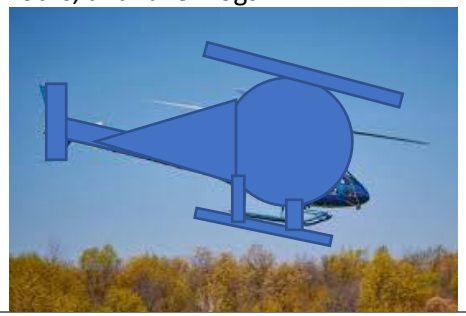
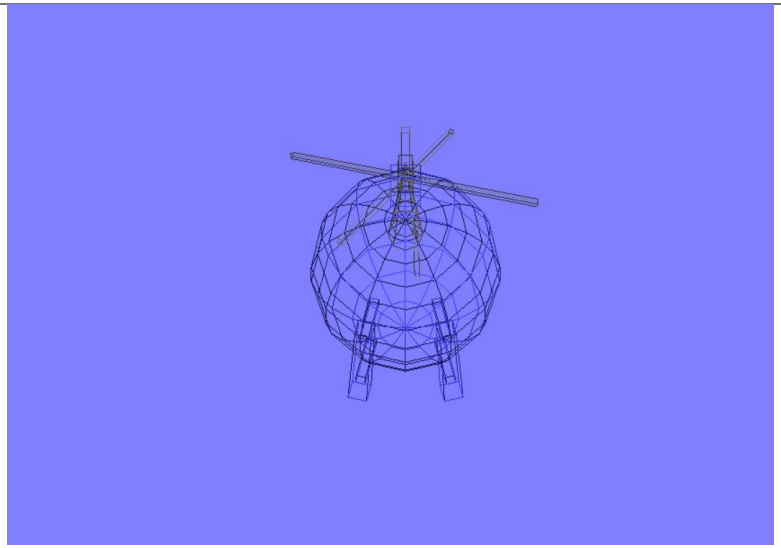
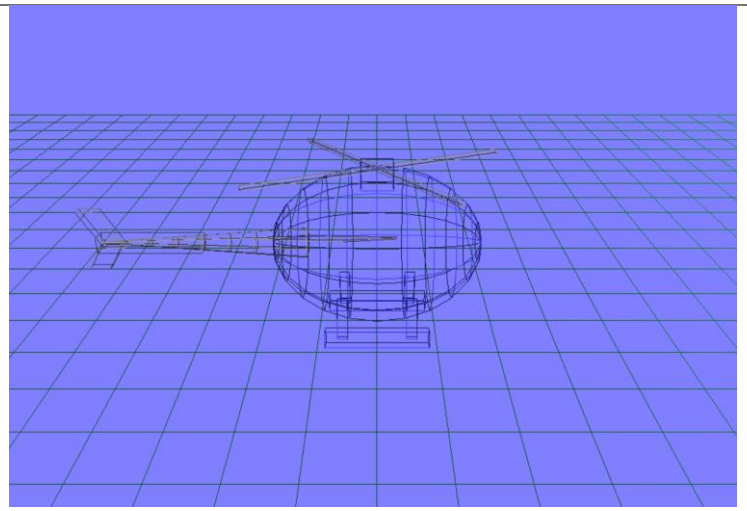
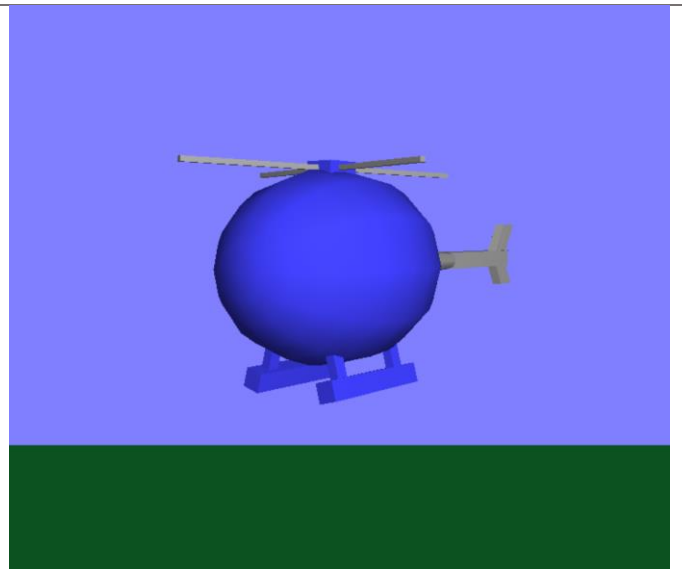
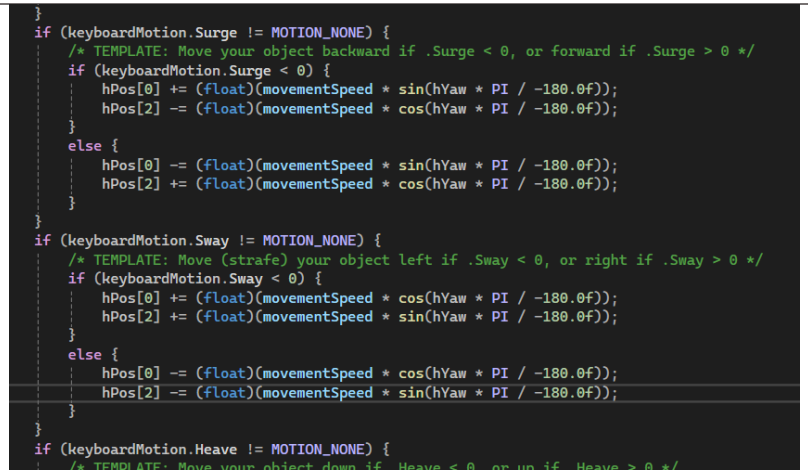
Logbook:

