PG-Simpson, Callum

Project report

Contents

[Overview, including asset listing and additional libraries 2](#_Toc89875318)

[An overview of the project 2](#_Toc89875319)

[A complete listing of all assets 3](#_Toc89875320)

[Meshes 3](#_Toc89875321)

[Images 3](#_Toc89875322)

[Textures 3](#_Toc89875323)

[Sound 4](#_Toc89875324)

[Software used 4](#_Toc89875325)

[A complete listing of any additional libraries added to the project. 4](#_Toc89875326)

[A description of features 4](#_Toc89875327)

[Basic Game Modelling 4](#_Toc89875328)

[Final game level intro-screen with a listing of keyboard/mouse controls 4](#_Toc89875329)

[Three primitive-based game objects 5](#_Toc89875330)

[Skybox changed 5](#_Toc89875331)

[Terrain textures 6](#_Toc89875332)

[Audio 6](#_Toc89875333)

[Heads Up Display 6](#_Toc89875334)

[Camera, Meshes, Lighting, and FX 7](#_Toc89875335)

[Final camera motion technique 7](#_Toc89875336)

[Mesh-based objects 7](#_Toc89875337)

[Lighting 8](#_Toc89875338)

[Special effects 8](#_Toc89875339)

[Physics, AI, and Gameplay 9](#_Toc89875340)

[Use of game physics 9](#_Toc89875341)

[Non-player characters (NPCs) and artificial intelligence (AI). 10](#_Toc89875342)

[Gameplay elements in line with the game theme 11](#_Toc89875343)

[Discussion section reflecting on the project. 11](#_Toc89875344)

[Strengths 11](#_Toc89875345)

[Weaknesses 12](#_Toc89875346)

[What would be required to expand the project into a complete game? 12](#_Toc89875347)

[Closing thoughts 12](#_Toc89875348)

# Overview, including asset listing and additional libraries

## An overview of the project

The title for my game is Monkey mode. Monkey mode is a 3D platformer with a jungle adventure theme.

The game takes place in the middle of an abandoned castle that has become heavily overgrown. Since the abandonment of the castle, a troop of monkeys now patrol the area with the desire for the giant banana. Gone rogue, the main character has taken it upon themselves to find the legendary giant banana. This is the game’s primary objective; if the giant banana is reached then the game is won(Giant banana located on top of vertical moving platform).

Most of the game is spent jumping between platforms or avoiding enemies. The player spawns in the middle of the castle and will need to find the giant banana by exploring the map. On the ground, multiple angry monkeys will try to prevent you from getting to the giant banana. The enemies will prevent you from getting to the objective by coming up to you and dealing damage or by using a projectile to knock you over. Therefore, the player must be careful in traversing the map as they will need to avoid the angry monkeys.

Throughout the game, the player will come across different items and platforms. There are two types of platforms, static platforms and moving platforms. The moving platforms will either move left to right or up and down. This adds a new mechanic of timing to the game as the player must jump with correct timing to stay on the platform and not fall.

Other gameplay elements come from the different items in the game. Each of these items will have different effects. The first item in the game is the banana. When picking up the banana item the player's health is increased by 50 points. The second item in the game is the melon. When the player picks up the melon item, the fog in the scene is removed and the yellow platforms become powered and start to move. To indicate this a cut scene displays the electricity special effect being applied to the moving platform. This item is important and must be obtained to get to the objective of the level. If the player falls off of a platform, they take 50 points of damage. If the player's health becomes less than zero, the player dies. Additionally, the game uses a timer that kills the player when it reaches zero.

As for the player's movement, the player must press WAD to make the player move. After inputting the W key, the monkey runs forward and using A and D allows the player to turn left or right. If the player wishes to move more slowly, they can hold the left shift button. This will allow for more precise movement throughout the map as this will be important when traversing between platforms.

To jump between platforms the play will need to press space. If the player lands on top of the platform they will land on it. Conversely, if the player hits the side they will bounce off and fall.

To give the game multiple levels of difficulty, 2 modes were added. The first mode is the easy mode where the player has increased health and has more time to complete the level.

Alternatively, the player can select hard mode which will have a more limited time, and if the player falls once, they die.

