Final Project

CS 550 - INTRO TO COMPUTER GRAPHICS

Project Number: Final
Project Name: Final Project

Video Link: https://media.oregonstate.edu/media/t/1_gdr0wrgl

Name: Tianle Yuan (933-946-576)

Email: yuant@oregonstate.edu

Time: December 7, 2020

1.The text from your proposal.

Introduction:

Horrible games are always one kind of the hottest game on the different game platforms. The Switch game, Luigi's Mansion 3, released by Nintendo company in 2019, won the favor of a vast number of players. Inspired by this game, the project will simplify the "room exploration" scenario, which has been shown in **Figure 1**. The program will construct an arm, which can be moved with the cursor, in a dark room. As the same as Luigi's Mansion 3, the tail of the arm will hold a spotlight, which can light different places of the room according to the position of the arm. The idea has been shown in **Figure 2**.

Figure 1. A scene of exploration of a room

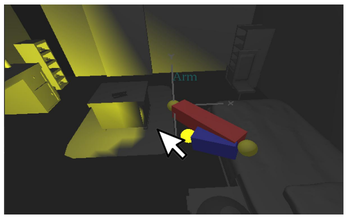


Figure 2. A rough demo

Core Techniques:

Besides the basic realization of spotlight and object texture, the hardest part of the project is the “adaptive” moving arm, which means the mobile angle of the arm structure is based on a solution of the real-time numerical equation rather than several single **gl_rotate3f()**. The mathematical formula can be expressed below**:**

$$\begin{cases} \min\limits_{\theta_1,\theta_2} [L_1 \cos\theta_1 + L_2 \cos(\theta_1+\theta_2) - x_0]^2 + [L_1 \sin\theta_1 + L_2 \sin(\theta_1+\theta_2) - y_0]^2 \\ \| \text{point}_{\text{tail}} - \text{point}_{\text{cursor}} \| < (L_1+L_2),\ \text{or}\ \| \text{point}_{\text{tail}} - \text{point}_{\text{cursor}} \| > (L_1-L_2) \end{cases} \quad \textbf{(1)}$$

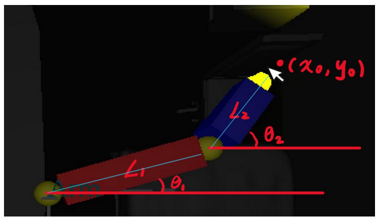


Figure 3. Parameters for numerical modeling

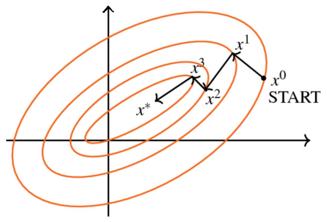


Figure 4. The Steepest Descent method

Work direction:

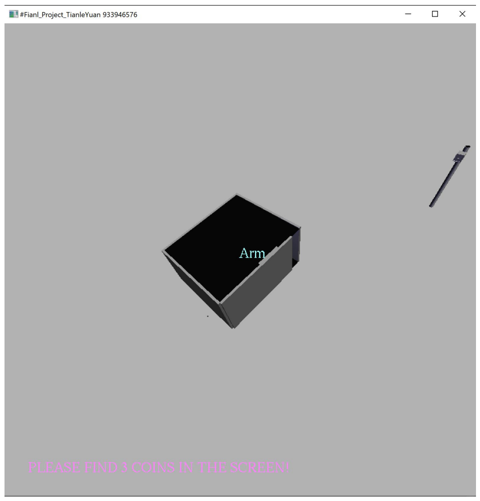
Situations of $\| \text{point}_{\text{tail}} - \text{point}_{\text{cursor}} \| > (L_1 + L_2)$, and $\| \text{point}_{\text{tail}} - \text{point}_{\text{cursor}} \| < (L_1 - L_2)$ can be realized by treating the arm as a sticker with no joints. They can be realized by using **gl_rotate3f()** according to the position of the cursor.

