Project for Gesture Based UI Development

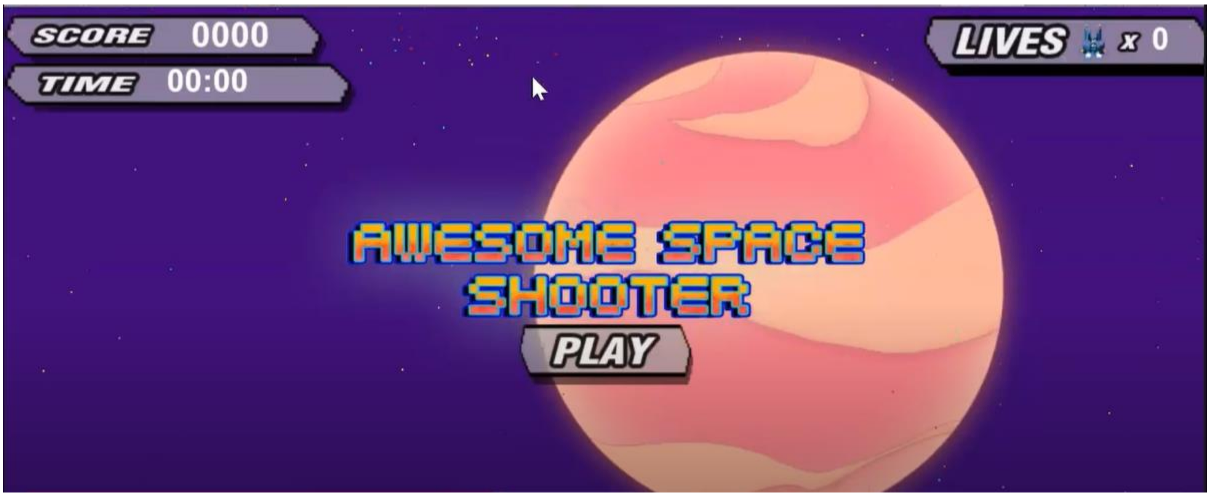
Documentation

G00337350

Ryan Higgins

SCREENCAST LINK - https://www.youtube.com/watch?v=V2Dvpeq88sU

Awesome Space Shooter

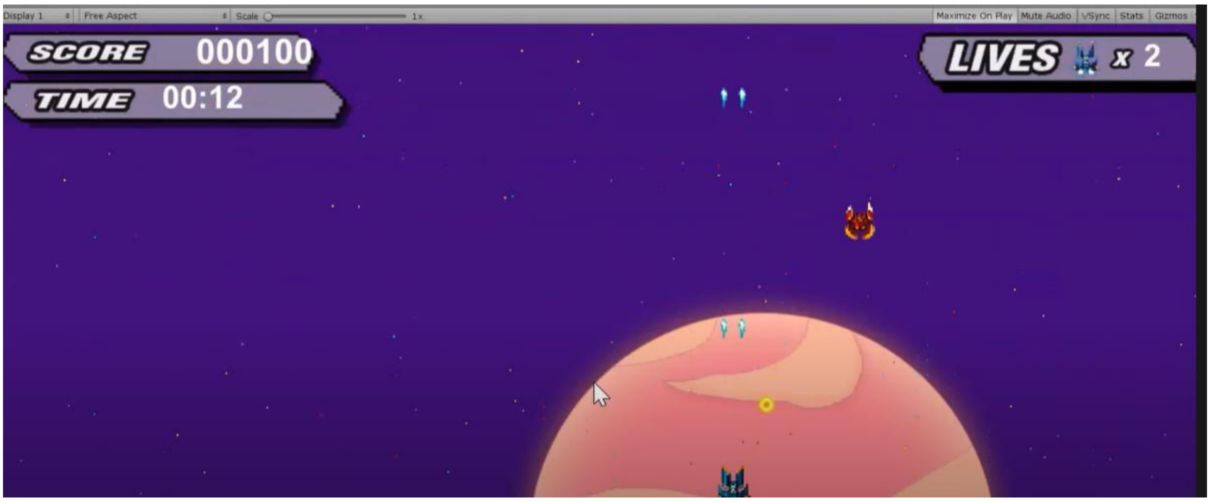


Develop an application with a Natural User Interface. There are several options available to you and this is an opportunity to combine a lot of technology that you have worked with over the past four years. You can reproduce a classic game or system using a gesture-based interface. For example, a platformer game or a navigation application using Kinect or voice control. This is a snippet of our brief which was given to us by our lecture Damian Costello. So, for my project I decided to do a classic 2d shooter game which I decided to name “Awesome Space Shooter”. This aim of the game is to kill the enemy ships which are spawning from the top of the screen and they are also firing bullets which if they hit the player ship will reduce a life. So, the player will start out with 3 lives and the score and time are both set to zero. The score will increment by 100 every time a player kills an enemy ship by either hitting with a bullet or