## A complete listing of all assets

### Meshes

* Monkey model from: <https://open3dmodel.com/3d-models/3d-model-animated-chimpanzee-rig_43177.html>

Date Downloaded: 01/12/2021

License for use: free for individual/educational use

Prepared for rigging in blender

Rigged using Mixamo

* Sign, Dirt Platforms, Star, Moving platforms from: <https://creazilla.com/nodes/1403702-platformer-pack-3d-model>

Date Downloaded: 01/12/2021

* License for use: free for individual/educational use
* Watchtower from: <https://free3d.com/3d-model/guard-tower-304393.html>

Date Downloaded: 01/12/2021

License for use: free for individual/educational use

* Banana model from: <https://www.turbosquid.com/3d-models/banana-cartoon-3d-model-1495166>

Date Downloaded: 01/12/2021

License for use: free for individual/educational use

* Crate model: <https://www.turbosquid.com/3d-models/3d-wooden-box-model-1304417>

Date Downloaded: 01/12/2021

License for use: free for individual/educational use

Castle model: <https://free3d.com/3d-model/castle-walls-37690.html>

Date Downloaded: 01/12/2021

License for use: free for individual/educational use

* Projectile model: Lab Sheet Model with previous banana model texture added

### Images

* Menu background from <https://opengameart.org/content/the-war-over-a-melon-field-main-menu-background>

Date Downloaded: 01/12/2021

License for use: free for individual/educational use

Modified using paint

### Textures

* Skybox texture from: <http://www.vwall.it/wp-content/plugins/canvasio3dpro/inc/resource/cubeMaps/>

Date Downloaded: 01/12/2021

License for use: free for individual/educational use

* Grass texture from: <https://assetstore.unity.com/packages/2d/textures-materials/glass/stylized-grass-texture-153153>

Date Downloaded: 01/12/2021

License for use: free for individual/educational use

* Pickup texture from: <https://www.textures.com/download/Fruit0017/20858>

Date Downloaded: 01/12/2021

License for use: free for individual/educational use

* Primitive texture from: <https://opengameart.org/content/wood-aged-seamless-handcrafted-texture-holz-dark-seamlessjpg>

Date Downloaded: 01/12/2021

License for use: free for individual/educational use

### Sound

* Rainforest track from: [https://orangefreesounds.com/rainforest-sounds/#](https://orangefreesounds.com/rainforest-sounds/)

Date Downloaded: 07/12/2021

License for use: free for individual/educational use

* Monkey sound effect from: https://mixkit.co/free-sound-effects/monkey/

Date Downloaded: 07/12/2021

License for use: free for individual/educational use

* Monkey sound effect from: https://mixkit.co/free-sound-effects/eat/

Date Downloaded: 07/12/2021

License for use: free for individual/educational use

### Software used

* Mixamo used to rig monkey model
* Blender used to modify models
* Paint was used to modify the main menu and create die/dmg BMP files.
* Gimp used to cut skybox into multiple files

## A complete listing of any additional libraries added to the project.

No additional libraries have been added.

# A description of features

## Basic Game Modelling

### Final game level intro-screen with a listing of keyboard/mouse controls

A picture containing qr code

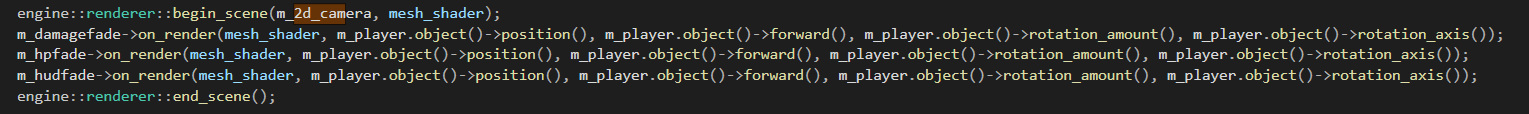
Description automatically generated

This is the intro screen for my game. To make it I found a free to use background and added a title and controls using paint. As for implementation, the menu has its own class, this renders the image I had previously created as a quad. Additionally, a on event method was used to indicate if the player had pressed 1 to or 2 to start the game which sets the start of game Boolean to false.

Text

Description automatically generated

To render the menu correctly, I made sure to switch to a 2D camera before rendering the menu.



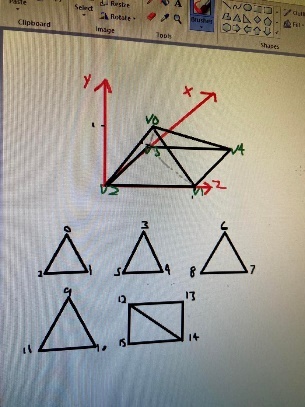
The controls for this game are very simple and are included in the intro screen. WAD allows the player to move forward, left, and right. Pressing space allows the player to jump and holding left shift makes the player walk slowly.