However, for the situation of formula (1), the solution to this problem can be classified as the Optimal Solution for the Unconstrained Problem. At present, it is known that the methods to solve this problem are: the Steepest Descent method, Newton method, and Quasi-Newton method. In order to have a smooth moving of the arm, the number of numerical iterations, the coefficients setting, and the step size will be the main consideration part. Also, not all the positions of the cursor will converge to the domain of the optimal solution, thus, the project will also need to set a threshold for iteration. Setting $\| \boldsymbol{\alpha}_k * \mathbf{d}_k \| < \varepsilon$ would also be useful.

2.What you actually did for your project, with images

My final project focused on the topic of "Horrible game of exploring the room" and created a simple interface for players to find 3 random gold coins in the room. There are four major parts I did in my project.

The first part would be the environmental scene reading. I used **loadobjfile.h** for loading **Room.obj** file to construct a scene in a room (the obj file does not have a roof and ground in the house). The material of the room has been set as dull material with deep gray color. **Figure 1** and **Figure 2** show the room's structure and settings when all the lights are turned off.


#Fianl_Project_TianleYuan 933946576
Arm
PLEASE FIND 3 COINS IN THE SCREEN!

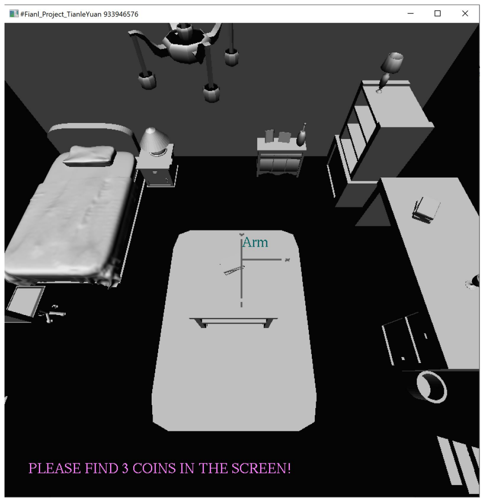
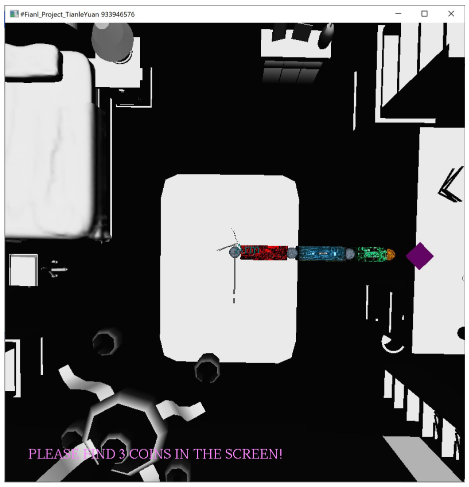

#Fianl_Project_TianleYuan 933946576
Arm
PLEASE FIND 3 COINS IN THE SCREEN!

Figure 1. The outside of the house

Figure 2. The inside of the house (room)

The second part is a "robotic arm" holding a flashlight. It is the core part of the project. I used Inverse Kinematics technique for calculating the animation of the arm (3 joints) following the cursor. CCD (Cyclic Coordinate Descent) is the algorithm used to realize the motion functions. In **Figure 3**, the purple square is used to indicate the display screen cursor's mapped place on Z-o-X plane. The primary degree of freedom of the arm is on Z-o-X plane, which means the arm's movement will follow the purple square on Z-o-X plane. **Figure 4** is the movement.


#Fianl_Project_TianleYuan 933946576
PLEASE FIND 3 COINS IN THE SCREEN!

