**Project Synopsis**

**3D Snake Game**

1. **Abstract and Keywords :**

**Abstract**

In this game the player controls a long, thin creature, resembling a snake, which roams around on a bordered plane, picking up food & trying to avoid hitting its own tail or the edges of the playing area. Each time the snake eats a piece of food, its tail grows longer, making the game increasingly difficult. The user controls the direction of the snake's head ( left, or right), and the snake's body follows. When the next game begins, the previous highest score will be displayed on the screen.

**Keywords**

This game is developed for Android based smartphone. This game was made using the Unity game engine with C# as the programming language.

* Running Snake.
* Glowing Fruit.
* Levels.
* Obstacles.
* High Score.

1. **Introduction :**

* Snake is a video game that originated during the late 1970s in arcades becoming something of a classic. It became the standard pre-loaded game on Nokia phones in 1998.
* In this snake game, the player controlled snake must collect the foods and stay away from its own growing body.
* The player also needs to stay away from the objects in the game.
* When the player hits the score to 100 the level will rise and so on.
* If the snake dies, the game is over.
* During the process of the game, the system will record a score based on the fruits collected.
  1. **. Problem Statement.**

The problem statement for Snake 3D can be described as follows:

Design and implement a 3D version of the classic arcade game Snake, in which the player controls a snake-like creature moving through a three-dimensional space. The snake must grow by eating food while avoiding obstacles and its own body. The game should feature a simple and intuitive control scheme, with responsive movement and collision detection. The game should also include sound effects and graphics that enhance the overall gameplay experience. The game should be challenging and engaging, with multiple levels of difficulty and modes of play to keep players engaged over time.

* 1. **Literature Review/Description of Present System**

"Design and Implementation of Snake 3D Game Using OpenGL": This paper describes the design and implementation of Snake 3D using OpenGL. The authors explain how they implemented the game's graphics, physics, and user interface, including the use of textures and lighting effects to create a realistic 3D environment.

"A comparative study of 2D and 3D Snake game design": This study compared the gameplay experience of 2D and 3D versions of Snake. The authors found that the 3D version provided a more immersive and engaging experience, but also noted that the additional complexity of the game's controls and mechanics may be a barrier to entry for some players.

"A Novel Approach to Snake Game: 3D Maze": This paper describes a 3D version of Snake in which the player navigates a maze-like environment, collecting food and avoiding obstacles. The authors highlight the game's use of 3D space to create a more challenging and dynamic gameplay experience.

"Development of a 3D Snake Game Using Unity": This paper describes the development of a 3D version of Snake using the Unity game engine. The authors discuss the challenges of implementing the game's physics and collision detection in a 3D environment, and how they overcame these challenges using Unity's built-in features and scripting tools.

"A Comparative Study of Classic Snake Game and 3D Snake Game": This study compared the gameplay experience of classic 2D Snake with a 3D version of the game. The authors found that the 3D version provided a more engaging and immersive experience, but noted that it also required more processing power and may be more difficult for some players to control.

Overall, the literature on Snake 3D suggests that it can provide a more engaging and immersive gameplay experience compared to traditional 2D versions of the game. However, there are also challenges associated with implementing Snake in a 3D environment, particularly in terms of controlling the snake's movement and avoiding collisions with obstacles.

* 1. **. Background/Limitations**

Snake games is often rely heavily on luck and chance. While skill can play a role in navigating obstacles . This can make the game feel unfair or frustrating at times, especially if the player is trying to achieve a high score.

Finally, while Snake 3D games can be fun and addictive, they do not offer the same level of complexity or depth as other types of games. They are often designed to be played in short bursts, rather than as a long-term, immersive experience.

The limitations of present implementation of Snake game are :

* 1 Whenever the Game Restarts the Player always starts from its intial position, it does not loads or save the game from the place where it dies.
* It is an Offline Game, and a Single Player Game.
* This game was built for Android Platform only.
  1. **. Aim & Objectives**

**Aim**

The Aim of the Game is as clear as Crystal, as the roll of every Game is to give you a realistic feeling and experience to your mind as if you are playing game in your realistic life. It removes the stress and relaxes your mind while playing and enjoying the game. Similarly the aim of the snake game is to get enjoyable and stressfree experience. As the Graphics of the game is in such 3D way that it makes us feel realistic.

