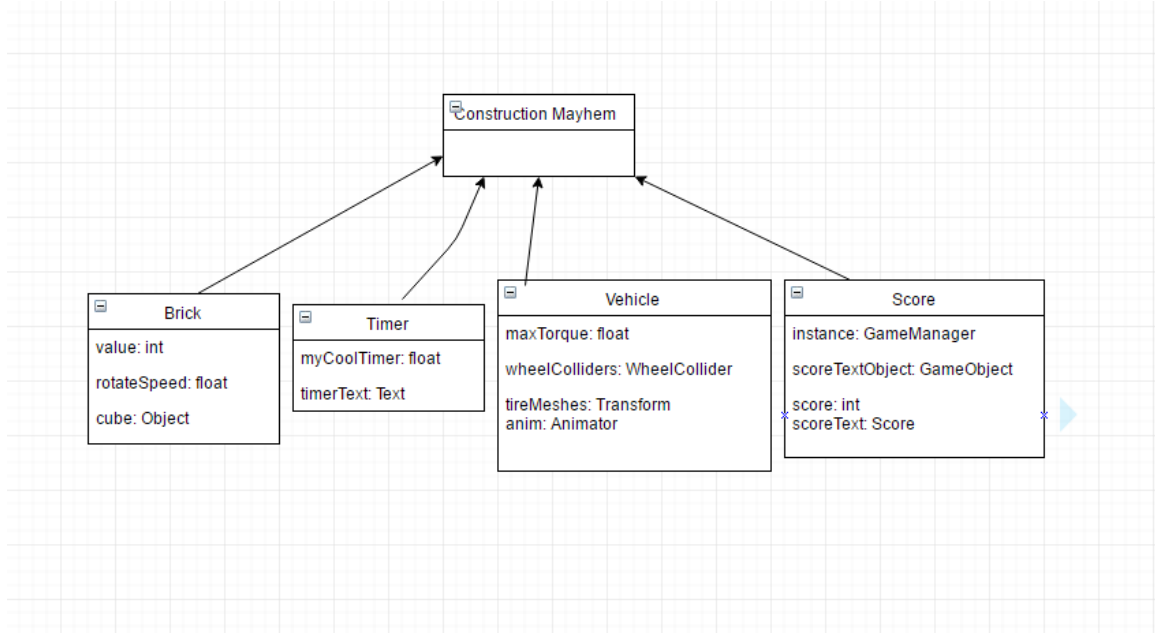
- The game must be developed and built Unity.
- The game should be designed in Maya.
- The game should be coded with C#.
- The game must be downloaded in order to use the game.
- The must be played on a laptop or desktop device.
- The game must give an option to play the game in the menu.
- The game must display the timer on the top right hand corner of the screen.
- The game must display the score on the top left hand corner of the screen.
- The game must provide a functional forklift and allow the lift to move up and down.
- When the countdown is up, the game must disable the script in the vehicle that allows it to drive, and the game must display the score and give an option to return to the menu.

### **2.1.2 User requirements**

- Users must download the game.
- They must play the game on a laptop or computer device.
- When they are in the menu, users should click on “Play Game” in order to access the content of the game.
- When users begin the level, they should be able to see the timer counting down on the top right hand corner of the screen.
- When users pick up bricks whilst operating the forklift, they should see the score on the top left hand side going up.
- When users collide with objects that have rigid bodies, they should be able to move them out of the way or lift them up using the forklifts functionalities.
- When users complete the level (when the timer runs out), they should no longer be able to control the vehicle. Also, they can select an option to return to the main menu where they can select the play the game again or choose to exit the game altogether.



## 2.2 Design and Architecture



### Brick

#### 1. value: int

This is the value that each brick is worth, and adds to your score each time you pick up a brick.

#### 2. rotateSpeed: float

This is the speed in which the bricks rotate in the map.

#### 3. cube: Object

This is the object used for creating the brick.

### Timer

#### 1. myCoolTimer: float

This is the timer that will countdown from the assigned number it is given.

#### 2. timerText: text

This is the text that will display the time on the GUI.

## Vehicle

### 1. **maxTorque: float**

This contains the speed of the vehicle.

### 2. **wheelColliders: WheelCollider**

Colliders are what keep the meshes from falling through the ground, the vehicle contains four of them which are places around the wheels, helping it drive on the ground along the mesh without falling through. When coded, this creates a field in which I assigned each wheel collider in order to control their movement.

### 3. **tireMeshes: Transform**

This creates a field where I assign each of the tire meshes into, which makes them spin when the user accelerates the car or turns the wheel.

