Class: Save

Responsibilities	Collaborations
• Saves vital game state information into an xml hierarchy • Outputs xml and stores locally	GameWorld

Class: Load

Responsibilities	Collaborations
• Loads all data from the xml save file • Constructs appropriate objects • Outputs a Data object storing all information	GameWorld Data EntityFactory Bug Commit FlashDrive Laptop ReadMe SwipeCard

Class: Data

Responsibilities	Collaborations
• Stores all game data.	

Class: Camera

Responsibilities	Collaborations
• Represents the player’s current view of the game world	

Class: Entity

Responsibilities	Collaborations
• Represents any object in the game world • An entity has a model, and several positional elements • Can be anything from a tree to a flash drive	Camera Player

Class: Light

Responsibilities	Collaborations
Represents a light source which is either: • The sun, a constant source of light, regardless of distance, or • An attenuating light source, one which fades with distance	GameWorld

Class: Bug

Responsibilities	Collaborations
• Represents the bug in the game • The Bug can only be interacted with	GameWorld NonPlayerCharacter

Class: Commit

Responsibilities	Collaborations
• Represents the commit item in the game • Used to defeat the bug and win the game	GameWorld Item

Class: FlashDrive	
Responsibilities	Collaborations
• Represents the flash drive item in the game • Stores an attribute which diversifies it from other laptop items	LaptopItem

Class: Item	
Responsibilities	Collaborations
• Represents the items • Stores the logic of how the players can pick up/interact with the items in the game	GameWorld MovableEntity

Class: Laptop	
Responsibilities	Collaborations
• Represents the laptop item in the game which is found in the office area • Can have code cloned off it	GameWorld Item

Class: LaptopItem

Responsibilities	Collaborations
• Represents an item in the game that can be stored in the player’s inventory • Deals with the logic involved when items in the player’s inventory are interacted with	GameWorld Item

Class: MovableEntity

Responsibilities	Collaborations
• Represents an entity in the game • Can change position and can be interacted with	GameWorld Entity

Class: NonPlayerCharacter

Responsibilities	Collaborations
• Represents a non-player character in the game • Currently, this class only applies to the bug	MovableEntity

Class: Player

Responsibilities	Collaborations
• Represents a player in the game with their associated camera.	Camera Terrain Entity GameWorld AudioController DisplayManager MovableEntity

Class: ReadMe

Responsibilities	Collaborations
• Represents a ReadMe item in the game • Takes the form of a tablet in the office area • Stores an attribute which diversifies it from other laptop items	LaptopItem

Class: SwipeCard

Responsibilities	Collaborations
• Represents a swipe card in the game • Handles the game logic that is related to the player interacting with the swipe card • Used to unlock laptops	GameWorld Item

Class: BoundingBox

Responsibilities	Collaborations
• Implementation of a bounding box • Used for checking the collision with entities supporting collision	

Class: CollidableEntity

Responsibilities	Collaborations
• Implementation of an entity that can be collided with • As of the current game implementation this class has little purpose.	StaticEntity

Class: OfficeEntity

Responsibilities	Collaborations
• Implementation of an entity that can be collided with • Implements a scaled collision hit box.	CollidableEntity

Class: StaticEntity

Responsibilities	Collaborations
• Represents a static entity which is an entity in the game world that cannot be moved • May potentially need to be checked for collision.	Entity

Class: EntityFactory

Responsibilities	Collaborations
• Entity factory which abstracts the creation of an entity • Loads the entity map for a given terrain or randomly generates entities for testing	Terrain EntityModel MovableEntity

Class: GuiFactory

Responsibilities	Collaborations
• Factory for creating the GUI components • Loads commonly used images • Constructs new GUI textures for items	GuiTexture Loader Inventory SwipeCard

Class: LightFactory

Responsibilities	Collaborations
• Factory class to handle the creation of light sources in the game • A light source is either the sun, a constant source of light, regardless of distance, or • An attenuating light source, one which fades with distance.	

Class: PlayerFactory

Responsibilities	Collaborations
• Factory class to handle the creation of Players in the game • Handles both the main player and other multiplayer players	GameWorld Player Data Camera

Class: TerrainFactory

Responsibilities	Collaborations
• Terrain factory with the responsibility of generating new terrains	GameWorld Player Data Camera

Class: EntityModel

Responsibilities	Collaborations
• Represents a complete entity model • Used by the renderer	Loader

Class: ModelData

Responsibilities	Collaborations
• Data class containing all the information about a wavefront obj model	BoundingBox

Class: ClientController

Responsibilities	Collaborations
• Initialises connection to the Server that is hosting the game • Creates a player depending on server response. • Sets an update when the GameController sends one	GameController Client

Class: ServerController

Responsibilities	Collaborations
• Initialises the Server player to be ready for incoming connection from other Clients. • Listens to the given port then creates a new Thread for that player	GameController NetworkHandler Server

Class: Client

Responsibilities	Collaborations
• Communicates to the Server thread on the server • Sends the players location • Sends any updates the Client Player has • Receives all the updates from the Server	GameController NetworkHandler

Class: Server

Responsibilities	Collaborations
• Communicates to one of the Clients it has been assigned to and constantly sends all of the Player's locations and their updates. • Listens to the Client and updates the Servers data storage	GameController NetworkHandler

Class: NetworkHandler

Responsibilities	Collaborations
• Listens to any updates from either the Server or Client Thread. • Updates the Player's gameWorld accordingly	GameWorld MovableEntity

Class: ActionController

Responsibilities	Collaborations
• Listens to the Keyboard and the Mouse and responds to the keypress events and delegates the appropriate command to the GameController. • Checks to make sure the Game isn't over	GameController GameWorld AudioController

Class: AudioController

Responsibilities	Collaborations
• Contains static methods that can be called at any time. • Plays a given audio clip	

