Practical Gaming 2024

# Name of Student…. Rokas Balzaravicius

# T Number …. T00225541

# Name of Project …. Forest Of The Living Dead

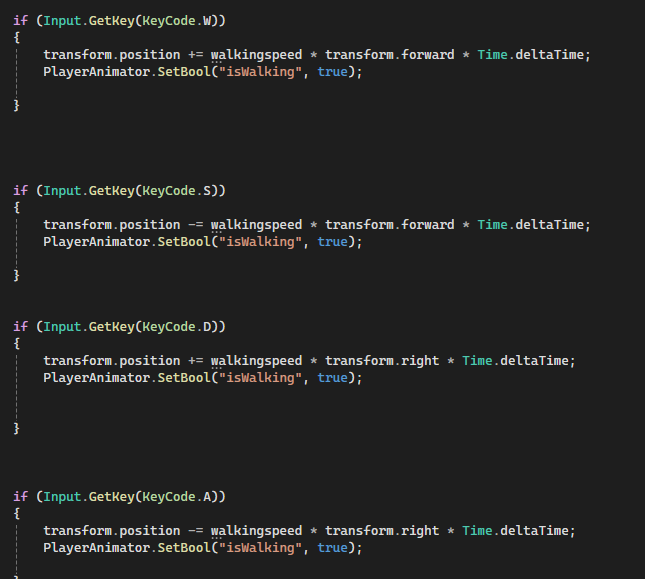
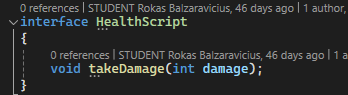
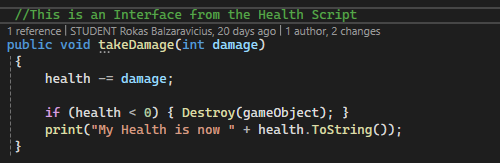
