**Group Submission – Front Sheet**

**Level 2 Group Project**

Component Number: 3

Programme Title: Games Computing

Group: GC3

Supervisor: Andy Cowe

The following table shows the agreed allocation of workload against this assignment submission. The table has been checked and all members in our group as identified in the Group List presented on Blackboard, are shown correctly in this table. The percentage values provided have been agreed and total 100%.

|  |  |  |  |
| --- | --- | --- | --- |
| Enrolment  Number | Surname | Forename | Agreed Percentage Workload |
| 10237560 | Bowes | Adam |  |
| 12266105 | Fisher | Thomas |  |
| 11207513 | Pearson | Darryl |  |
| 12281943 | Graves | Joshua |  |

Contents

[Introduction 3](#_Toc386476702)

[Requirements 3](#_Toc386476703)

[Tools and research 3](#_Toc386476704)

[Project planning and management 3](#_Toc386476705)

[Artefact Implementation 3](#_Toc386476706)

[Testing Strategy 3](#_Toc386476707)

# Introduction

For this assignment the team needed to produce an artefact which would be a serious game with the intent of informing the player of the challenges facing the conservation of marine environments. The game needed to have a focus on action whilst remaining casual and appealing to a wide audience. Also the game could be built on PC but needed to be suitable for a direct port on to a mobile platform. The team chose to make a game that put the player in charge of managing a beach with the intention of showing them how litter left on beaches can affect marine life. This was done with the intention of showing the player the way they impact marine life from a first-hand perspective as most people at some point have been to a beach. With this game the team intended to put the player in a management role to see how individual buildings on a beach can contribute to pollution. As well as this, the team wanted to show how people visiting the beach caused pollution and what measures could be taken to reduce the effect.

# Requirements

## World

The game world needs to be a bounded beach area on which the player can begin to develop in to a popular destination. The beach will be affected by litter which will be created by visitors to the beach and the beach will graphically deteriorate to reflect this, Also the NPC’s will react to the litter by leaving the beach.

One of the changes made from the original concept was the piers. The team chose to remove the piers on the grounds that they added little to gameplay and did nothing to enhance the message.

## Objects

The game has a wide range of objects that allow the beach to take shape. The buildings that are available are as follows: lifeguard stations which increase the number of people allowed on the beach, fisheries which create income based on how clean the beach is but generate waste in the process, hotels which increase the number of people who come to the beach and ice-cream shops, gift shops and clubs which NPC’s visit and gave money but also cause the NPC’s to have litter which they may drop.

Some of the changes made involve the removal of some aesthetic items. These were removed due to the way the map is created. As the beach can be made any size the team would need to develop way to seed the map with these objects which would be time consuming

## Goals

The goals of the game are to make the beach a popular place to visit by building different structure to attract and then meet the needs of the NPC’s, at the same time the player will need to manage the pollution generated by the visitors which decreases the popularity of the beach. The player needs to manage the beach in a sustainable way to ensure the both the environment and the visitors are happy despite them being in conflict.

# Tools and research

## Unity

For this project the team chose to use unity to develop the game. Unity has gained a large amount of popularity and has also has a large supporting community which made problem solving and dealing with bugs very easy in most case. Unity has proven to be a very versatile tool for making games, the built in PhysX allowed for quick and optimised responses to collisions and the visual scene editor makes debugging and scene building very simple. The ability to edit code as the game is running is very useful also as it means new features can be tested quickly without waiting.

## GitHub

The team chose GitHub to compile the groups work. This was done so that changes could be kept track of and so that everyone would be up to date when working on the project. GitHub has been very useful allowing changes to be viewed quickly by other team members. It also prevented anyone working on an old version of the game which would have meant that their code might not work with the latest version.

GitHub has a few problems which come from working with Unity. Due to the fact that a team member would need to run Unity in order to see any changes made to code, Unity would often change its metadata and mono would update the project files which means when merging an old version of the project with a new one some conflicts emerge which can sometimes cause problems

//Josh’s software

# Project planning and management

Due to a large workload from other projects, it was necessary to have a dynamic and stable way to communicate with team. To this end the team used a Facebook group. The group allowed us to have a space where members could report on their progress and tasks could be given. As a result there was a large amount of communication as the team members asked each other for assets and code to be finished. Facebook also has a time stamp on its posts meaning when things were said and assigned can be reviewed. Using Facebook however had a few disadvantages, due to the nature that the tasks were assigned as they became apparent the critical path was not well tracked and instances occurred when team members had to wait for things to be finished by others. Another way the project was managed was with GitHub. GitHub stores the time and amount contributed of each commit to the project. This meant that progress of team members could be tracked and the tasks completed could be monitored.

# Artefact Implementation

//what we did plus challenges

One of the first challenges faced was creating the map. The map needed to have changeable rows and columns while also allowing buildings to be built on individual tiles. To do this the tiles were created in a nested loop which created and moved the tiles depending on the rows, columns and the tile size. By having this array, the tile objects can be accessed and by having the rows, columns and tile size as public they can be accessed from other scripts.

Another challenge was creating a set of NPC’s that could navigate a dynamic and changing world while still being simple enough to run on a phone. To tackle this the NPC’s were given enlarged capsule colliders and the buildings were also given capsule colliders so that when an NPC walked in to a building the colliders would prevent them from walking through a building and would instead get pushed around meaning that the NPC’s give the impression that they walk around each other and obstacles without putting a large load on the CPU.

The NPCs were a main part of the gameplay and needed to go to shops to generate money as well as react to the environment so players could see the effect it had. To do this the NPCs were given a set of needs, these needs had a random number assigned from a range representative to the need in question (ice-cream took between 10 and 100 seconds while buying a gift took between 60 and 100 seconds), this meant that the price shops sold at might not represent how much money is made adding an extra layer of complexity. Amongst the NPC needs there was also a need to get rid of litter acquired from shops. If this need was not filled by a bin in time then they would drop the litter. To enhance the focus on the effects of pollution, the NPCs also have a tolerance for litter. This means that if a beach is too littered then they will make a verbal comment on the state of the beach and then leave, this brings the issue to the players’ attention whilst having a negative impact on the income from the beach.

Building placement had to be implemented which meant that a message had to be sent to a building control object from the GUI control object telling it what building to make. As the buildings had a type and a size the data had to be passed by an array as unity does not allow multiple parameters to be sent to a function using the “sendmessage” function. As well as this, a ray had to be passed from the camera and through anything that wasn’t a tile. This was achieved by using tags which allowed the ray to identify different objects and so only react to tiles. This also allowed the buildings to be shown on the tiles before they were actually placed

# Testing Strategy