Cinedemo

Computer Graphics Project 7: Cinematography

20171678 이찬희

Target Users

> Camera director who wants to test their cinematography and create a demo animation in a virtual stage.

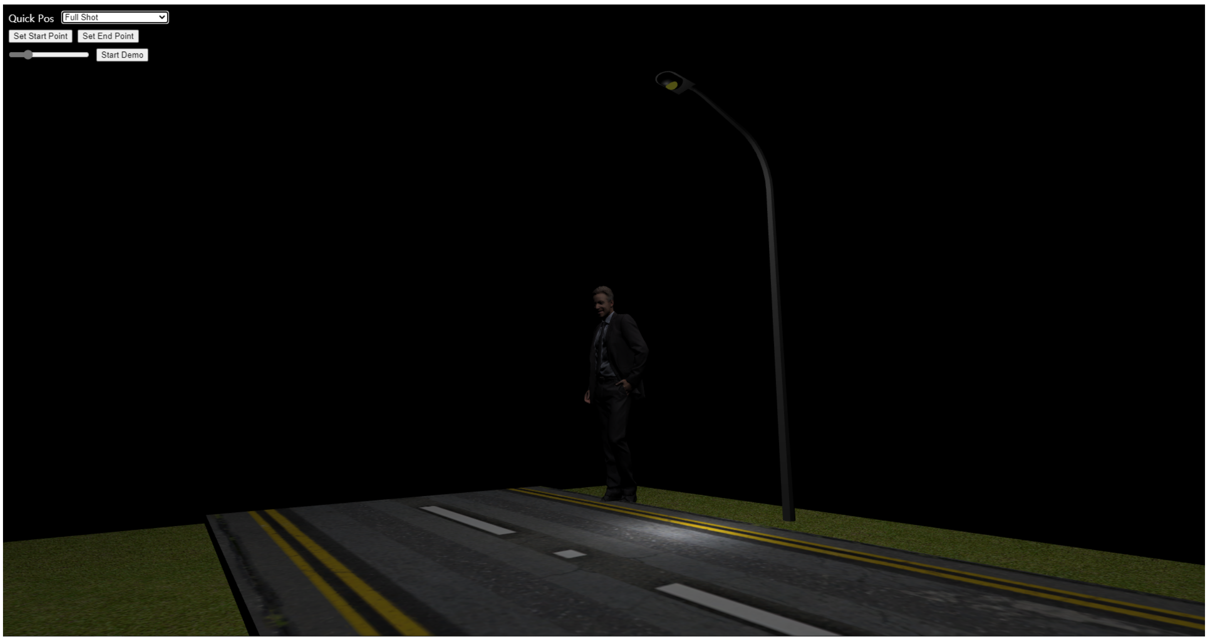
Features

- There is an virtual scene of a street lamp and a person standing next to it on the side of the road.
- User can move, pan, tilt camera using arrows and 'w', 'a', 's', 'd, 'u', 'j' key.
- User can select camera from presets at the top left select box.
- Users can use buttons and sliders in the upper left corner to set the camera's starting position, ending position, and the length of the animation before running the demo animation.
- When a user clicks the 'Start Demo' button, the camera moves along a set path for a set time to test the camera's movement.