### Three primitive-based game objects

For all my primitive objects I first planned them using paint as shown below.

Diagram

Description automatically generatedA picture containing text, whiteboard

Description automatically generated

My three primitive based objects were the isosceles triangular prism, square pyramid, and triangular prism. These were each implemented through their own class and then instantiated in the example layer constructor. All three objects have been textured and positioned into place.

Additionally, to make sure the light was reflected correctly, cross-product was used. This allowed me to calculate the normal using the vectors of the edges for each face.

### Skybox changed

The skybox in the game has been changed to something more appropriate for the theme (although not very visible due to the surrounding castle). This was created by cutting a skybox into the 6 different images using Gimp and editing the terrain code in the example layer constructor appropriately.

Text

Description automatically generated A picture containing text, screenshot

Description automatically generated

### Terrain textures

Based on the feedback given in milestone 1, I decided to repeat the texture in my game. This was achieved by editing the texture coordinates in the cuboid class. Additionally, I changed the texture to a higher quality one as the old texture still looked blurry even after it was getting repeated. This gave me a really good-looking result without any changes in performance.

A person walking through a field of green grass

Description automatically generated with low confidence

### Audio

The first audio played in my game is a Rainforest ambient sounds track; this is to match the theme of the level. This is played when the game starts.

The second sound played is a monkey sound effect. This is played from the AI monkey when the player has picked up the banana object.

The third sound effect is played when the player eats the mushroom powerup. This plays an eating sound effect.

The last sound effect is played by the player's monkey upon entering hard mode.

### Heads Up Display



For my HUD I used a combination of text with graphics. To do this I rendered multiple quads while using the 2d camera and then rendered text on top.

The first element the HUD displays is the timer. This indicates the time the player has left. This is important as if the time reaches 0 the player will die.

Secondly, the HUD displays the health of the player as a health bar. The size of this bar represents how much health the player has left.

To implement this feature, I rendered a quad mesh and changed its size each time there was a change in HP. This can be seen below.

Text

Description automatically generated

## Camera, Meshes, Lighting, and FX

### Final camera motion technique

The first camera technique I have implemented in the game was a 3rd person camera. This puts the camera behind the player and follows them as they move. The camera rotates around the player as the player moves left or right. To implement this an update camera method has been added to the player class.

The second camera technique I decided to use was an on-rails camera. This camera is used when the player gets the pickup that activates the moving platforms. Here the moving rail camera acts like a cutscene, displaying the platforms starting to move. To implement this, I added a variable called m\_play\_cutscene. This is set to false by default and is set to true once the player gets the pickup. This then enables the camera through the code displayed below.

Text

Description automatically generated

### Mesh-based objects

There are many mesh-based objects in the scene. Most notably are all the platforms. They are all rotated, translated, and scaled into the correct position to allow the player to jump between them successfully. The different types of platforms used are dirt platform, moving platform and crate. Another example of mesh-based objects in the game is the items such as the banana and mushroom. All the mesh-based objects were either already textured or had their texture applied by me using blender. Some examples of mesh-based objects can be seen below.

  A picture containing floor, indoor, bedroom

Description automatically generated

### Lighting

For the three lights required in the scene, I used a point light, spotlight, and directional light.

The point light in the game is the lighting type I chose to move. The way the light moves is by following the player from above and sends out green lights all around.

The spotlight is placed on top of the star object to give it the illusion of giving out light. This shines white light in front of it indicating where the pickup is

The directional light has been set to a red light which in combination with the green point light creates a really interesting effect.

Some of the lighting effects in combination can be seen below.

A picture containing text, green

Description automatically generatedText

Description automatically generated

### Special effects

There are three different techniques for special effects in my game. The first technique occurs when the player is damaged. This adds a red tint effect to the entire screen. To achieve this, I used a new class called damage fade which overlays a quad of a red image set to transparent over the player’s screen.

A screenshot of a video game

Description automatically generated with medium confidence

The second technique I used was fog, this makes the scene feel fuller and adds a low visibility effect. I used this technique in my game as it makes it more difficult to find the correct path to the final goal, while also providing cover for the AI enemies.

The last effect I used was an electricity effect. This is applied when the moving platforms are activated. The electricity effect implies that the moving platforms are powered and can now be used.

A picture containing grass, plant, vegetable

Description automatically generated

## Physics, AI, and Gameplay

### Use of game physics

The Physics in my game was achieved through a combination of bullet and custom physics.