hitting into it. The time will also increment by a second so the longer you can stay in the game the better. The game starts with the menu which Is displayed in the image above and to start the game the user must press the play button.

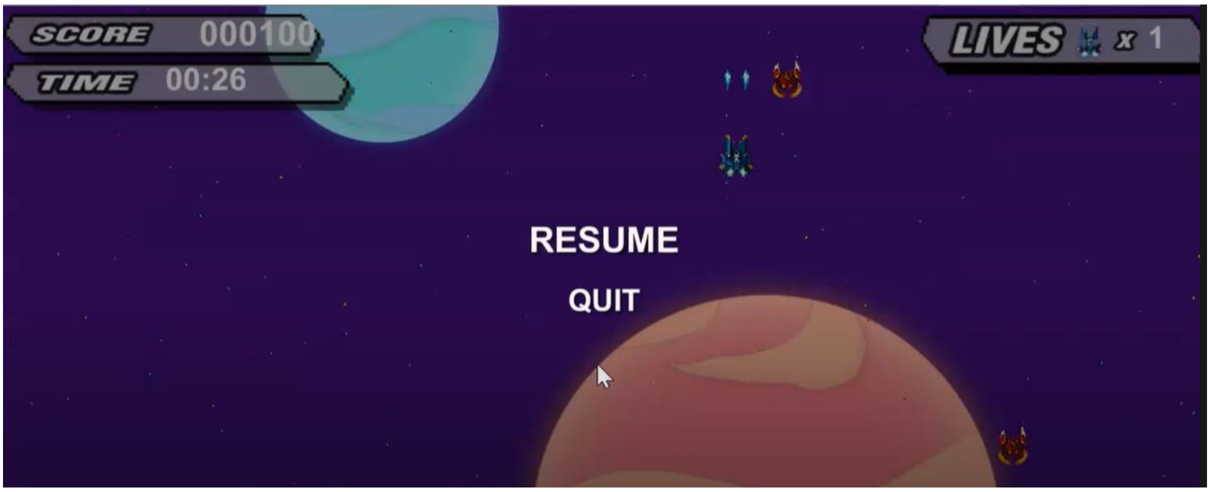
When the game launches this is where the player can then use the voice controls so for controlling the player ship the following voice controls can be used:

- Up – moves the ship in an upwards direction
- Down – moves the ship in a downwards direction
- Left – moves the ship left
- Right – Moves the ship right
- Shoot – Makes player shoot.

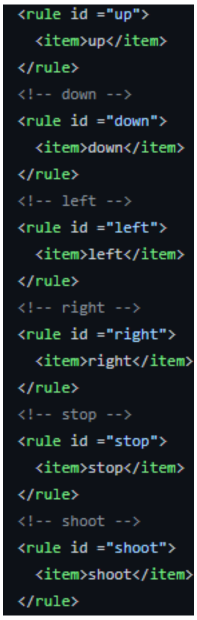
During the game to access the pause menu if the player was to say “pause” the menu will then appear and the game will stop and also the timer will stop and then if the player says “resume” the game will then resume.



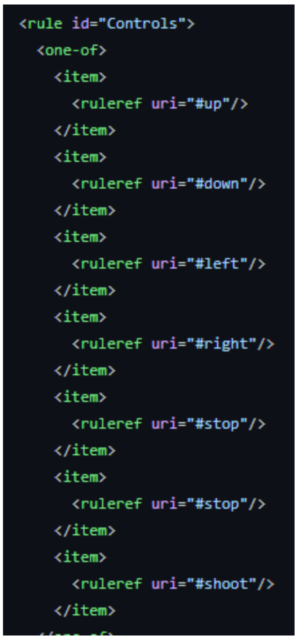
The image above shows an image of the game what the game is like when it is ran. You can see here that the time and score are working and the lives is also working and the enemy has shot a bullet towards the player and the player has also shot a bullet towards the enemy.



