

BSc in Software Development – Year 4

Submission Document

Voice Recognition Project

Jina Kim

G00353420

Developed using Unity version: 2019.4.11f1

This is a voice controlled 2D game.

The voice control completely replaces the mouse and keyboard as a method of control in this unity game project.

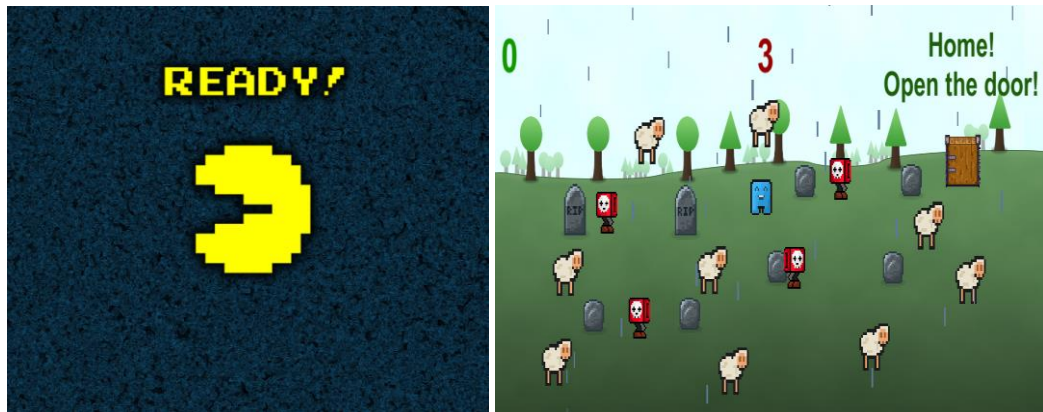
It contains 12 different grammar rules.

1. Start a new game
2. Finish the game
3. Player's movement state
4. Movement directions
5. Fire the bullet to kill the enemies
6. Change weapon
7. Start raining (Make the rainy effect on game scene)
8. Stop raining
9. Pause the game
10. Resume the game
11. Display the game story
12. Stop displaying the game story

Features

- **Navigation**

- The splash scene should be displayed in the beginning. When a user says, "Start a new game", it will navigate to the main game scene.



- When a user says, "Finish the game" or "Quit the game", it will navigate to the first scene.
- When the player finishes the game or dies, it will navigate to the appropriate scene at which point the player can command "Start a new game" or "I want to play".



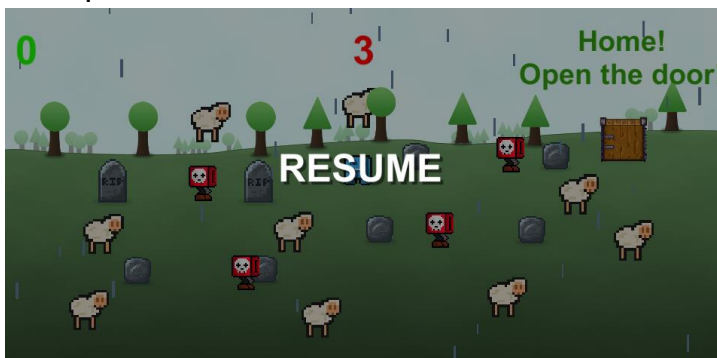
- **Shooting** - The player must be provided with a weapon in order to kill an enemy.

The weapon will be spawned when the player says "Shoot" or "Fire the bullet".

If the player wants to change weapon, the user can say "change weapon"



- **Score** — points are given for killing enemies and collecting sheep. (1 point each)
- **Player movement** — horizontal and vertical movement.
The game character will be controlled by what the user says. E.g. “Go up”, “Move right”, “Move left”, “Go down”.
- **Game Completed** — The player should be able to complete the game and receive recognition from the game that it is over.
- **Quit Game** — The player should be able to quit the game and stop playing. When the user says, “Finish the game” or “Exit the game”, it will navigate to the first scene.
- **Pause Game** — The user should be able to pause the game. When the user says, “Pause the game”, it will pause the game and give the player the option to continue.