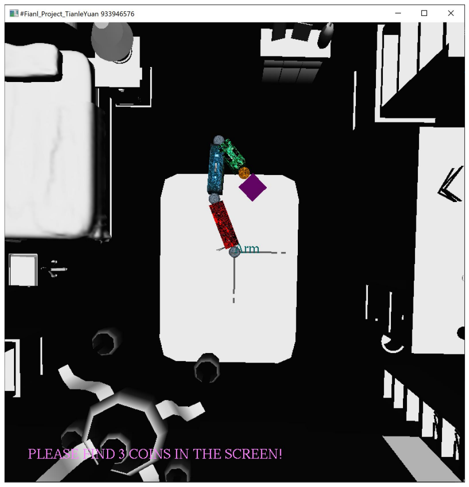
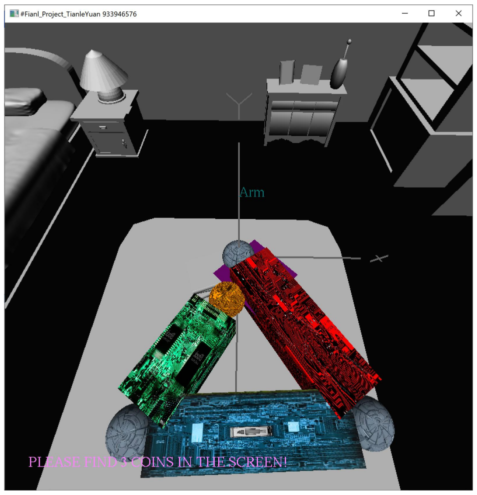

#Fianl_Project_TianleYuan 933946576
Arm
PLEASE FIND 3 COINS IN THE SCREEN!

Figure 3. The robotic arm on Z-o-X plane

Figure 4. The movement following square

From **Figure 5.**, it is not hard to find that all the joints used the gray metal texture, the first arm used the red electronic texture, the second arm used the blue electronic texture, the third arm used the green electronic texture, and the bulb used the yellow electronic texture.


#Fianl_Project_TianleYuan 933946576
Arm
PLEASE FIND 3 COINS IN THE SCREEN!

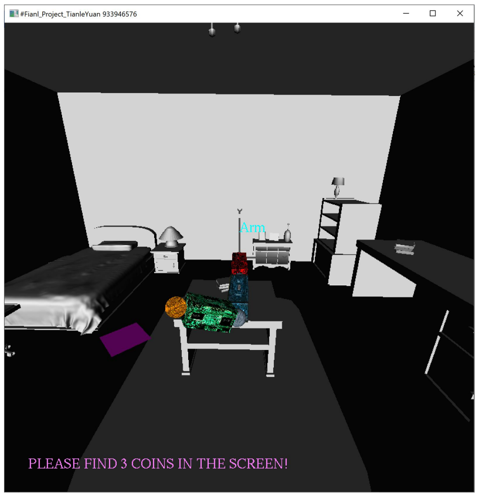

#Fianl_Project_TianleYuan 933946576
Arm
PLEASE FIND 3 COINS IN THE SCREEN!

Figure 5. The textures of joints, arms, and bulb **Figure 6.** The degree of freedom of the arm system

In **Figure 6**, we can see that player can also use "W" and "S" on a keyboard to raise up and lay down the third arm so that the bulb light at the end of the arm can hit everywhere in the room.

The third part is the soul of the game. The project has applied **rand()** function in C++ to create 3 random coins every time the program runs. In order to get the quality like gold, the program set the material of the coins with high shining value and deep orange-yellow color.

