# CS460 - Final Project Report

# PIXAR Lamp - LUXO

Selin Dinc – Sinem Ozden

## Abstract

The Luxo project's aim is to recreate the Pixar's iconic opening. In this project, we made some adjustments in the scene while recreating the animation. Instead of making the lamp jump on the "I" letter in the "PIXAR", we wanted to use the lamp as "I" letter. Also, in our version we changed the color of the letters and the light. To recreate the animation of the iconic lamp, we used openGL and C++. This paper gives detailed information about the steps to recreate the animation of Pixar's opening in our way for the project Luxo.

## Introduction and Literature Review

This project aims to create a lamp object that can do various actions that can create an animation. While we were researching how to create the lamp, we realized that people mostly preferred to create the lamp object by using the triangles according to the Bresenham's Algorithm and creating their own algorithms instead of using the openGL functions. Another option to create the lamp was using an object which created previously. However, in our project we wanted to design our own lamp. Thus, we decided to make the lamp object by using the openGL's functions.

Our planned movement sequence has some differences compare to the original animation of Pixar. In the original animation the lamp enters the screen from the right side, then moves until it sees the “I” letter and jumps on top of it until it replaces the “I” letter. When the “I” letter is gone the lamp looks towards the camera and the animation ends. In our version, lamp jumps from right side of the screen until it sees the missing “I” letter then lamp move towards the empty spot and turns its head till it looks towards the camera.

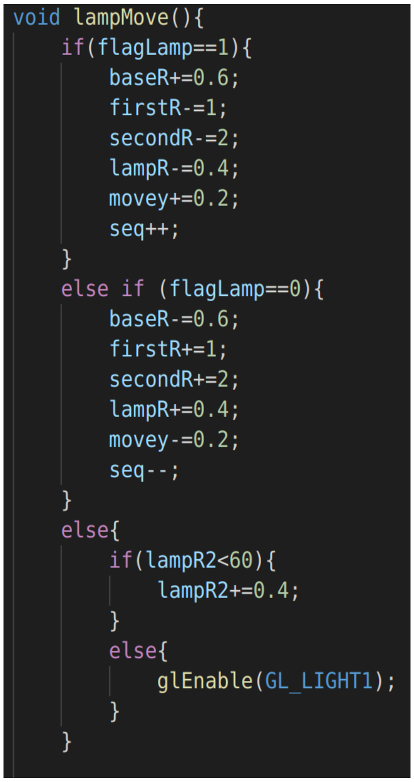
For the Luxo’s movement we found a paper from the Luxo’s own creators that explains the process of creating 3D computer animations based on 2D hand drawings in a detailed way. The paper explains some key principles while creating 3D animations and make them look realistic. We used these principles especially while creating the squash and stretch movement and planning the timing.

## Method and Algorithms

Our project consists of three main parts. The first part is to draw the lamp and move it, the second part is to create the 3D environment and the lighting, the third and the last part is to draw the “P_XAR” in the 3D environment. We mainly used openGL functions for lamp creation, lighting, background creation and movement.

To create the lamp, we generally made use of the third homework. We divided the lamp into segments to deal with the movement easier and draw the lamp easier. Each segment of the lamp has a different function to draw them. These functions draw the segments and all these functions are called in the display function. We created all the parts of the lamp with gluCylinder function. We first created the base, then the two body parts and the lamp part. Base has the plate and a stick to hold the joint. The plate consists of a shallow cylinder and a disk to cover the top of the cylinder. The both body parts have dual vertical struts and a single joint. The head of the lamp has a joint and a stick to hold it the head part. The head part consists of a cylinder and a cone. The head part also has a sphere that looks like a light bulb. The red-light source located where the sphere is. The difference between our lamp and the original “Luxo Lamp” is we do not have the springs because it would complicate the animation process. Also, the shape of our joints are cylinder instead of triangle.

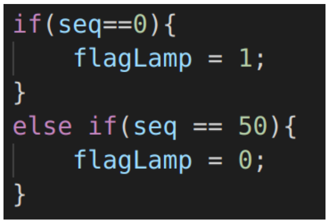
To redisplay the moved lamp we used the glutTimerFunc function. This function recalls the display function as specified time passed. We defined this function like we defined the display function in main.cpp. The main function creates a callback for timer and calls the timer function that we created after the specified time passed. The timer function calls the glutRedisplay function which recalls the display function again with the updated angles and coordinates.



```
void lampMove(){
    if(flagLamp==1){
        baseR+=0.6;
        firstR-=1;
        secondR-=2;
        lampR-=0.4;
        movey+=0.2;
        seq++;
    }
    else if (flagLamp==0){
        baseR-=0.6;
        firstR+=1;
        secondR+=2;
        lampR+=0.4;
        movey-=0.2;
        seq--;
    }
    else{
        if(lampR2<60){
            lampR2+=0.4;
        }
        else{
            glEnable(GL_LIGHT1);
        }
    }
```

Image 1: The folding and unfolding motion. The image shows how the angles of segments are changed. (folding and unfolding motion.)