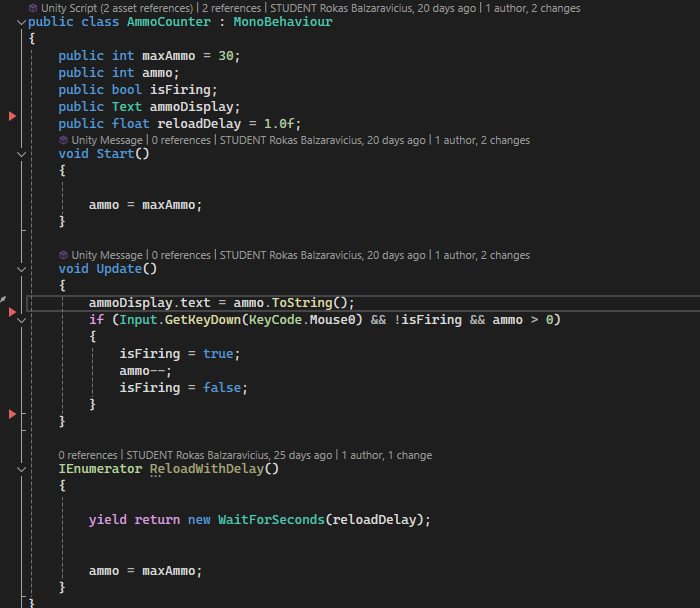
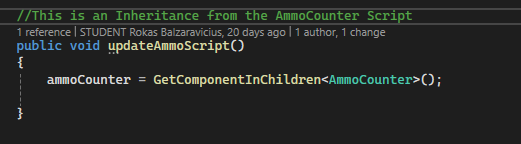
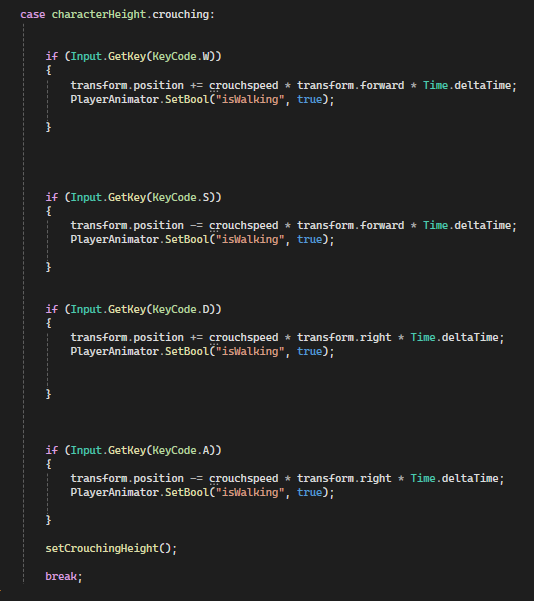
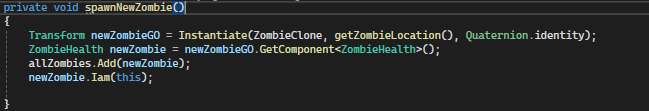
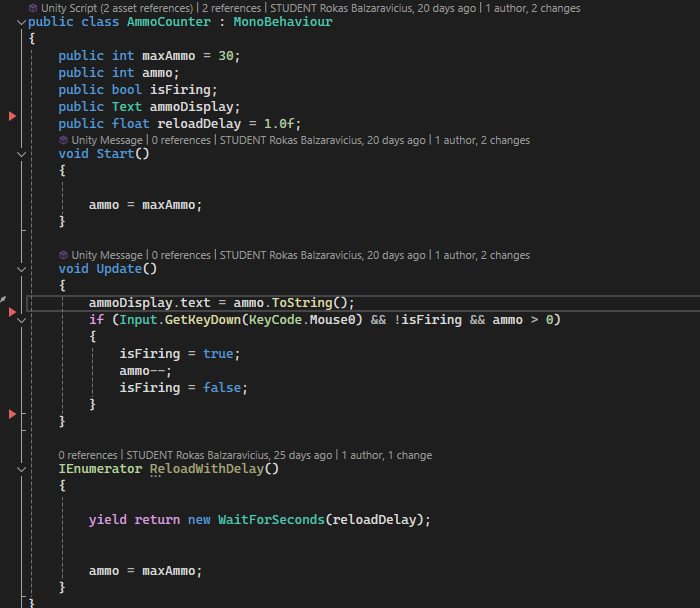
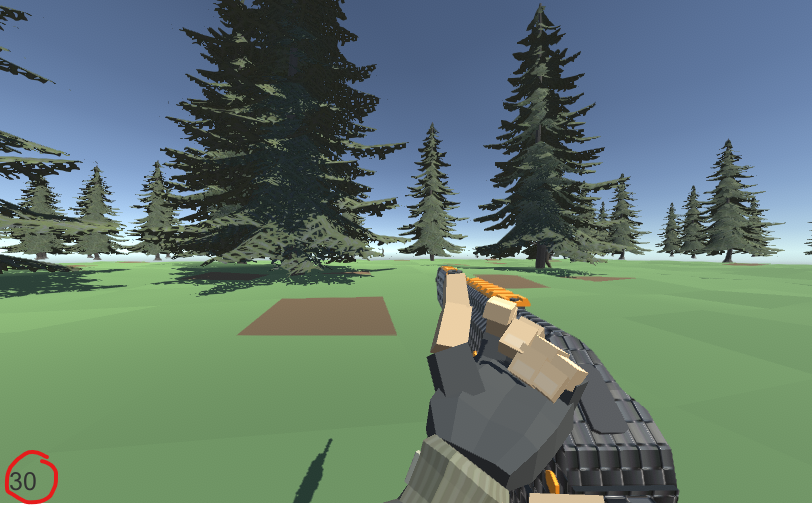
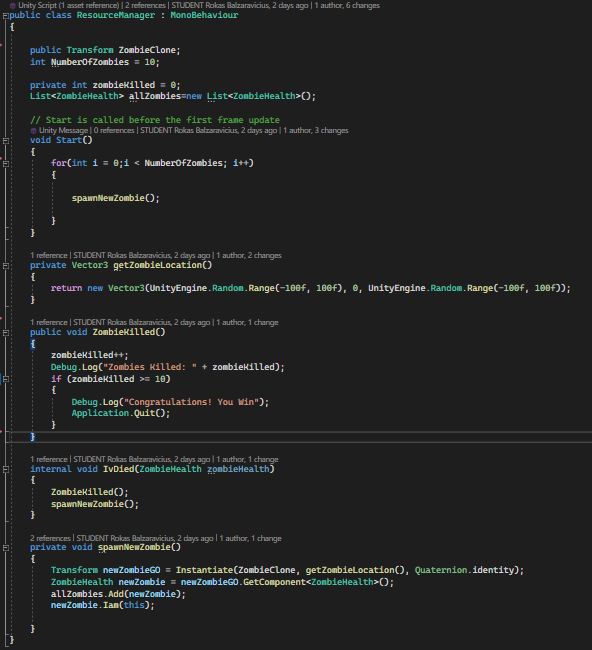
# Gameplay