## Score

### 1. **instance: GameManager**

When this script 'GameManager' is attached to a brick, the 'instance' is indicating that the game object it is attached to must be destroyed, which adds onto the score.

### 2. **scoreTextObject: GameObject**

This is the GameObject that is destroyed (adding onto the score).

### 3. **score: int**

This integer is where the value is added each time a GameObject is destroyed.

### 4. **scoreText: Text**

This is what displays the score on the GUI.

## 2.3 Implementation

### Collectables.cs

```
16
17 void OnTriggerEnter(Collider other) {
18     if (other.gameObject.tag == "Brick")
19     {
20         GameManager.instance.Collect(value, gameObject);
21         AudioSource source = GetComponent<AudioSource>();
22         source.Play();
23     }
24 }
25
26 }
```

Here we see the code responsible for picking up bricks. When you are operating the forklift, the collider on the lift is looking for game objects with the tag “Brick”. So if a game object with a tag “Brick” triggers the collider when entered, “GameManager.instance.Collect(value, gameObject);”, collect this instance of its’ value and gameObject.

### GameManager.cs

```
15 void Awake ()
16 {
17     if (instance == null)
18         instance = this;
19     else if (instance != null)
20         Destroy (gameObject);
21
22     scoreText = scoreTextObject.gameObject.GetComponent<Text> ();
23     scoreText.text = "Score: " + score.ToString();
24 }
25
```

Here is the code responsible for destroying the gameObject after we collect its’ value. Here we see “if the instance does not equal to null, then destroy the object”, in other words, destroy the instance of the game object that the GameManager collects in Collectables.cs. When the item is destroyed “the scoreText is the value within the destroyed gameObject, which will be display on screen as “Score:” plus the integer displayed as string”.

## myTimer.cs

```
24     void Update()
25     {
26
27         myCoolTimer -= Time.deltaTime;
28         if (myCoolTimer > 0)
29         {
30             timerText.text = myCoolTimer.ToString("f0");
31         }
32         else if (myCoolTimer < 0)
33         {
34             button.SetActive(true);
35             timerText.text = "0";
36             GameObject.Find("Vehicle").GetComponent<Car>().enabled = false;
37         }
38     }
39 }
```

This is the code that disables the car script and enables the button that allows the user to return to home page to appear, when the timer reaches zero. We see this where it says “else if myCoolTimer is less than zero, the button is set to active, the timers text will appear and zero, and that the component Car (which is the script Car.cs that enables the car to drive), found in the game object “Vehicle”, is not to be enabled.

## AnimCont.cs

```
34     void ActivateLift()
35     {
36         if (Input.GetKeyDown(KeyCode.RightShift))
37         {
38             anim.Play("Up");
39         }
40         else if (Input.GetKeyDown(KeyCode.RightControl))
41         {
42             anim.Play("Down");
43         }
44     }
45 }
46 }
```

This is the code that enables the forklift’s lift to move up and down. We see here that “if the RightShift key is pressed, then play the animation “Up”, or else, if the RightControl key is selected, play the animation “Down””.

## 2.4 Graphical User Interface (GUI) Layout

### Menu



This is the simple and self-explanatory game menu.

### Countdown Timer



Here we see what happens when the timer reaches zero. The player loses control of the car as the script is disabled, leaving their score displayed on the top right hand corner and a new option to return to the main menu.

## Destructible Environment



An example of the forklift at work, physically removing a counter top that appears to be in the way of hidden bricks.

## 2.5 Testing

### Trunk Test;

Each participant was given a copy of Construction Mayhem, and asked the following questions.

#### Participant 1

- 1. Locate the games title from the menu:** When asked, participant spotted Construction Mayhem title instantly.
- 2. Play the game:** Participant clicked the “Play Game” button instantly spotting it, as it is the only option present on the menu.
- 3. Drive the car:** Participant was quite familiar with the “w,a,s,d” desktop controls, and navigated the car perfectly.
- 4. Move an object out of the way using the lift, the buttons for this are Right shit to move it up, and Right Control to move it down:** User drove in front of a table and struggled at first to line the forklift up with it, but when he had it aligned he had no problem picking it up and moving around.

5. **Collect a brick:** At first user didn't know where the bricks were as the prototype doesn't have many placed around it, but when he spotted them rotating they picked one up straight away.

