Gesture Based UI Project Documentation

William Vida

1 Purpose of the Application

This game is a 2D endless platformer developed using Unity 2020.1.10f1. The game uses the keyboard and mouse and uses voice commands to accompany the game's mechanics. The language settings and keyboard should be set to US English and the microphone must be enabled. The game is run using a resolution of 1920x1080. In the build settings, the main menu scene is set to 0 and the game scene is set to 1. In this game, the player controls a character and must destroy enemies and avoid obstacles while also using generated platforms to collect coins. The player's weapons are aimed at the mouse. If a valid phrase is recognised then it will be shown on screen but this is not necessarily a sign that the voice command did its intention.

Link to the GitHub repository: `https://github.com/WilliamVida/Gesture-Based-UI-Project`.

2 Gestures Identified as Appropriate for This Application

The main menu consists of three buttons, to play the game, go to the instructions menu, quit the game and go back to the main menu from the instructions menu. Voice commands can be used to navigate through the menu along with the mouse. The phrases for the main menu voice commands are stored in MainMenuGrammar.xml. These voice commands are straight forward to use.

The pause menu consists of three buttons, to resume the game, go to the main menu and quit the game. The pause menu can be accessed by clicking the escape key on the keyboard or by saying a variation of "pause" while the game is playing. The pause menu buttons can be clicked by using a mouse or they can be used by using voice commands. The phrases for the pause menu are stored in PauseMenuGrammar.xml. The voice command to access the pause menu is contained in GameGrammar.xml which is for the game voice commands. These voice commands are straight forward to use.

The phrases for the main game are stored in GameGrammar.xml. The game allows the player to use voice commands to switch between any of the two weapons available. The option to reload can also be done using voice commands. These voice commands are straight forward to use. The two actions can also be done using the mouse and keyboard. The game contains temporary power-ups and they can be acquired by moving the player in their path or by saying a variation of the power-up while it is on screen. This was done to make it easier for the player as there may be unable to acquire them as there might be objects in the way.

3 Hardware Used in Creating the Application

As I did not have any unique hardware such as a Kinect I had to use voice commands from Unity using a microphone from my laptop.

4 Architecture for the Solution

Architecture

```
private void PhraseRecogniser()
{

}
```

5 Issues

When changing weapons by voice and changing weapons by using the keyboard or mouse, sometimes one or both of them are unable to be fired. As the player goes faster, it becomes faster to acquire power-ups by voice as there is a delay by Unity.

6 Conclusions & Recommendations

Overall

7 References

Every time I used code from another website, the link to the website is placed in its respective script.

The power-up sprites were taken from https://callofduty.fandom.com/wiki/Perks.

The boosters for the player were adapted from: https://www.youtube.com/watch?v=__y10OuwVdM&ab_channel=Imphenzia.