**Part 2 – Main Loop & UML Diagram**

**Pack Mules**

Initially, start the game up and load all the assets from the assets folder and assign them correctly to a manager. This includes grabbing the audio files, textures file, and the map information itself.

Once all the game asset files have been loaded, start the game state itself. This part will process which screen we are on.

A while loop will start – this is the main game loop and will continue to run until the game is closed.

**ProcessInput** is used to record player input, whether it be tapping on something, hovering the mouse, or striking the ball. The command is then sent to **UpdateEngineState** in either a List data structure, which interprets the command and makes the necessary changes, going down the list until everything has been done.

**UpdateEngineState** is where the main bulk of the game will be done. This function first checks what screen we are on (Main Menu, Options, Game).

If we are in the Game state, it also calculates all the moving objects, whether the player can hit the ball, the scoreboard, the course selection, and more. The **UpdateGameState** will be called from the **UpdateEngineState**. This will call functions such as ‘bool isPlayerTurn()’, ‘bool isBallMoving’ while in the Game state and process movement (velocity) of all objects.

Similarly, **UpdateEngineState** will call its respective update function depending on which state it is in (**UpdateGameState**, **UpdateMainMenuState**, etc.)

Game mainGame(){

GetAssets()

while(!mainGame){

ProcessInput();

UpdateEngineState();

Draw();

}

}

**UML Diagram**