## Participant 2

1. **Locate the games title from the menu:** When asked, participant spotted Construction Mayhem title instantly.
2. **Play the game:** Participant clicked the "Play Game" button instantly spotting it, as it is the only option present on the menu.
3. **Drive the car:** Participant was not familiar with the "w,a,s,d" controls but used the directional buttons instead.
4. **Move an object out of the way using the lift, the buttons for this are Right shit to move it up, and Right Control to move it down:** User couldn't get the hand of the lift at first but after driving through objects to move them began seeing the benefits of using the lift to maneuver them out of the way as this made getting around objects much easier. Although they complained that the objects seemed to be "weightless".
5. **Collect a brick:** The participant spotted the bricks earlier while driving the car around but struggled lining the bricks up with the fork, although after many attempts did get the hang of it quite easily.

## Participant 3

1. **Locate the games title from the menu:** When asked, participant spotted Construction Mayhem title instantly.
2. **Play the game:** Participant clicked the "Construction Mayhem" title before realizing there was a button present on the menu.
3. **Drive the car:** Participant was familiar with the "w,a,s,d" controls, and began driving the car quite perfectly.
4. **Move an object out of the way using the lift, the buttons for this are Right shit to move it up, and Right Control to move it down:** At first had difficulty

getting used to lining the lift up with object to move them but got the hang of it quickly.

5. **Collect a brick:** The participant drove straight up to a brick and picked it up with no issues, but other bricks they had difficulty lining the forklift up with.

## ***2.6 Evaluation***

The three participants who took part in the Trunk Test gave me very positive feedback regarding the gameplay although there were a few complaints that helped me make the game more user friendly. Every participant brought up the issue of no instructions on how to operate the forklift. Most PC Gamers would be familiar with the basic “w,a,s,d” controls to move around but when it came to operating the forklift in the game the participants would have liked more indication on what buttons to use to operate it as they played the game.

Although moving objects around to clear a path becomes quite easy to do once you get the hang of it, the bricks are quite hard to spot. This was an issue raised by all three participants and one I acted on immediately as this is the main function in the game that contributes to the final score. I have fixed this by making the bricks larger and made them spin faster than they previously were, which leaves them much easier to spot amongst the other objects.

One complaint I got by participant two was that the objects they could move around were weightless, and that a chair, for example, shouldn't be as easy to move than an entire water filter I had placed beside the swimming pool. I fixed this issue by editing the mass of the objects, giving bar stools and pint glasses much lower masses than tables, and beer barrels.



### **3 Conclusions**

Creating Construction Mayhem and building it was a major task and at some points proved quite ambitious which forced me to take many steps back. A major downfall I had was putting most of my time into planning and designing rather than functionality, as this left out a lot of exciting aspects such as multiplayer and an enemy AI. With this in mind, the original and ambitious game plan actually opened up dozens of other doors to very cool and exciting gaming ideas, leaving me with a lot of content to work with, which ultimately led to the development of the final product.

While developing the final product I left a lot of the code till last, in order to build the map around the code and functionality, as opposed to the other way around. The reason I did this was to give the game more of an interactive environment as I aimed for almost everything to be destructible. Although this added to the environment of the game, by the time I got down the functionality and code it really took a lot longer than I expected and left me with very little time to finalize the product, leaving me to forget the most basic features such as instructions.

The issues that I had been left with during the final stages of development were quickly tackled when I began testing. The first issue raised was the lack of instructions which I tackled instantly, but what was overlooked entirely was that there was no difference in “weight” (or mass) between each of the items, that one participant I had tested brought to my attention which, once I fixed, gave the gaming a more realistic feeling (for example, pint glasses previously weighed the same as the tables they were placed on).

## **4 Further development or research**

One aspect in the game which I could have added would have been an enemy AI. One idea I had was a truck or a cement mixer that would drive after you and try to sabotage your mission or even destroy the bricks you aimed to collect. Another idea would have been a wrecking ball that could have locked onto you when you stayed in an area for too long (perhaps trying to line the forklift up to collect a brick) and come crashing down on you knocking you off course.

Multiplayer was a quite ambitious idea but would have been very doable. Two players could have raced against time to try and beat each-others score before the timer ran out or both players may have been able to work together on a team against an enemy AI to collect as many bricks as possible. I could have designed a level where one player is the enemy AI and your aim is to destroy more bricks than the other player can collect.

The entire point of the game is to collect bricks around a building site and this isn't just a random setting I came up with. One functionality I abandoned was spawning buildings and that was going to be in the final development of my project where the user was required to collect a certain amount of bricks and spawn buildings around the building site. The score count would only show the user that they have enough bricks to spawn a building (which would reset after each building is spawned) and the main aim of the game was to build the entire structure (which was supposed to be a hotel resort on a beach) before the timer ran out. Your calculated score would then be the amount of time it took you to complete this task, and this score would then have been stored in a database showing user names and high scores that would have been accessible in the main menu (another ambitious development I had to abandon).

