

# 3D Snake Game

WeiQuan Mao  
Stanford University  
mwq@stanford.edu

Jiyao Yuan  
Stanford University  
yuan999@stanford.edu

## 1. 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.

## 2. 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].

## 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.

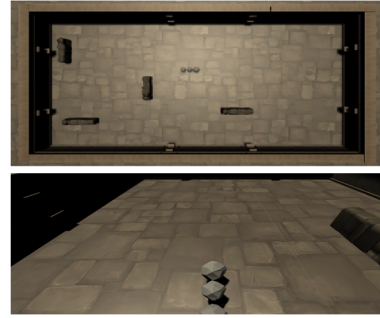


Figure 1. Two different perspectives of the game (*upper*: top view, *bottom*: snake view)

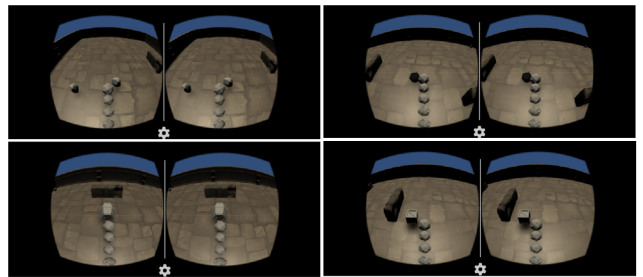


Figure 2. Four different blocks of the game. *Top-left*: fruit. *Top-right*: bomb. *Bottom-left*: decelerator. *Bottom-right*: accelerator.

## 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.

## 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" key, 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  $y_{prev}$  and current yaw  $y_{cur}$ . We determine whether the difference between  $y_{prev}$  and  $y_{cur}$  is from  $60^\circ$  to  $120^\circ$ . If not, nothing happen; if yes, we update  $y_{prev}$  as  $y_{cur}$  and make left turn. The same method is applied to right turn.

## 4.3. Snake Movement

General speaking, snake movement is by updating the position for each nodes. The snake has two components, which are head and main body. Head is the first node of the snake. For head, it will read the current direction of the snake, and move towards that direction in a default distance (which equals to the diameter of the node). For each node in main body, it will move to the position that its last node stays.

Another challenge is how to determine the time interval between two adjacent movement. In majority case, it is a constant value except for the case that "left" or "right" is inputted. When the snake turn left or right, it will move immediately instead of waiting for a constant time.

## 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 snake. 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.

## 5. 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!

## 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](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.