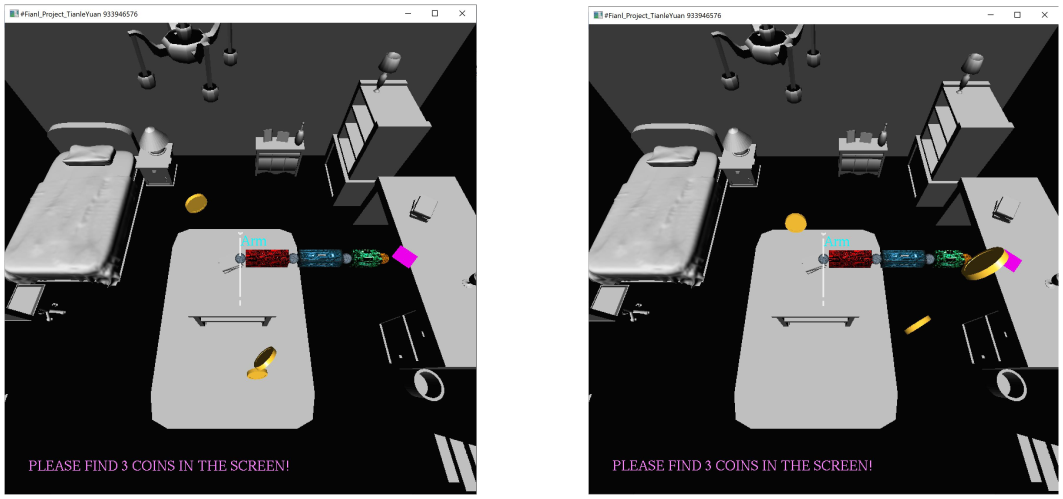
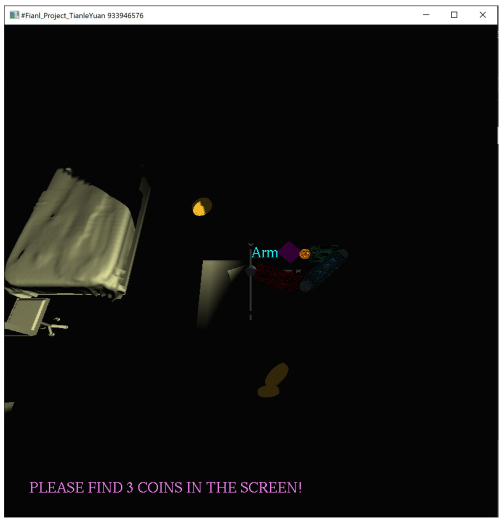

#Fianl_Project_TianleYuan 933946576
Arm
PLEASE FIND 3 COINS IN THE SCREEN!
#Fianl_Project_TianleYuan 933946576
Arm
PLEASE FIND 3 COINS IN THE SCREEN!

Figure 7. Three coins' property of randomness (the program has run twice)

The fourth part is the scene light and the game fog. There are two light sources in the game. One, the spotlight used to simulate the flashlight effect, is at the position of the bulb. The other, the point light used to simulate the effect of game fog, is at the top of the house so that players can see the whole house while cannot clearly see all the contents in the house. As shown in **Figure 8** and **9**, the flashlight is set as light warm yellow color and the "fog light" is set as dark cold blue color. When is hit by the flashlight, the place's default blue color will be neutralized and finally shows the material's gray color.


#Fianl_Project_TianleYuan 933946576
Arm
PLEASE FIND 3 COINS IN THE SCREEN!

igure 8. When there is only a flashlight source

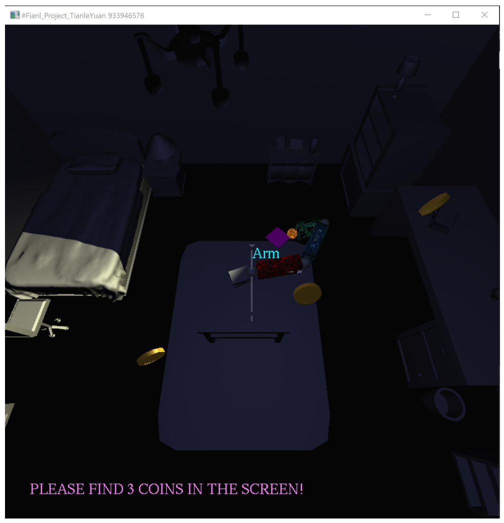

#Fianl_Project_TianleYuan 933946576
Arm
PLEASE FIND 3 COINS IN THE SCREEN!

Figure 9. When the fog light source is also added

3. How your project differs from what you proposed, and why

There are two places that my project differs from what I proposed.

The first place is the method used to find the solution of arms' rotation degree. Originally, I wanted to use the Steepest Descent method to solve the mathematical formula expressed below:

$$\begin{cases} \min\limits_{\theta_1,\theta_2} [L_1\cos\theta_1 + L_2\cos(\theta_1+\theta_2) - x_0]^2 + [L_1\sin\theta_1 + L_2\sin(\theta_1+\theta_2) - y_0]^2 \\ \|\text{point}_{\text{tail}} - \text{point}_{\text{cursor}}\| < (L_1+L_2),\ \text{or}\ \|\text{point}_{\text{tail}} - \text{point}_{\text{cursor}}\| > (L_1-L_2) \end{cases} \quad \textbf{(1)}$$

However, this method runs slow and not precise because of numerical property. Thus, the project finally used the CCD method, which can be thought of as a greedy algorithm. Even the CCD method needs several steps to get the final solution, it works faster and more precisely than methods like the Steepest Descent method, Newton method, and Quasi-Newton method.