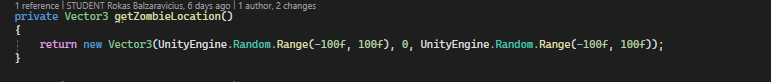
Describe how to play the game here, specify keys/mouse etc. what needs to be done to unlock further features etc.. i.e. a walkthrough which covers all of what is to be seen to be marked.

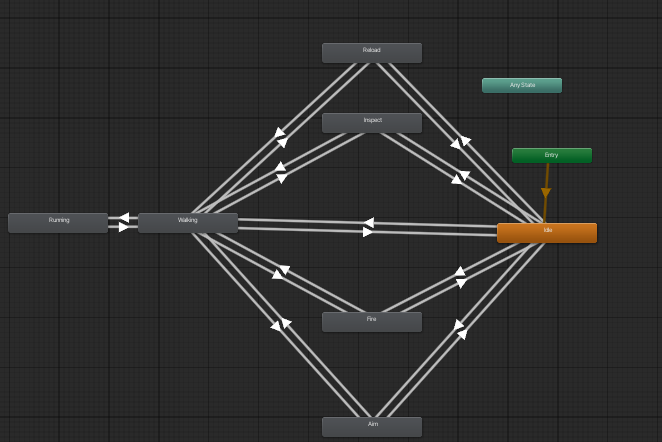
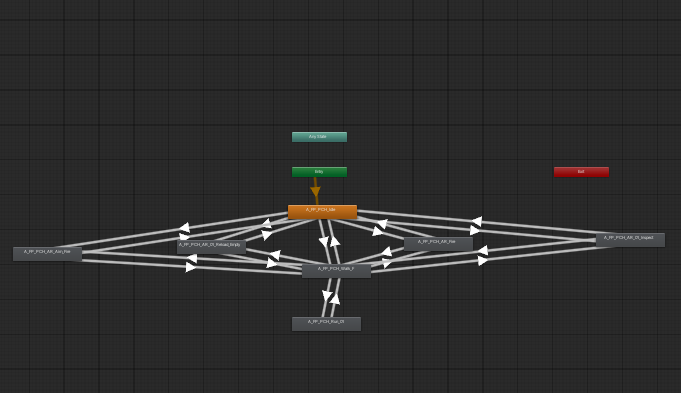
**To play this game you use the W, A, S, D to move the character around. To look around you use the mouse. Sprinting is done by pressing the left shift key. Shooting is done by left click and aim is done with holding the right click. You can inspect the weapon while holding the F key. You can reload the gun while pressing and holding the R key. To crouch in the game, you press and hold left ctrl key. Switching Weapons can be done by using the numbers 1 and 2.**