When moving the player, gravity is applied to the players up speed. This gives the player more natural movement.



Physics have also been applied to the enemy’s projectile. To do this, bullet physics were used alongside a projectile class. This enables accurate motion of the ball regarding their velocity, acceleration, gravity, friction and rolling friction.

A ladybug on a leaf

Description automatically generated with medium confidence

Furthermore, my game is heavily based on collision detection as it is a platformer. To achieve collision detection, I used Axis-Aligned Bounding boxes. This was a good choice for my game as the platforms would all be axis-aligned naturally. Each platform in the game has two collision boxes (see image below).

A picture containing green, container, box

Description automatically generated

The main collision is used to detect if the player is colliding with the side of the platform. If this is true, then the player is bounced back away from the platform. Alternatively, the top of the box has its own collision as this allows the game to detect if the player is coming from above. If the player's feet box collides with the top collision box on the crate, then the player can walk around on the box. Overall, this produces a nice effect where the player can jump from platform to platform as long as they land correctly.

### Non-player characters (NPCs) and artificial intelligence (AI).

The 5 enemies in this game are angry monkeys that want to prevent the player from getting to the giant banana.

A group of people walking in a foggy field

Description automatically generated with low confidence

As for intelligence, the NPC’s use a state machine to decide what action to take next. An NPC can do multiple actions, each represented by one of these six states: idle, on guard, running, shooting, jumping, chasing. The relationships between these states are displayed in the diagram below.

Diagram

Description automatically generated

To shoot the player, the NPCS throw a projectile using bullet physics. If this hits the player, it knocks them back. As displayed by the diagram above, the monkeys will generally try to avoid the player and only attack if they get really close. This deals 50 points of damage repeatedly. If the player is not in range, then the monkey will become idle and randomly switch between jumping or patrolling. The reason I added randomness to the NPCS was to make the movement seem more natural. This was achieved through the code below.

A screenshot of a computer

Description automatically generated with low confidence

### Gameplay elements in line with the game theme

One of my gameplay elements is powerups.

The first powerup which gives an advantage to the player is the mushroom. This allows the player to jump higher while also increasing the fall distance needed to take fall damage. When the player walks over the powerup, animation and sound are played to indicate the item has been consumed and the item disappears.

The second powerup on my game is the banana. Similar to the mushroom, when the player walks over the banana, it is consumed. This item increases the player's life by 50 points, which is just enough to allow the player to survive one more fall.

The last powerup is the melon. When the player picks this item up a cutscene is played using an on rails cam and also removes the fog. Additionally, this powerup powers the moving platforms with electricity so they start to move.



My second gameplay element is a timer. Depending on the difficulty level, the player either has 1 or 3 minutes to complete the level. If the timer reaches 0, the player dies and has to start over. This adds challenge to the game as it limits the time the player has to complete the level. The time left in the game is displayed through the HUD.

## Discussion section reflecting on the project.

### Strengths

The strength of this game is the platforming. Overall, I’m very pleased with how the jumping from platform to platform plays. Having the idea to split platforms into two collision boxes rather than one was a real turning point for my project and enables a fun gameplay experience.

Additionally, I am very pleased with how the main character turned out. I especially enjoy how well the multiple animations alongside rigged models work in combination with realistic physics-based movement. Getting the animations to work simultaneously with the movement was very tricky so it feels rewarding to see the player move around smoothly.

### Weaknesses

A weakness of the game is the level design. Given more time I would have loved to make the level more complex by adding many more platforms.

Another weakness is my code optimisation. For the most part, my game uses correct use of classes and logical design, but I still feel that there are some places where I could have added functions into classes or created some new methods to avoid some repetition. It was only later in the course as my skills developed that I realised: instead of repeating classes I could have optimised one class (such as damage fade and main menu). If given a full release I would go back and change this.

### What would be required to expand the project into a complete game?

To expand the project into a complete game I would need to develop multiple levels with more complex designs. Additionally, I would need to iron out some bugs within the game such as the character bouncing back when not intended. To do this heavy testing would need to be completed.

Moreover, to enable easier level design, a simpler way of creating mesh-based objects would need to be added as currently placing new objects in the level can be quite a task.

### Closing thoughts

Looking back on this project I am really happy with how far I’ve come and the experience I’ve gained. It is my first-time using C++, so it has been very useful for developing my programming skills. Additionally, as this module has such a detailed insight into the heart of a game engine, it has greatly developed my understanding of game programming as a whole and has prepared me with skills that will be useful throughout my career.