Coursework

Games Programming 2 - M3I625656-19-A

Ruari McGhee – S1432402

# Main Game

## Class Description

Handles running the game loop

## Class Components

### Setting Up Game

* Setting up displays
* Setting up Game Managers

### Running Game

# Display

## Class Description

## Class Components

### Setting Up Display Devices

### Double Buffer Rendering

# Game Object

## Class Description

A base from which a game entity is built from by adding different components

## Class Components

### Constructor

Gameobjects require a reference to the scene that they are in so that if a component of the gameobject needs other information from that scene then it has a path to do so.

### Component Handling

Each gameobject uses a component pattern to handle adding additional behaviour to the gameobject

#### Adding & Getting Components

* Method adding and getting components use generic types

#### Updating Components

* Each GameObject stores it’s added components in a vector
* This makes it easy for the gameobject to iterate through it’s added components and call the update method on each component

### GameObject Implementations

Subclassing from the GameObject class allows the user to create premade GameObjects for GameScenes to implement

The subclassed gameobject uses its constructor to set up any components that the gameobject may need.

# Transform

## Class Description

Stores the object’s position and rotation values, also allows for the gameobject to have a parent which allows the gameobject to inherit its parent’s transform values

## Class Components

### Position

#### Local Position

#### Global Position

* If the gameobject has a parent, then the object’s new local position is calculated by getting the parent’s global position
* This is done manually so that the order of multiplication can be controlled (z then y then x)
* If the gameobject has no parent, then the object’s local position will be the same as its global position (the gameobject’s parent could be considered as being the gamescene’s origin)

### Rotation

#### Euler Rotation

The rotation of the object in radians

#### Quaternion Rotation

Manually calculated from the relevant Euler rotation value

### Parenting

When an object’s parent is changed the class recalculates the object’s position and rotation values so that they are relative to its new parent

# Game Scene

## Class Description

A container for holding the multiple gameobjects required to build a level for the game.

## Class Components

### Scene GameObjects

Each scene contains a vector of GameObjects

When creating a new gameobject it must be added to this vector so that the scene can correctly update it throughout the scene’s lifetime

### Handling Scene Information

Some GameObjects need to access information from other gameobject’s in the scene

An example of this is when the physic’s handle class needs to know all the box collider’s in the scene

# Scene Manager

## Class Description

Implements a singleton pattern to ensure only one instance on the class exists in the project. For this reason the constructor is also kept private

## Class Components

### Game’s Current Scene

Since the game can only be in one scene at a time

### Changing Scene

Any class that has a reference to the game’s scene manager can change the game’s current scene. To keep those classes from needing to know about the possible GameScene implementations this is handled by the changeScene method only needing a string to determine what scene to move to.

# Audio Manager, Font Manager, Mesh Manager, Shader Manager, Texture Manager

## Class Description

Each manager handles the storage of a type of loaded items, they also ensure that items are only loaded once to optimize memory usage.

Like the SceneManager these manager’s implement a singleton pattern to ensure that only one instance of each class exists in the project.

## Class Components

### Storing Loaded Items

Each manager has a map to store loaded items, when an object requests to load a new item from resources the managers first checks to see if the item has already been loaded.

If not, then the manager loads the item and then stores it in the map by using its file path as the key.

# Audio

## Class Description

## Class Components

### Setting Up Audio Devices

### Loading Audio Files

### Playing Sound

# Mesh

## Class Description

Handles loading a model file

## Class Components

### Loading A Model

### Changes to Obj\_Loader

# Shader

## Class Description

The Shader class handles loading shaders from the resources folder to create a shader program

## Class Components

### Loading a Shader From Resources

Since a shader is made from a fragment shader and a vertex shader the parameter used for loading the shader file ignores the file type.

### Shader Lighting

The fragment shader handles the lighting required on the object.

Lighting is split into 3 part;

Ambient light – a universal color applied to the object

Diffuse Light –

Specular Light -

# Texture

## Class Description

## Class Components

### Loading A Texture From Resources

# Time

## Class Description

Handles calculating the difference in time between game frames

## Class Components

### Calculating Delta Time

### Information Accessibility

Delta time is kept as a static variable however is kept private in the time class however to prevent other classes from changing its value it is kept private within the class. For other classes to use its value it has a static getter method.

# Input Manager

## Class Description

Handles detection and consuming user inputs

## Class Components

### Input Detection

SDL only has the ability to detect when keys are pressed down and when keys are unpressed, to handle keys that are being held down the manager holds this info in a map.

This map uses SDL’s keycodes as keys and Booleans as values to represent if a key is currently being held down.

Every frame the manager consumes the currently pending events and updates its map of inputs depending on the event. If the

### Information Accessibility

Other classes can check if the user has pressed a specific key down using the input manager’s isKeyPressed method

# Component

## Class Description

Components are classes designed to be added to gameobjects to store information or enable behaviour

## Class Components

### Constructor

Let’s the component know what gameobject that it is attached to

### Update

Since the Update method is intended to contain additional behaviour for subclassed components to define the base update method is defined as a pure virtual method.

This ensures that even if the subclassed component doesn’t need to define additional behaviour to call every frame it must at least define its own version of the Update method.

# Ball Controller

Inherits from - Component

## Class Description

Components are classes designed to be added to gameobjects to store information or enable behaviour

## Class Components

### Constructor

Lets the component know what gameobject that it is attached to

### Update

Called every frame

# Box Collider

Inherits from - Component

## Class Description

Components are classes designed to be added to gameobjects to store information or enable behaviour

## Class Components

### Constructor

Lets the component know what gameobject that it is attached to

### Update

# Camera

Inherits from - Component

## Class Description

Components are classes designed to be added to gameobjects to store information or enable behaviour

## Class Components

### Constructor

Lets the component know what gameobject that it is attached to

### Update

Called every frame

# Camera Mount

Inherits from - Component

## Class Description

## Class Components

### Constructor

Let’s the component know what gameobject that it is attached to

### Update

Called every frame

# Environment Controller

Inherits from - Component

## Class Description

Components are classes designed to be added to gameobjects to store information or enable behaviour

## Class Components

### Environment Movement

Called every frame

# Physics Handler

Inherits from - Component

## Class Description

A component that when added to a gameobject allows the gameobject to move with forces and react to other colliders in its scene

## Class Components

### Velocity

A measurement of the translational forces currently affecting the object

* Velocity Calculations

### Torque

A measurement of the rotational forces currently affecting the object

Uses similar calculations to handling velocity but the y axis only has air resistance affecting it

### Collision Detection

Before the object’s new position is applied the class checks if the object is colliding from anything

#### Broad Phase – Sphere vs Sphere Collision

First the class checks for

#### Narrow Phase - Box Vs Sphere Collision

### Collision Resolution

#### Reflecting Velocity

#### Moving Object Outside of collider

#### Torque Addition

# Renderer

Inherits from - Component

## Class Description

Component for handling rendering a gameobject, requires a mesh, texture and shader to render

## Class Components

### Renderer Setup

### Rendering Object

# Sphere Collider

Inherits from - Component

## Class Description

A container for storing a sphere collider’s information

## Class Components

### Collider Details

Only needs to contain a