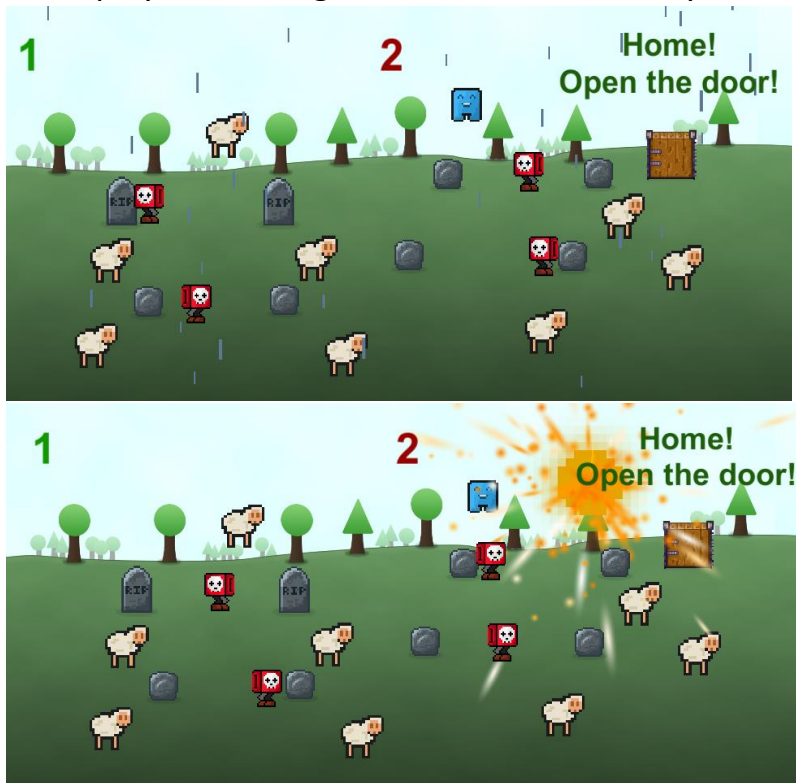
When the user says, “Resume the game”, it will resume the game.

- **Health System** – The player must be able to lose health when attacked by enemies. When the player has run out of health (The user has 3 lives in this game), they will have to respawn which will bring them back to the start of the game.
- **Weather condition**

The player can control the weather condition.

When a user says, “Start raining” or “It’s raining”, the rainy effect will be displayed on the game scene.

When a user says, “Stop raining” or “it’s not raining”, the sun effect will be displayed on the game scene and the rainy effect will be disappeared.



- **Display the game story**

The player can check the game story during the game.

When a user says, “Display the game story” or “show game story”, the game story will be displayed.

When a user says, “It’s enough” or “don’t display game story”, the game story will not be displayed.



XML Grammar

```
<rule id="toplevel" scope="public">
  <one-of>
    <item>
      <ruleref uri="#playstate" />
    </item>
    <item>
      <ruleref uri="#movestate" />
    </item>
    <item>
      <ruleref uri="#killEnemies" />
    </item>
    <item>
      <ruleref uri="#changeWeapon" />
    </item>
    <item>
      <ruleref uri="#makeRain" />
    </item>
    <item>
      <ruleref uri="#gamePauseResume" />
    </item>
    <item>
      <ruleref uri="#showGameStory" />
    </item>
  </one-of>
</rule>
```

The player can start or finish the game

```
<!-- Start and Finish the game rules (Rule 1,2) -->
<rule id="playstate">
  <one-of>
    <item>
      <tag>out.action = "new";</tag> <!-- Start the game -->
      <one-of>
        <item> Start a new game </item>
        <item> Begin a new game </item>
        <item> New game </item>
        <item> I want to play </item>
        <item> Start </item>
        <item> Play </item>
        <item> Begin </item>
      </one-of>
    </item>
    <item>
      <tag>out.action = "quit";</tag> <!-- Finish the game -->
      <one-of>
        <item> finish the game </item>
        <item> exit game </item>
        <item> I give up </item>
        <item> Finish </item>
        <item> Exit </item>
      </one-of>
    </item>
  </one-of>
</rule>
```

The player can control the movement, E.g. move right, move left, move up, move down.

```
<!-- Controls the player movement rules (Rule 3) -->
<rule id="movestate">
  <one-of>
    <item> Go </item>
    <item> Move </item>
  </one-of>
  <one-of>
    <item>
      <ruleref uri="#direction" />
    </item>
  </one-of>
</rule>
```


Player's movement directions.

```
<!-- Player's direction rules (Up, Down, Left, Right) (Rule 4) -->
<rule id="direction">
  <one-of>
    <item>
      <tag>out.action = "up";</tag>
      <one-of>
        <item> up </item>
        <item> top </item>
      </one-of>
    </item>
    <item>
      <tag>out.action = "down";</tag>
      <one-of>
        <item> down </item>
        <item> bottom </item>
      </one-of>
    </item>
    <item>
      <tag>out.action = "right";</tag>
      <one-of>
        <item> right </item>
        <item> right side </item>
      </one-of>
    </item>
    <item>
      <tag>out.action = "left";</tag>
      <one-of>
        <item> left </item>
        <item> left side </item>
      </one-of>
    </item>
  </one-of>
</rule>
```