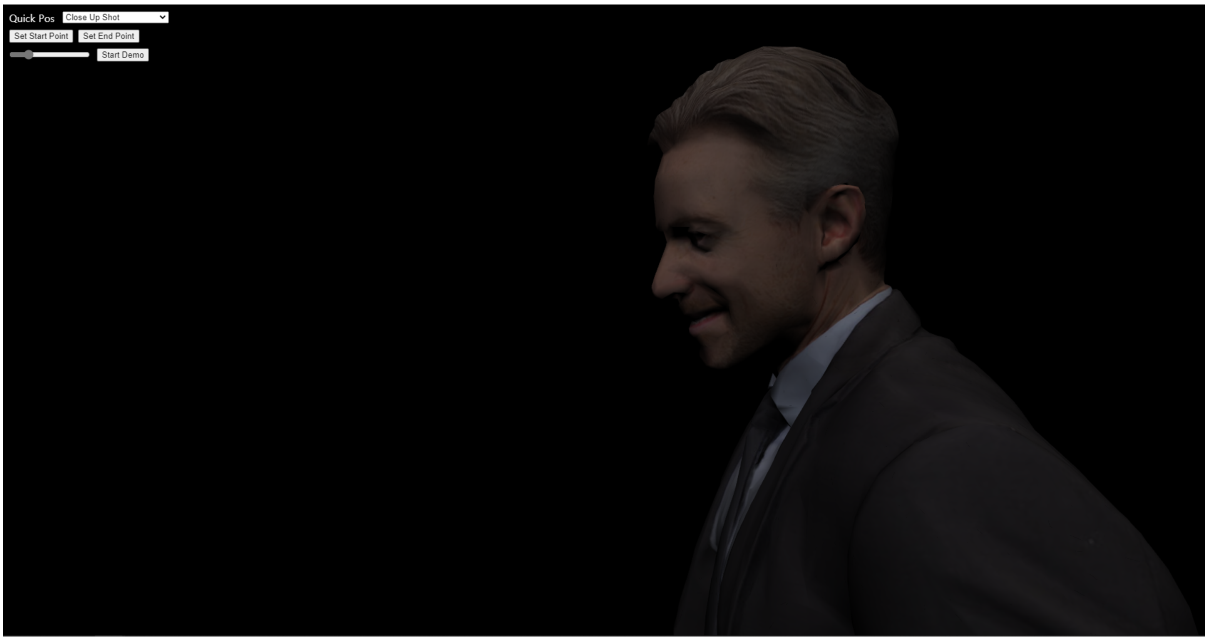
Concept and Idea

I thought it would be nice to have a software that could check the movements of the camera in advance when filming a movie or producing an animation. Since I thought it was important to check the scene through moving animation, not just stationary images, I devised a program that allows me to move the camera in advance by setting the start and end positions of the camera.

Screen Shots



Quick Pos: Full Shot

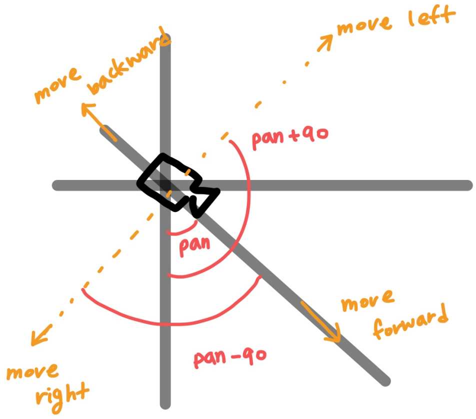


Quick Pos: Close Up Shot

Algorithm & Code

Camera Movement

The motion of the camera is implemented by adding several functions to the example code covered in class. First of all, keymapping is implemented in a way that I think is more intuitive.



The left-right movement of the camera on the xy plane was calculated using the pan angle, and I added adjustments to the tilt angle on the z-axis.

In addition, the ability to move the camera on the z-axis was added.

Animation Demo

```
if (cameraIsMoving) {
        if (animationLength == animationCount) {
            cameraIsMoving = false
        } else {

            cam_x = getMidPoint(startPos.cam_x, endPos.cam_x)
            cam_y = getMidPoint(startPos.cam_y, endPos.cam_y)
            cam_z = getMidPoint(startPos.cam_z, endPos.cam_z)
            pan = getMidPoint(startPos.pan, endPos.pan)
            tilt = getMidPoint(startPos.tilt, endPos.tilt)

            updateCamCenter()
            animationCount += 1;
        }
    }

function getMidPoint(start, end) {
    return start + (end - start) / animationLength * animationCount;
}
```

We used the CameraIsMoving variable to determine whether the animation is currently running. If the animation was running, the current camera position was calculated.

Link to the project

https://editor.p5js.org/chan2ie/present/Nh6fE7YmO