Class: GameController

Responsibilities	Collaborations
• The main game loop which calls all of the respective renderers and constantly checks that the game should continue due to other factors changing it. • Delegates to the GameController with any game updates.	GameWorld MasterRenderer GuiRenderer ClientController ServerController ActionController AudioController

Class: TimeController

Responsibilities	Collaborations
• Handles the lighting and the shading timing in the game. • Updates the Players view	GameWorld MasterRenderer

Class: ShaderProgram

Responsibilities	Collaborations
• A shader class that delegates between the Java content and the open gl glsl shader. • Delegates between the GLSL library to calculate the colours at a given point	

Class: EntityShader

Responsibilities	Collaborations
• A shader class that handles the open GL bindings for the entity shader program. • Delegates between the GLSL library to calculate the colours at a given point	ShaderProgram

Class: GuiShader

Responsibilities	Collaborations
• A shader class that handles the open GL bindings for the Gui shader program. • Renders all of the GUI elements seen. • Delegates between the GLSL library to calculate the colours at a given point	ShaderProgram

Class: SkyboxShader

Responsibilities	Collaborations
• A shader class that handles the open GL bindings for the skybox shader program. • Delegates beween the GLSL library to calculate the colours at a given point • Renders all of the skybox in the sky	ShaderProgram

Class: TerrainShader

Responsibilities	Collaborations
• A shader class that handles the open GL bindings for the terrain shader program. • Delegates beween the GLSL library to calculate the colours at a given point • Renders all of the terrain on the ground	ShaderProgram

Class: Terrain

Responsibilities	Collaborations
• The terrain is an area of a location. • A terrain depends on the blend map and height map to know how to style the terrain and is made up of a texture pack.	

Class: GuiTexture	
Responsibilities	Collaborations
• A 2d image texture used to render the in the gui. For example the inventory or a pop up message when other events occur	

Class: ModelTexture	
Responsibilities	Collaborations
• Represents a Texture to be used on an obj model and is a java representation of that model	

Class: TerrainTexture	
Responsibilities	Collaborations
• A reference class to an open GL VBA binding	

Class: TerrainTexturePack	
Responsibilities	Collaborations
• Wrapper class to hold all of the background, red, green and blue textures	TerrainTexture

<table>
<tr><td colspan="2">Class: TextureData</td></tr>
<tr><td>Responsibilities
• Wrapper class to hold a textures data - a buffer of all of the texture information</td><td>Collaborations</td></tr>
</table>

<table>
<tr><td colspan="2">Class: Loader</td></tr>
<tr><td>Responsibilities
• Loader class to handler the loading of the wavefront obj files into OPENGL VAO and VBO.
• Uses slickutil library to aid loading</td><td>Collaborations</td></tr>
</table>

<table>
<tr><td colspan="2">Class: Maths</td></tr>
<tr><td>Responsibilities
• Contains static methods that helps the RenderEngine and other classes to compute mathematical calculations</td><td>Collaborations</td></tr>
</table>

<table>
<tr><td colspan="2">Class: OBJFileLoader</td></tr>
<tr><td>Responsibilities
• An OBJ file loader that takes an .obj model and parses it into a vertex object, containing all of the vertices and textures of that model</td><td>Collaborations</td></tr>
</table>

Class: Vertex

Responsibilities	Collaborations
• Vertex class containing all the information about a given vertex - the position, the textures and the length	

Class: DisplayManager

Responsibilities	Collaborations
• Display management class to handle the resolution and the FPS of the game. • Updates the resolution and handles the closing of the display	

Class: HelpScreen

Responsibilities	Collaborations
• A class for displaying the Help screen. • Makes it's own GuiRenderer to render it's own elements	GuiRenderer GuiFactory DisplayManager

Class: PlayLoadHelpScreen

Responsibilities	Collaborations
• Displays the Play Load Help screen and deals with simple key press logic as the action controller is not yet created. • Creates the GameController when the correct option is taken	GuiRenderer DisplayManager GameController

Class: TitleScreen

Responsibilities	Collaborations
• Initial drop page when the game is first started. • Handles the logic for the key presses as the action controller is not created yet.	AudioController DisplayManager GuiRenderer GuiTexture PlayLoadHelpScreen

Class: EntityRenderer

Responsibilities	Collaborations
• Delegate renderer to handler the rendering entities. • Has the render method that is called in the game loop that renders this Entity	EntityShader

Class: GuiRenderer

Responsibilities	Collaborations
• A simple class that only renders 2D images given the list of guiImages	GuiShader

Class: MasterRenderer

Responsibilities	Collaborations
• Master renderer handles all the delegation to the individual render class. • This is in the gameController where is does all the rendering in the game	EntityShader EntityRenderer TerrainRenderer TerrainShader SkyboxRenderer Camera Light Entity Terrain

Class: SkyboxRenderer

Responsibilities	Collaborations
• Delegate renderer to handle the rendering of the world's skybox • A skybox is a cube of png images wrapped around the player's position in the gameWorld	SkyboxShader

Class: TerrainRenderer

Responsibilities	Collaborations
• Delegate renderer to handler the rendering of the terrains • A terrain acts as a surface of the game world and has altered vertical y.	TerrainShader TerrainTexturePack

Class: GameWorld

Responsibilities	Collaborations
• Initialises the game • Delegates creation of entities, terrain, light and gui images to the relevant factory • Stores game state • Handles movement of items from inventory to actual game world • Updates entities accordingly if updates have been sent by server • Updates game state changes triggered by timed events (e.g. night/day cycle)	EntityFactory TerrainFactory LightFactory GuiFactory PlayerFactory GuiTexture GuiMessages Terrain Player Light GameController Inventory Data Commit LaptopItem SwipeCard AudioController