The player can attack the enemies with weapons.

```
<!-- Attack enemy rule (Rule 5)-->
<rule id="killEnemies">
  <one-of>
    <item>
      <tag>out.action = "shot";</tag> <!-- Shot the bullet -->
      <one-of>
        <item> Fire the bullet </item>
        <item> Shoot the enemy </item>
        <item> Fire </item>
        <item> Shoot </item>
        <item> Attack </item>
      </one-of>
    </item>
  </one-of>
</rule>
```

The player can change weapons.

```
<!-- Change weapon rule (Rule 6)-->
<rule id="changeWeapon">
  <one-of>
    <item>
      <tag>out.action = "change";</tag> <!-- Change player's weapon -->
      <one-of>
        <item> change weaspon </item>
        <item> change bullet </item>
      </one-of>
    </item>
  </one-of>
</rule>
```

The player can change weather conditions.

- Rainy weather.
- Sunny weather.

```
<!-- Change weather rule (Rainy weather, Sunny weather) (Rule 7,8)-->
<rule id="makeRain">
  <one-of>
    <item>
      <tag>out.action = "rain";</tag> <!-- Display rainy effect -->
      <one-of>
        <item> it is raining </item>
        <item> raining </item>
        <item> start raining </item>
      </one-of>
    </item>
    <item>
      <tag>out.action = "norain";</tag> <!-- Stop displaying rainy effect -->
      <one-of>
        <item> it is not raining </item>
        <item> not raining </item>
        <item> Stop raining </item>
      </one-of>
    </item>
  </one-of>
</rule>
```

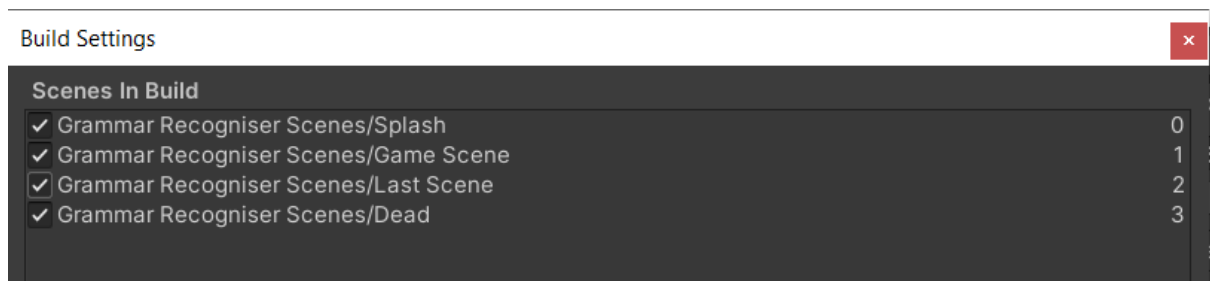
The player can pause or resume the game.

```
<!-- Game Pause and Resume rules (Rule 9,10)-->
<rule id="gamePauseResume">
  <one-of>
    <item>
      <tag>out.action = "pause";</tag> <!-- Pause the game -->
      <one-of>
        <item> pause the game </item>
      </one-of>
    </item>
    <item>
      <tag>out.action = "resume";</tag> <!-- Resume the game -->
      <one-of>
        <item> resume the game </item>
      </one-of>
    </item>
  </one-of>
</rule>
```

The player can display the game story.

```
<!-- Display Game Story rule (Rule 11,12)-->
<rule id="showGameStory">
  <one-of>
    <item>
      <tag>out.action = "display";</tag> <!-- Display the game story -->
      <one-of>
        <item> display game story </item>
        <item> show game story </item>
      </one-of>
    </item>
    <item>
      <tag>out.action = "donotDisplay";</tag> <!-- Stop displaying the game story-->
      <one-of>
        <item> it is enough </item>
        <item> I got it </item>
        <item> don't display game story </item>
      </one-of>
    </item>
  </one-of>
</rule>
```

Image of Build Settings



1. Splash Scene – The first scene.
2. Game Scene – Main game scene.
3. Last Scene – When the player finishes the game this scene will be displayed.
4. Dead Scene – When the player loses all lives, this scene will be displayed.

Hierarchy