The second place is the game mechanics. Originally, I just set that players can explore the room with only one light source in the room. However, considering the interestingness and vraisemblance of the game I added the random-showing coins and the game fog with the second light source.

Furthermore, there are some details also have been changed such as the number of the arms and joints, the material of the object in the scene, and the cursor indicator on Z-o-X plane (the purple square).

4. (optional) Any impressive cleverness you want us to know about

There is a good idea to create game fog. In the game angle, a dark space doesn't mean players cannot see anything in the space but means players can see the boundary of objects rather than their details. Thus, the existence of game fog is really important for a horrible game.

"Neutralization" is the simplified idea that I provide for this game to implement the game fog. As real life tell us, the cold light blend with the warm light can help objects to show their own color best. Thus, when is hit by the warm flashlight, the place's default blue color will be neutralized and finally shows the materials' original color. Players can see the objects in the space but cannot clearly locate them, so they have to use the flashlight's light to make sure of their location.

5. What you learned from doing this project (i.e., what you know now that you didn't know when you started)

Now I know deeply about the theory of the robotic arm mechanism. CCD algorithm is the common algorithm frequently used in the field of computer animation. It just starts from the physical theory of arms itself, applied the property of matrix operation, and simplified the transformation of different coordinates between different arms.

Also, I have learned how to get random numbers from C++. Before using rand() function, we need to set srand() function ahead to give the start point of the random chain, which means the random function in C++ actually is just a pseudo-random.

6. Any images that are especially representative of what you did

Most of the pictures I have shown in the answers of question 2 step by step, but right now, I want to show players an instruction, how to play this game:

Figure 10: The initial scene is showing;

Figure 11: Click "Moving" in the menu to rotate the scene at a suitable angle;

Figure 12: Click "Pointing" in the menu to change the place of the purple square so that control the robotic arm;

Figure 13 & 14: After a move to the right position on Z-o-X plane, try to use "W" or "S" on a keyboard to raise up or lay down the third arm until the flashlight hits the coin that you want to find. Game Finished.

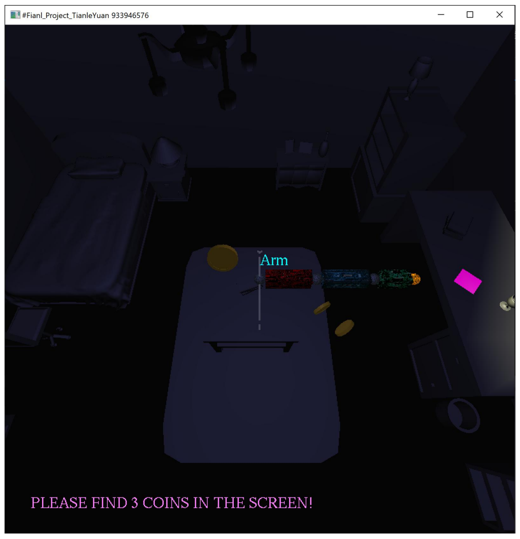

#Fianl_Project_TianleYuan 933946576
Arm
PLEASE FIND 3 COINS IN THE SCREEN!