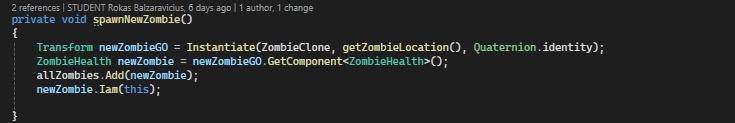
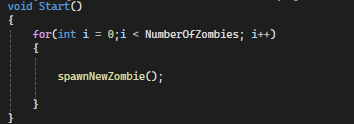
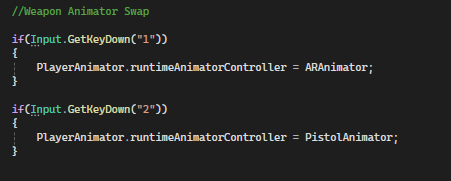
# Coding

Under each of the following headings, please describe the concept, why is it or isn’t it useful/needed, where do you implement in your project, you may provide screenshots or cut and past code segments etc..

* **Frame Rate Independence**
  + Frame Rate Independence is when objects in your game move based on some criteria other than which rendering frame you are on.
  + Frame Rate Independence is very useful as it can make your game look smoother, more realistic and more immersive.
  + I Implanted Frame Rate Independence in my Character movement script to make my character move
  + 
* **Interfaces**
  + An Interface will define method and property names that anything using them is required to have.
  + An interface is useful as it allows a user to carry out a task within the game world either through direct input or through an action.
  + I implemented interface in my health script which affects my character and the zombies in the game.
  + 
  + 
  + 
* **Inheritance**
  + Inheritance allows you to create new classes that inherit the attributes and behaviours of existing classes and modify or extend them as needed.
  + Inheritance is useful as the ability to reuse code from the base class in derived classes which avoids duplication and redundancy.
  + I implemented Inheritance by making an ammo counter script which will also be used in player movement script so I used inheritance to not include repetition.
  + 
  + 
* **Case pattern**
  + A case pattern is a selection statement. When the switch executes code of one of the conditions based on a pattern match with the specified match expression.
  + A case pattern is very useful because it makes the design process cleaner and more efficient.
  + I implemented a case pattern in my player movement script for the function to crouch.
  + 
  + 
* **Observer Pattern**
  + The observer pattern is a design pattern that allows objects to be notified of changes to the state of another object.
  + An observer pattern is very useful as it supports the principle of loose coupling between objects that interact with each other.
  + I implemented an observer patter in Zombie Health script.
  + 
  + 
  + 
  + 
* **Polymorphism**
  + Polymorphism is like giving different objects the power to perform the same action in their own unique way.
  + Polymorphism is very useful as it allows for a single variable name to be used to store variables of multiple data types such as int, float and double.
  + I have no implemented polymorphism in my code.
* **Communication between scripts/game objects**
  + Having communication between scripts is essential if you want to access functions or components between game objects.
  + This is very useful as it is needed to references to items in the game and need to pass data around with efficiency and speed.
* **Instantiation and Prefabs**
  + Instantiation is used to create a new instance of a prefab of a game object.
  + Instantiation is useful as it makes a copy of your prefab and allows less process for the application.
  + I implemented this is the Manager script as I Instantiate the zombies in the game
  + 
* **Magic Numbers** 
  + A magic number is a number with no variable attach to it
  + A magic number is not useful as if it needs to changed you would have to go through the whole code and change every single one.
  + I have implemented magic numbers in my code for random range in the zombie location.



* **Model Animation**
  + Model animation is using Animations
  + Model animation is useful as it shows movement of the character and doesn’t just stay static in a t-pose position.
  + I have two animators for the character as I have two different weapon types and the animations for them are different.
  + 
  + 
* **Self-made models and or animations**
* **Interactions between objects/scripts**
* **Propper code placement**
* **Code repetition**
  + Code repetition is when identical code appears multiple times in code.

  + 
  + 
* **Animator Swap**
  + Animator Swap is when the character switches between two animators.
  + This is useful as the swap makes that the two weapons in the game have their own animations.
  + I have implemented this in my game in Character movement script.
  + 
* **Feature 2**
* **Feature 3**