**Functional Specifications and Design**

1. **Use Cases**

1. After starting the Voice\_Control Plugin and then the 2D Platformer game, character should jump on voice command "1"

2. After starting the Voice\_Control Plugin and then the 2D Platformer game, character fires on voice command "b"

3. Plugin should alert and not start if started without turning Windows Speech Recognizer to ON

1. **Choices**

1. Chose short commands such as "one" and "b", so that response time is fast otherwise a long command can delay it thereby affecting the game

2. Chose to make use of System.speech library in C# as every windows(8 onwards) has a speech recognizer which serves as a voice recognition engine

3. Selected a game (2D Platformer) based on Unity as it is one of the most popular game engines

1. **Design**

****

1. **Voice\_Control** is an application written in C# making use of Windows Speech Recognition. It keeps on listening to user and upon the correct match creates a “.tmp” file (jump\_voiceControl.tmp, fire\_voiceControl.tmp) in “My Documents” for every command as soon as it is spoken.

Pre-requisite: Cannot run unless Windows Speech Recognition is ON.

1. **2D Platformer:** Modified the function update ( ) in controllers : PlayerControl.cs and Gun.cs. In addition to key press for respective action, function now checks in every frame if the .tmp files (jump\_voiceControl.tmp, fire\_voiceControl.tmp) for respective command is present in “My Documents”. If present action is applied to frame.

These files are deleted immediately once the action is done

1. **Assumptions**

1. Speech Recognition is enabled and microphone set up already

2. User uses a headset as sound from game can interrupt with voice command

3. Plugin is launched before game

4. Time to speak and interpret a command is acceptable for the game

5. Application has rights to create .tmp files in My "Documents"

1. **Future Improvements**
2. Could make use of sockets which could transmit real time data over UDP or TCP to which the game controllers could listen, but .tmp file creation serves the purpose for now
3. Starting the game should run the Voice\_Control plugin in the background instead of user starting it.(Could not be done due to time limitations)
4. **Unit Test Cases(Tested on Windows 10 x64)**
5. **Test case**: Should jump on saying "1"

Steps:

1. Start Windows Speech Recognizer

2. Start Voice\_Control Plugin and then the 2D Platformer game

3. Press the mic to listening mode and start the game

4. Say "1"

**Expected Result**: Character jumps

**Observed Result**: Character jumps

1. **Test case** : Should Fire on saying "b"

**Steps**:

1. Start Windows Speech Recognizer

2. Start Voice\_Control Plugin and then the 2D Platformer game

3. Press the mic to listening mode and start the game

4. Say "b"

**Expected Result**: Character fires gun

**Observed Result:** Character fires gun

1. **Test case**: Alert must be shown if Plugin started without starting Windows Speech Recognizer

**Steps**: Start Voice\_Control Plugin without starting Windows Speech Recognizer

**Expected Result**: Alert must be shown

**Observed Result**: Alert is shown to start Windows Speech Recognizer