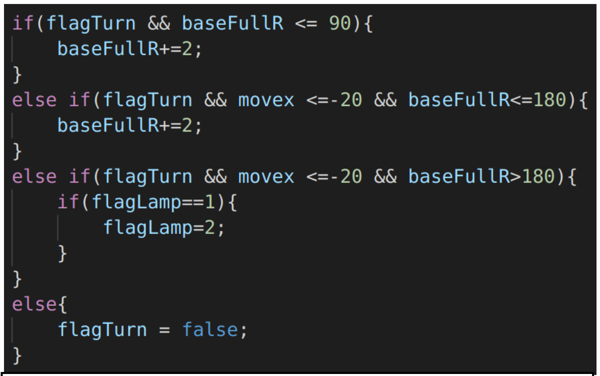
The lamp has three main movements. The first movement is jumping, the second is moving forward and the third is rotating. According to the Lasseter’s paper “Principles of Traditional Animation Applied to 3D Computer Animation” to create a realistic jump movement “squash and stretch” principles should be used. The squash movement of the lamp is folding, and the stretch movement of the lamp is unfolding. We used the Luxo as an example and tried to mimic the jumping movement. For jumping we created a global sequence variable. We gradually increased or decreased the angle of the segments to create the folding effect of the lamp. Every time the display function is called the display function calls the lampMove function which arranges the folding effect of the lamp. If the sequence hits zero the lamp starts the folding movement by increasing the angles of the segments and increasing the sequence variable, when the sequence hits 50 the lamp starts the unfolding movement which decreases the angles and the sequence variable. With sequence an infinite loop is created because the sequence integer is constantly increasing or decreasing as long as the flag is still valid. We changed the flag only when lamp had to stop.



```
if(seq==0){
    flagLamp = 1;
}
else if(seq == 50){
    flagLamp = 0;
}
```

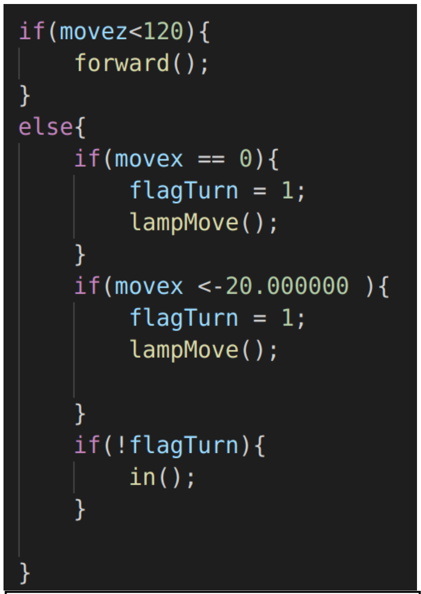
Image 2: The sequence and arranging the flags.

To move forward we simply moved the base plate forward. However, we needed to consider the timing principle for a realistic movement. For rotation we give specific angle (90) to base when the lamp changes its direction. When the lamp's head rotate, we gave 70 degrees to make the lamp's head part rotate and look towards the camera.



```
if(flagTurn && baseFullR <= 90){
    baseFullR+=2;
}
else if(flagTurn && movex <=-20 && baseFullR<=180){
    baseFullR+=2;
}
else if(flagTurn && movex <=-20 && baseFullR>180){
    if(flagLamp==1){
        flagLamp=2;
    }
}
else{
    flagTurn = false;
}
```

Image 3: The rotation algorithm.



```
if(movez<120){
    forward();
}
else{
    if(movex == 0){
        flagTurn = 1;
        lampMove();
    }
    if(movex <-20.000000 ){
        flagTurn = 1;
        lampMove();

    }
    if(!flagTurn){
        in();
    }

}
```

Image 4: The forward and inward movement.

At some point lamp turns and moves in a different direction. To arrange this, when lamp moved forward 120 times, we stopped the forward movement and rotate the base. Then start the movement toward the inner screen (z axis). After this movement continues for 20 times, we stopped the lamp's movement and turned the top part for 70 degrees.

We also decided to add a light bulb to the lamp. To make it we used a directional light with a red light. Light moves as the base moves and because it’s a directional light it only lights up the place that in front of the light. With this additional lighting of the lamp we currently have two light sources at the environment.

While creating the lamp we also started to set-up the background and the environment of the lamp. For the 3D environment we draw a wall and a floor by using glNormal, glVertex. For the wall we also used glutSolidCube function. Then we set the gluLookAt function to make the angle that was appealing to the animation. However, this created a problem with the size of the lamp. To fix it we scaled the lamp by using glScale as we pushed the matrix. We also tried to change the lighting to make the environment brighter and shadows defined.