I also wanted to host the game online to make it more accessible for users to download and although this isn't too difficult a task, I decided to put every bit of time I had left in making the map as enjoyable and user friendly as possible.

## 5 Appendix

### 5.1 *Project Proposal*

#### Objectives

Blocks are falling from surrounding buildings on a construction site and you're the driver of a truck. Your mission is to endure the weather conditions, bumpy terrain and increasing speed of the blocks to successfully stack them on the back of the truck to prevent them from breaking, and construct a building in the meantime on the back of the truck (so your blocks must be stacked on top of one another to successfully win the mission).

There will be three levels to beat, and as the time progresses through each level the blocks will fall at a faster pace and the environment will change to make it more difficult for the user to complete the tasks (such as rain, snow, wind changes and explosions of some kind).

#### Background

I wanted to design a challenging 2D stacker game originally but decided against this because similar games exist, such as Tetris. I then pondered the idea of a 3rd person stacker game but ran into the dilemma of overkill, it was too simple an idea for the complex coding and designing required which could also lead to a lot of empty space being designed. So then I came up with the idea that instead of you controlling falling objects, it would be the user controlling the platform with the goal of reaching a certain height, as opposed to the typical stacking game routine that limits the heights of the objects stacking.

To add significance to my game I came up with the idea of the platform being the back of a truck, and the map being a building site. I done this to give the objectives sense, so you're not just mindlessly stacking objects to win. The idea is also significant because it allows me to develop on making the game more challenging, so that not only are objects

just increasing in speed but weather conditions and bumpy terrains can affect the outcome of the game as well.

## Technical Approach and Details

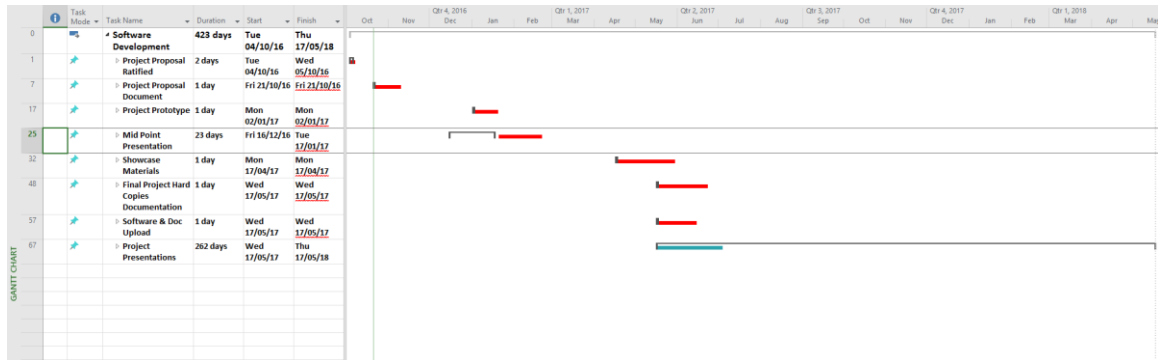
I will be using Unity for developing the game, Blender for developing the objects and vehicles, and the language I will be coding in is C#. So, the truck and the different building parts that will be falling are going to be designed in Blender. When I import these into Unity, this is where I will design the layout of the map, and implement code that allows the objects to fall on a timer (and increase speed over time).

I will be coding in the trucks driving abilities, score points that recognize the objects falling and when the game begins and ends, there will be a timer that controls the speed of the objects falling also. My game will also require two cameras (following the truck, and one birds eye view) to help align the truck with the falling objects and this can be implemented in Unity. Over time I will surely run into more technical issues that need to be added, removed, or amended.

## Evaluation

- I will have to test the trucks steering and capabilities of moving around the terrain (colliders will be implemented for this testing, so the truck recognizes when it bumps off other objects).
- Tests will have to be done on the pieces falling as well. If the pieces are allowed hit the ground, the level should fail and the game should restart, but if the building is complete, the game should recognize this and move onto the next level.
- The speed of the blocks falling, the weather conditions, and the bumpy terrains that get increasingly more difficult must also be tested. The environment changing should affect the difficulty of the game play BUT it shouldn't make the levels impossible.

## 5.2 Project Plan



## 5.3 Monthly Journals

### My Achievements;

#### 8<sup>th</sup> of October 2016

This month, I was able to research development software and the current gaming market place to decide my project I pitched to 3 lecturers on the 5<sup>th</sup> of September. They judged my project positively and gave me excellent feedback to work on over the next few weeks before I begin and finish my project proposal.