**Objectives**

* It passes our Free boring time.
* It makes our Mind relax and stressfree.
* The Game makes our brain tactics sharp.
  1. **. Project Motivation**

As in today’s world the level of game is increasing day by day to a better level. Previously there were 2D games, to which inherited

1. **Description of Proposed Work**
2. **Technology/Language/Development Tools/ Hardware**
3. **Conclusion & Future Scope**

**Extras:**

Conclusion

We build a 3D VR snake game using Unity in this project. Further improvements including history score and leisure mode will be considered in the future. Hope you enjoy this game!

4.4. Four Types of game blocks There are four types of blocks in the game, including fruit, bomb, accelerator and decelerator. When the snake touches fruit, we add one more node on snake, which has the index that equals the length of the snake (assuming zero indexing). There is a variable nodes[] in our code that stores the information of nodes of the snake. And a score text will show up above the head of the sanke. Every time it touches fruit, the player gains one point.

When the snake touches bomb, the game ends. The text ”End Game” will be shown on the screen immediately. The score will keep stay on the screen. Also, when the snake touches itself or walls, the game will end as well. When the snake touches accelerator or decelerator, the speed of the snake will be updated, which is achieved by changing the value of default time interval for snake movement.

References

[1] 3d snake assets. https://github.com/ AwesomeTuts/3d-snake-tutorial, 2018.

[2] M.Gotsis. Games,virtual reality,and the pursuit of happiness. https://www.researchgate.net/publication/ 40020916\_Games\_Virtual\_Reality\_and\_the\_ Pursuit\_of\_Happiness, 2009.

[3] J. Vegas. Unity tutorial for beginners. https: //www.youtube.com/watch?v=G9BdFZ2MCXc& list=PLZ1b66Z1KFKik2g8D4wrmYj4yein4rCk8, 2015.

4.2. Orientation Tracking (VRduino)

The game has two inputs: either ”left” or ”right”. One feasible way to input direction is by reading keyboard input, i.e. when pressing ”leftarrow” key, the snake turns left, when pressing ”rightarrow” kew, the snake turns right. To make sure that each pressing corresponds to exact one turn, we use function Input.GetKeyDown. Besides keyboards input, we track the orientation of head which is then converted into ”left” or ”right”. Using the scripts provided on Piazza, quaternion of head could be obtained by reading data from VRduino. Afterwards, yaw could be computed from quaternion. For each update, we have two variables, previous yaw yprev and current yaw ycur. We determine whether the difference between yprev and ycur is from 60◦ to 120◦ . If not, nothing happen; if yes, we update yprev as ycur and make left turn. The same method is applied to right turn.

4. Game Design

4.1. Snake tracking

In order to track the movement of the snake, two functions are implemented. The first one is to follow the position of the snake. When the snake moves, the camera is supposed to move along with the head. Aiming to get the best perspective, we set the camera 0.6 unit above the head and 2.5 unit behind the head, which could be fine-tuned manually. The second one is to track the rotation of the snake. We directly set the orientation of the camera related to the input, so that when the snake turns left or right, our view would follow its rotation.

3. Game Description

Our VR snake game is similar to typical snake game with regards to the basic logistics. It inherits the model of a 2D snake game and creates a 3D view which could be controlled by human movement. The model mainly includes a snake, four types of blocks and a fixed-size region that restricts the activity range of the snake. The goal of the game is to make the snake grow as long as possible. The score is calculated and displayed on the screen according to the length of the snake. One of the main differences of our game from the typical one is that the perspective could be changed from top view to snake view, as shown in Figure 1. Another difference is that: besides the inputs of keyboard , head movement is also designed as an input, which will be discussed in more details in section 4. Moreover, there are four types of blocks in this game (Figure 2), including fruit, bomb, accelerator and decelerator. As could be implied from semantics: fruits lengthen the snake; bombs end the game; accelerators accelerate the snake; decelerators decelerate the snake.

Introduction

For this project, we are going to construct a 3D virtual reality (VR) snake game using Unity and VRduino. Snake is a video game concept where a player controls a line-style snake which grows in length, with the wall and snake itself being the primary obstacle. To improve user experience, several concepts are included in our game, including snake perspective, four innovative blocks, snake velocity and so on. Inputs, such as keyboard inputs and head movement, are also studied in this project. In the following parts, some related works, detailed game description for players and technical design of our project will be discussed.

Related Work

In terms of the history of snake game design, it dates back to the arcade game Blockade that was developed and published by Gremlin in 1976. In 2017, Google released their version of the snake game as an Easter egg which implies its popularity. Furthermore, with the rapid speed of VR gaming industry, developing a user-friendly environment is of vital importance and necessity [2], which is the goal of this project. As for technical aspect, we mainly learn from two resources: Unity tutorial provided on Youtube channel [3] and 3D snake assets provided by [1].