This is the in-game pause menu which can be activated by saying pause game and to restart the game just say resume.

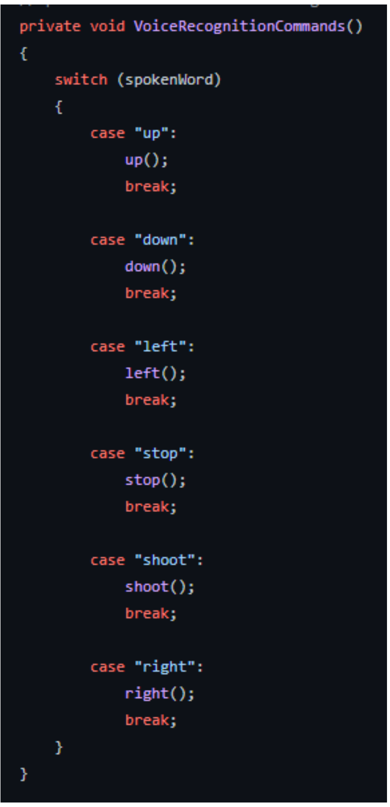


```
<rule id ="up">
  <item>up</item>
</rule>
<!-- down -->
<rule id ="down">
  <item>down</item>
</rule>
<!-- left -->
<rule id ="left">
  <item>left</item>
</rule>
<!-- right -->
<rule id ="right">
  <item>right</item>
</rule>
<!-- stop -->
<rule id ="stop">
  <item>stop</item>
</rule>
<!-- shoot -->
<rule id ="shoot">
  <item>shoot</item>
</rule>
```



```
<rule id="Controls">
  <one-of>
    <item>
      <ruleref uri="#up"/>
    </item>
    <item>
      <ruleref uri="#down"/>
    </item>
    <item>
      <ruleref uri="#left"/>
    </item>
    <item>
      <ruleref uri="#right"/>
    </item>
    <item>
      <ruleref uri="#stop"/>
    </item>
    <item>
      <ruleref uri="#stop"/>
    </item>
    <item>
      <ruleref uri="#shoot"/>
    </item>
```