#### 4<sup>th</sup> of November 2016

I researched and attained the relevant software I will be using for the development of my project

#### 9<sup>th</sup> of December 2017

I have created the GUI (Menu) for my project with all the relevant options to select, and have begun the development of my car.

#### 6<sup>th</sup> of January 2017

I have created a GUI and connected this to the scene where you can play a prototype of my game. In this scene, you drive a car to three different check points (under a time limit), when you reach these check points bricks will fall from the sky. As you collect each brick this is added to the score that is presented on the top left hand side of the screen.

### ***10<sup>th</sup> of February 2017***

Developed the track my vehicle will be driving around, and developing code to spawn bricks when reaching a checkpoint.

### ***10<sup>th</sup> of March 2017***

Designing my track, levels, code, and trying to improve the graphics.

### ***7<sup>th</sup> of April 2017***

Finished all the code required for the full functionality of the game. I have created a Main Menu, and a destructible environment that the user should take advantage of to get to objects in the game that adds to the score, and created .exe files of the game ready for testing.

## **My Reflection;**

### ***8<sup>th</sup> of October 2016***

I spent very little time on researching engines to begin building my game on and put too much time into researching a target audience. Although it was the target audience that helped me focus on delivering a good pitch, I have no left myself with little time to begin testing out different engines I will be building my game on. This isn't too much of a problem, as I am familiar with Blender and Unity (from previous projects), and I do enjoy trying out new software.

### ***4<sup>th</sup> of November 2016***

I have done a few tutorials on software I am using but very little time on developing items for my game, this has cost me time but as I am familiar with the programs this won't be an issue for further development.

### ***9<sup>th</sup> of December 2016***

I spent a lot of time developing the car model, this was because every time I imported it, the trailer on the back of the car wasn't visible in Unity due to how thick the walls were. This has cut into the technicality of my project but I am currently working on that to finish before I upload my technical report on Sunday, the 11<sup>th</sup> of December.

### *6<sup>th</sup> of January 2017*

After my Mid-Point presentation, and during the development of my prototype I have set a much clearer path on what I want to do with my project. The beginning idea was a bit too ambitious but this realization has opened up plenty of doors to work on the functionality of my game, which has been a major benefit rather than a disadvantage.

### *10<sup>th</sup> of February 2017*

Designing my track is taking much longer than anticipated but I have planned for these drawbacks and have time to get my code completed which should be finished entirely by next month, when the next journal is due.

### *10<sup>th</sup> of March 2017*

The design of the track has taken longer than predicted, and my code wasn't as perfect as I predicted but I left time for myself if these drawbacks occurred.

### *7<sup>th</sup> of April 2017*

Designing the game took up a lot more time than I intended but I predicted this would happen as the technology I was using to create the map was new (Maya), but very similar to older technologies I have used in the past such as Blender. The idea and functionality of my game changed dramatically since pitching the idea, which was far too ambitious to begin with. The more I coded the more ideas came to mind making the game more interesting (such as adding a destructible environment).

## **Intended Changes;**

### *8<sup>th</sup> of October 2016*

The lecturers have given me very good feedback and constructive criticism over my pitch, and I now intend on including some of their ideas into my project proposal and hopefully, the game itself.

### *4<sup>th</sup> of November 2016*

I will begin the development of a prototype of my project for showcasing in December, and filling in my Requirements Specification Proposal

### ***9<sup>th</sup> of December 2016***

I should have a running car, that will collect a block and add a point to the games score. I should also have the game connected to the GUI so that you can run the game by selecting the Play option in the menu.

### ***6<sup>th</sup> of January 2017***

I will build a track and a map and fix the functionality of the game to get one level working. I need to implement the code I have done to get the bricks to spawn as triggers as well as falling when the timer runs out.

When the basics are done, creating the other levels and difficulties, as well as a database to keep the high score, shouldn't be an issue. Also, I may be able to work on more ambitious developments of my game expressed in the Evolution of my uploaded documents.

### ***10<sup>th</sup> of February 2017***

Finishing my code and first track so I can begin the working on vehicle and level selections (difficulty).

### ***10<sup>th</sup> of March 2017***

Finish the designing so I can focus on the code and functionality of my game.

### ***7<sup>th</sup> of April 2017***

Make changes to the game after testing is completed.

## **Supervisor Meetings;**

**Date of Meeting:** 26<sup>th</sup> October 2016

*Items discussed:* Each person attending the meeting gave an overview of their project and Frances Sheridan (our supervisor) requested we each send her a copy of our project Proposals for her to look over.



**Date of Meeting:** 28<sup>th</sup> November 2016

*Items discussed:* My progress and if I needed any assistance.