Eralp Tunalı

Eren Barış Bostancı

İrem Arpağ

COMP410 TERM PROJECT REPORT

3D GAME IMPLEMENTATION

**3D GAME IMPLEMENTATION OPHIDO**

1. **INTRODUCTION**

Ophido game is a single player survival game where the main aim is to survive as much as possible. The main character Ophido, which looks like a cube or list of cubes depending on the size, is controlled by the player. Whenever Ophido eats a food, the size of Ophido increases. Ophido is afraid from the enemy, which looks like a rabbit. If Ophido interacts with the enemy, the size of Ophido decreases. There are power ups in the game, if Ophido interacts with power ups, which looks like a cube, Ophido gains some powers such as duplicating the size, ability to pass through the walls and immortal mode. The final component of the game is the wall, if Ophido hits the wall it dies immediately and the game ends. The main goal of our project is to implement a challenging 3D game. To make our game challenging we make players to experience different events and handle them differently. Another aim of our project is to implement a fun to play and demanded 3D game. The game must be complex but not beyond the limits, because if the game becomes extremely complex the demand to play the game will be decreased so we implemented our Ophido game complex while not compromising from fun to play the game. Lastly, the game graphics must be smooth enough to attract new users to play the game. Also, the better graphics in our game will reduce the eye exhaust of the players.

1. **BACKGROUND/REVIEW**

To achieve our aims and goals, we have to acknowledge some background information. Our background information to implement the Ophido game are C, C++, OPENGL, Viewing, Projections, Geometry, Transformations, Shading and Texture Mapping. Our projection model to view the game is perspective projection. Our main character Ophido supports two different viewing type, shading and texture mapping. The option for texture mapping is, mapping a basketball ball to Ophido. The food, which is a sphere, is also supporting two types of viewings, shading and texture mapping. The option for texture mapping is, mapping a cheese to the food. The enemy, which is a rabbit, just supports shading. The wall which is also a main element of the game supports shading. The power ups, which are cubes, just supports shading. All these configurations can be set from the options menu in our Ophido game.

1. **IMPLEMENTATION DETAILS**
2. **OPHIDO**

Our Ophido class has its own constructor and draw methods. The additional method in Ophido is damage, grow and move. When Ophido interacts with a food, its size is increased by one especially since Ophido holds many cubes, we add one cube to the end, the grow method does that. When Ophido interacts with the enemy, its size is decreased by one, the damage method does that. When the key W(front), A(left), S(back) or D(right) is pressed the first cube of Ophido moves to that direction and the rest of the cubes are setting target to the cube which is in front of them.

1. **FOOD**

Our Food class has its own constructor and draw methods. There is no additional method in this class, since this class is a sphere which does not moves until being eaten, we handled the respawn of the food in the main. When the Ophido interacts with our food, the visibility of food is set to 0 and respawned in a random location.

1. **ENEMY**

Our Enemy class has its own constructor and draw methods. The additional method in this class is tick, since the enemy is moving towards one axis X or Y, while not being interacted with Ophido the movement is handled in the tick method. We handled the Ophido interaction in the main, when the Ophido interacts with our enemy, the visibility of the enemy is set to 0 and the enemy is respawned in a random location with random direction X or Y.

1. **POWER UPS**

Our Power Up class has its own constructor and draw methods. The difference of this class is the draw method since there are many types of power ups, while drawing the power up the color of the power up is set according to its type. Since this power up is a cube which does not moves until being eaten, we handled the respawn of the power ups in the main. Also, when Ophido interacts with a power up, Ophido is buffed according to the type of the power up.

1. **CHALLENGING PARTS**

While implementing our 3D Ophido Game, we were challenged in some parts of our project. Those parts were handling the interactions between Ophido-Food, Ophido-Enemy and Ophido-PowerUp. Since our Ophido is containing cubes, to make a realistic interaction with the Enemy(Rabbit) or Food(Sphere) we considered an invisible cube around those objects and when those two cubes interact, we assumed that Ophido hit to the enemy or food. The following figures shows the methods in detail.