Figure 10. Initial scene

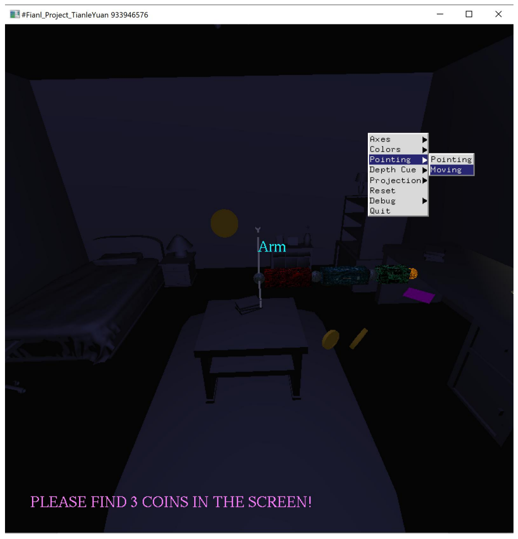
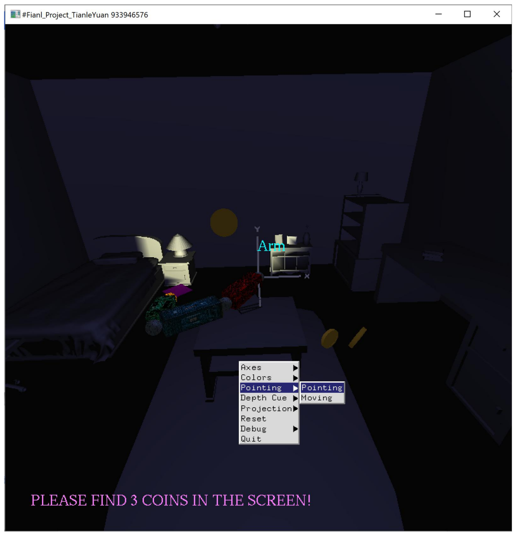
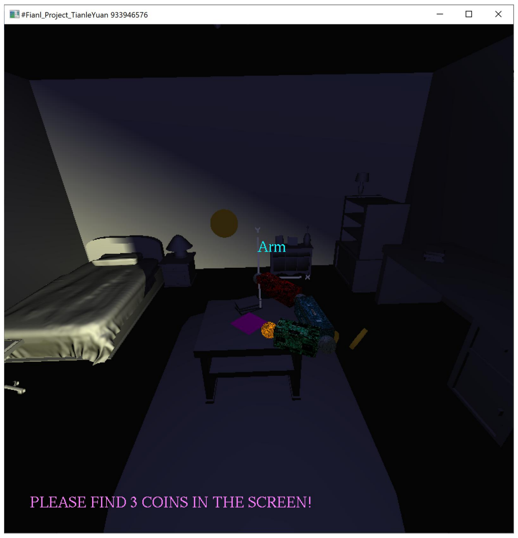

#Fianl_Project_TianleYuan 933946576
Axes
Colors
Pointing
Depth Cue
Projection
Reset
Debug
Quit
Pointing
Moving
Arm
PLEASE FIND 3 COINS IN THE SCREEN!

Figure 11. Moving the scene



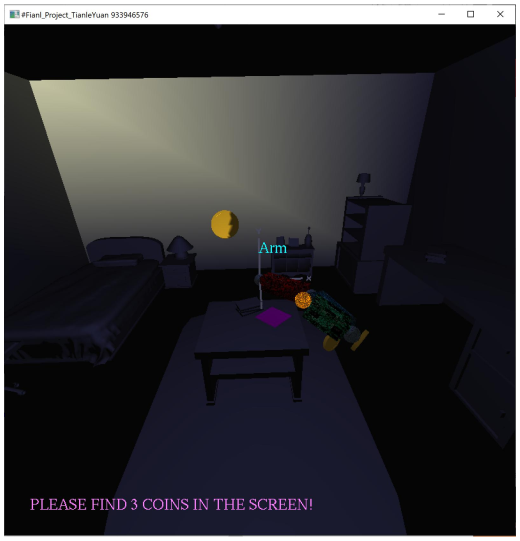
PLEASE FIND 3 COINS IN THE SCREEN!

Figure 11. Pointing in the scene



PLEASE FIND 3 COINS IN THE SCREEN!

Figure 12. Changing the position of purple square



PLEASE FIND 3 COINS IN THE SCREEN!

Figure 13. Pushing "W" or "S" to hit the object

7. A link to the video showing off your project - be sure it is unlisted.

https://media.oregonstate.edu/media/t/1_gdr0wrgl