For the last part we draw the “P_XAR” in the 3D environment. We decided to push and pop a new matrix every time as we start a new letter. We also draw the “P_XAR” letters by using gluCylinder. For this part we could have used some models that created previously but we decided to draw the letters ourselves.

We also draw the x, y, z axis with different colors as a reference while creating the environment. However, after the development process finished, we comment out this part for a better animation.

## Results

As a result, since we have some differences from the original animation our project became a version of our own.  Firstly, we decided to change the jumping movement of the lamp above the “I” letter of the “PIXAR”. Instead of doing that, we got rid of the “I” letter and simply relocate our lamp between the “P” and “X” letters by animating it.  This decision resulted in removing the bouncing movement of “I” letter from our project.

Another difference of our project is we did not want to use previously made models for the letters. We draw the “P_XAR” by using gluCylinder like we draw the lamp. Since we draw the letters with gluCylinder we did not able to have a decent font for our letters. It made our environment a little bit less similar to the original one.

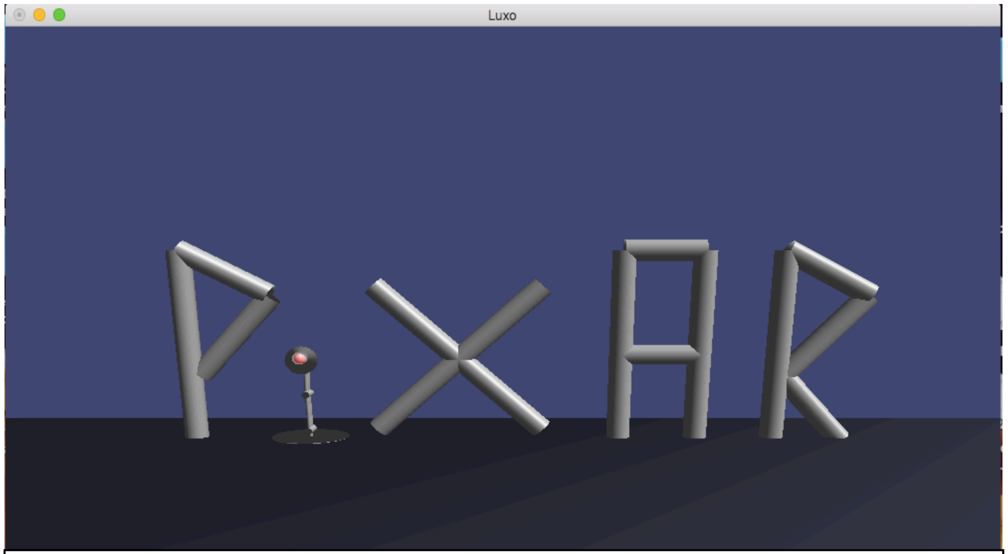


Image 5: The final scene of the animation.

Our lamp is visually similar to the famous Pixar lamp “Luxo” with a few design changes such as not having springs and different shape of joints. Also, the movement of our lamp has a similarity with the “Luxo” because we applied the 3D animation principles to our lamp. The lamp fold while jumping and unfolds while landing.

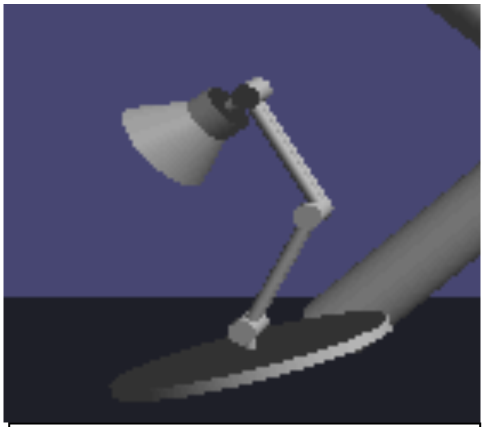


Image 6: The lamp.

We decided to make our lamp move slower compare to “Luxo” because we wanted to emphasize our movements instead creating a totally realistic and fast movement.

## Conclusion and Discussion

As a conclusion, we started the project with the intention of creating the same animation of the Pixar's, but throughout the process we created our own version with some differences compared to the original one like we mentioned in detailed above. In the development process we try to solve the problems in our ways, this is probably the main reason that we have differences compare to the original animation. In the end, we almost meet with the all requirements that we stated before in our proposal.

## References

*John Lasseter. 1987. Principles of traditional animation applied to 3D computer animation. In Proceedings of the 14th annual conference on Computer graphics and interactive techniques (SIGGRAPH '87). Association for Computing Machinery, New York, NY, USA, 35–44. DOI:https://doi.org/10.1145/37401.37407*

*Anon.Retrieved May 16, 2020 from https://www.glprogramming.com/red/*