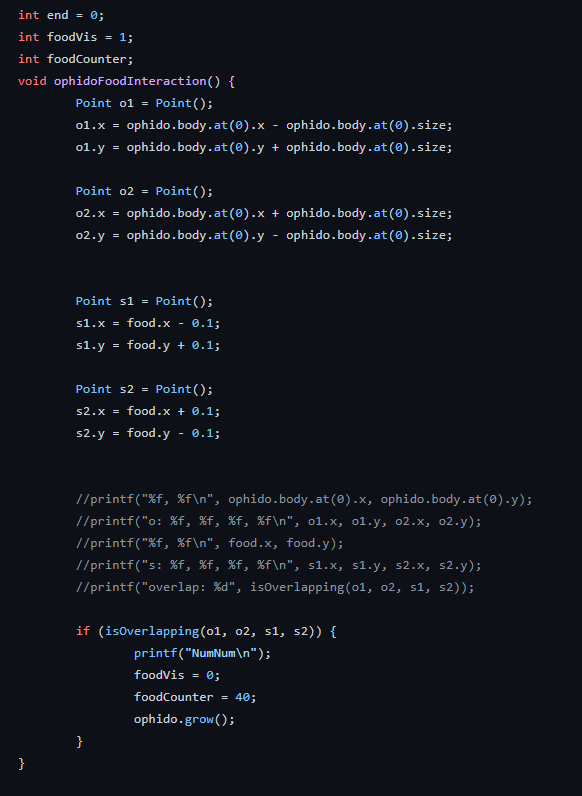


Fig1: Ophido Food interaction.

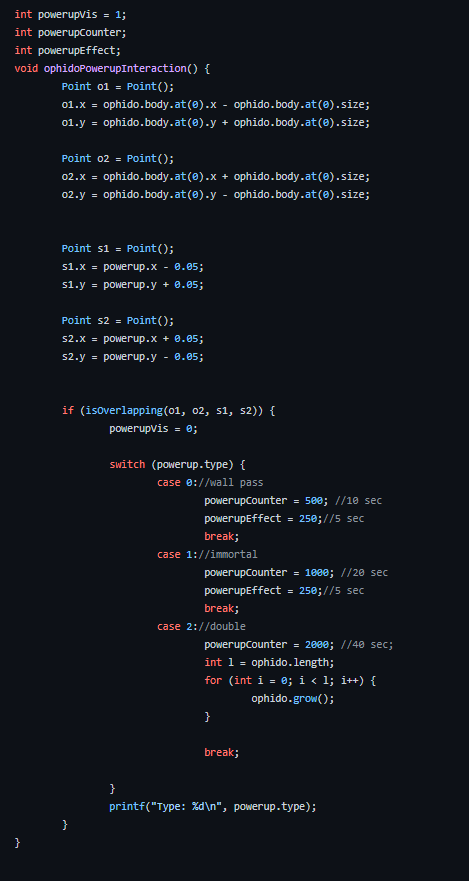


Fig2: Ophido Power Up interaction.

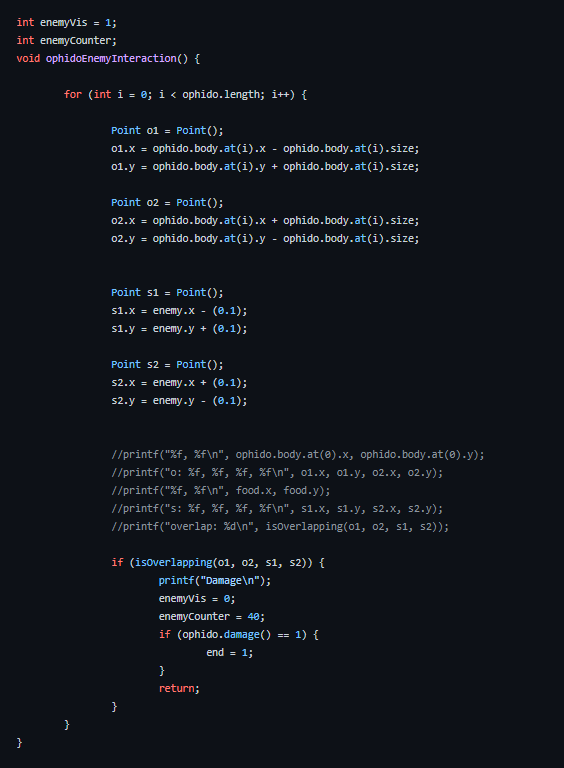


Fig3: Ophido Enemy interaction.

1. **EVALUATION**

To evaluate our project totally, we handled the basics of our Ophido game. We learned how to implement texture mapping and shading to different objects. We learned how to handle the collusions between the game objects. We did this with cube collusion methods, however since all of the objects are made of triangles, we could iterate through all the triangles of the object and determine whether there is a collusion. However, this method could slower our animation. The overall game was fun to play and complex enough to let users demand to play the game.

1. **CONCLUSION & FURTHER IMPLEMENTATIONS**

To conclude, we created a 3D survival game with different game objects. Our game provides smooth graphics and several challenges, the game is complex but not beyond the limits to make the players have more fun. The further implementations to our game can be that we can include a random movement to the enemy which both increases the difficulty and the fun of the game. Also, we can include some different maps to the game and players can experience new maps when they are bored on the same map. Furthermore, we can improve our view angles to the game by adding an option to the players to look from the first person sight to the game. To sum up, all the game basics are working, we reinforced our shading and texture mapping knowledge and created a fun 3D game with smooth graphics.