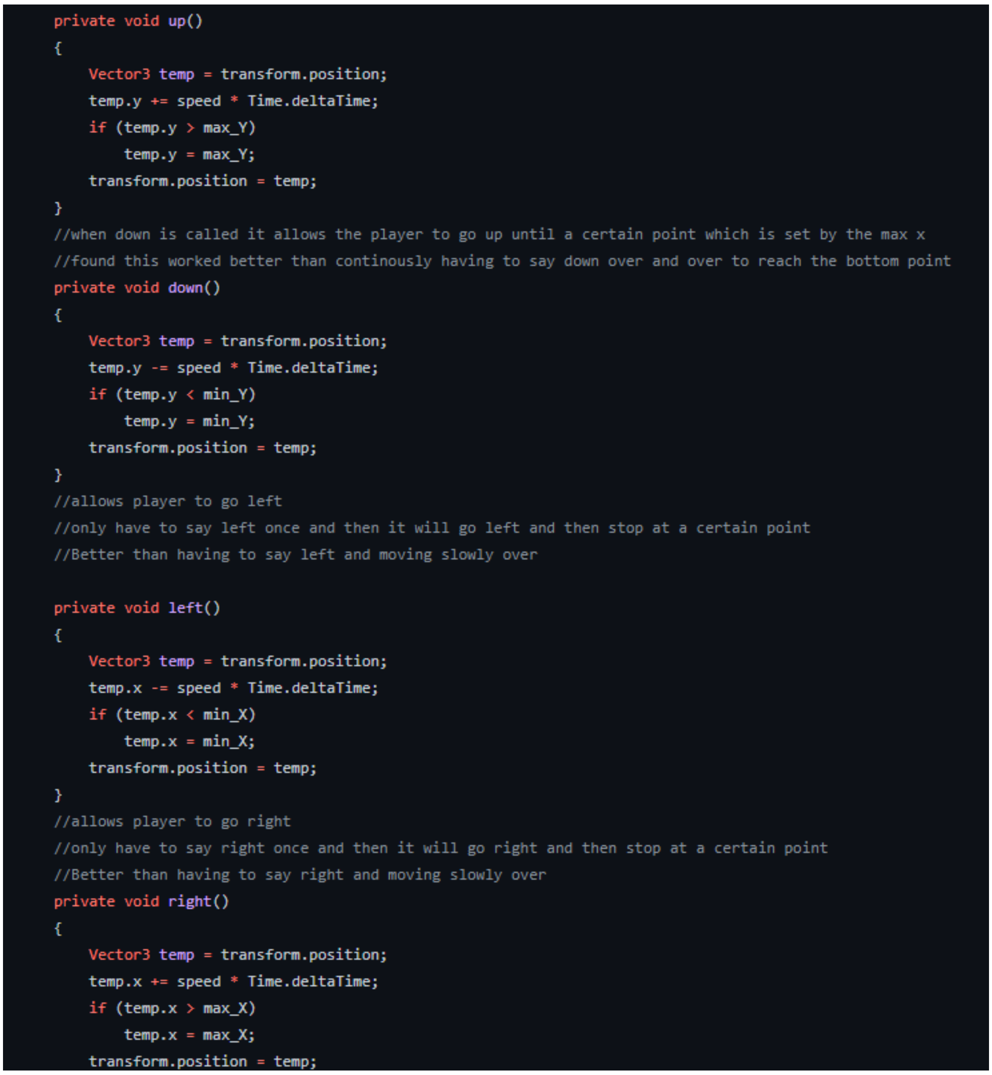
This is the grammer.xml file which is then called in the playermovement.cs file and below you can see where it is called in the playermovement.cs file.



```
private void VoiceRecognitionCommands()
{
    switch (spokenWord)
    {
        case "up":
            up();
            break;

        case "down":
            down();
            break;

        case "left":
            left();
            break;

        case "stop":
            stop();
            break;

        case "shoot":
            shoot();
            break;

        case "right":
            right();
            break;
    }
}
```

This is the switch statement where if the correct word is spoken in then this will recognise the word and then execute the correct action.

So below here I have decided to include some of the methods which are in the above switch statement just to display to you what happens in the script when something is said.



```
private void up()
{
    Vector3 temp = transform.position;
    temp.y += speed * Time.deltaTime;
    if (temp.y > max_Y)
        temp.y = max_Y;
    transform.position = temp;
}
//when down is called it allows the player to go up until a certain point which is set by the max x
//found this worked better than continously having to say down over and over to reach the bottom point
private void down()
{
    Vector3 temp = transform.position;
    temp.y -= speed * Time.deltaTime;
    if (temp.y < min_Y)
        temp.y = min_Y;
    transform.position = temp;
}
//allows player to go left
//only have to say left once and then it will go left and then stop at a certain point
//Better than having to say left and moving slowly over

private void left()
{
    Vector3 temp = transform.position;
    temp.x -= speed * Time.deltaTime;
    if (temp.x < min_X)
        temp.x = min_X;
    transform.position = temp;
}
//allows player to go right
//only have to say right once and then it will go right and then stop at a certain point
//Better than having to say right and moving slowly over
private void right()
{
    Vector3 temp = transform.position;
    temp.x += speed * Time.deltaTime;
    if (temp.x > max_X)
        temp.x = max_X;
    transform.position = temp;
```

Hardware Used in the Game

For this project I would have liked to use a piece of equipment like a connect or a raspberry Pi but due to Covid - 19 and travel restrictions it wasn't possible so unfortunately it meant that I had to develop a game using the voice controls which in hindsight I believe turned out really well.

So, the only hardware I used for this application was the microphone on my PC which I was using. The microphone Is used to get the input of words which will then run the game.

Conclusions and Recommendations

In conclusion from doing this project I believe that I learned many new skills which include including a functional pause menu which is also voice controlled, the animations I done I think really help improve the game as they look very well, I also learned a lot of about many different games from researching and trying to do voice controls on games such as snake. Looking back on this project I found it difficult and found unity to be unreliable sometimes as It might not recognise my voice sometimes which was frustrating but overall I do think it was a good experience as I believe I did learn a lot from doing this.