Time Spent	Date	To do	Work done	Detail	Bugs
8am – 9:30am	27/08/24	☒Plan elements	Planned out helicopter hierarchical model and other needed models for the complete scene.	Chose to use the helicopter model below as the reference for my model design. I also chose to use 1 GLU unit as a meter in the scene which will make it clear how big to make the models. I have decided to go for a garden/forest theme with a pond, trees, rocks, and fallen logs. 	

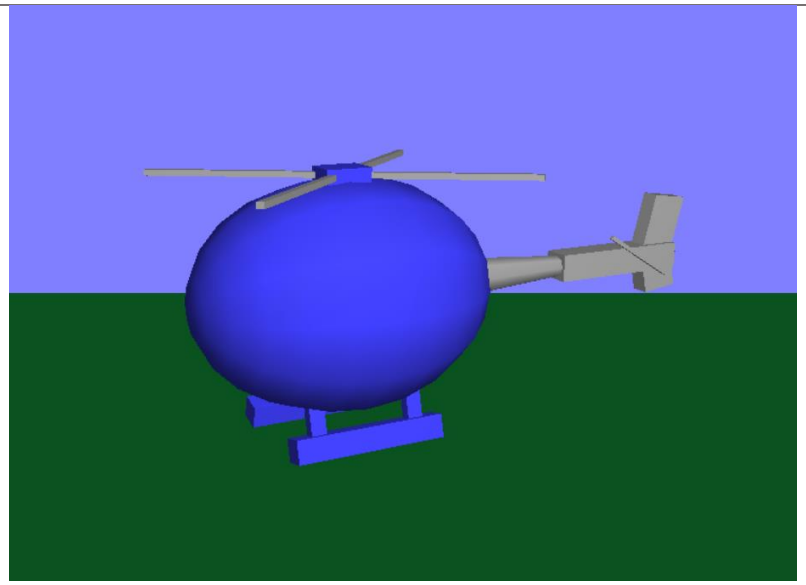
8am – 12pm	17/09/24	☐Create hierarchical Model for helicopter	Made the hierarchical model for the helicopter. Added in rotor angles to mimic movement.	 Created the hierarchical model of the helicopter as follows: Helicopter base / \ \ Top rail Front tail part Bottom rail legs / / /\ Rotor 1 and 2 Back tail part Bottom rails left and right / \ Back wing top and bottom Back rotor Based on this hierarchical model, I created the helicopter using solid cubes, cones, and quadric meshes. While creating the model, I came across an issue with the back rotor skewing to the side as if it was constrained by a rectangle. I attempted to undo rotations, scaling, and transformations that were before it to see if that fixed the issue, but the skewing was still present	Issue with back rotor skewing when added into the hierarchical model (under tail object)

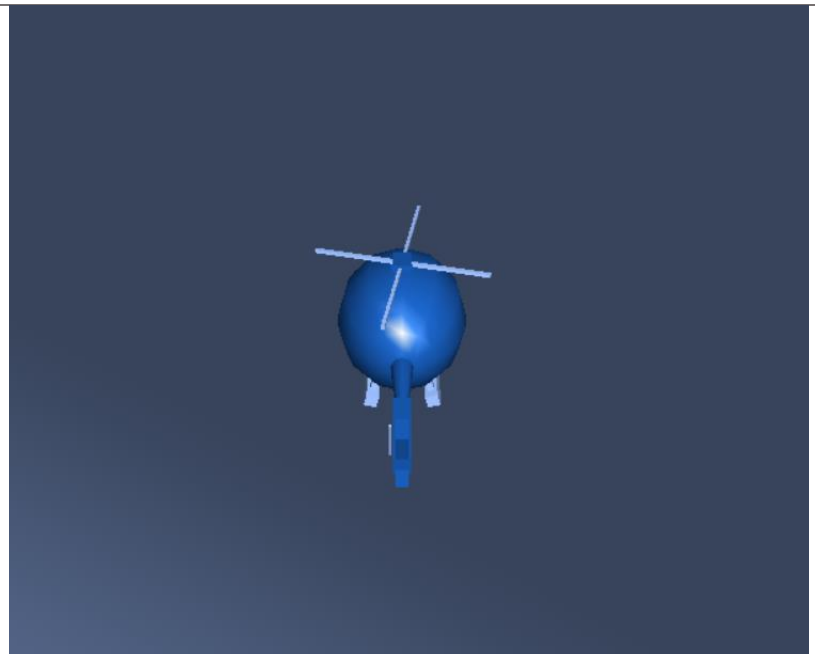
12pm – 1pm	18/09/24	☒Create grid ground	Created function to draw grid with custom grid size and total size.	 I created a function for drawing the grid both wired and solid using GL_QUADS. I then used glPolygonMode with both GL_FILL and GL_LINE to draw the grid based on the current renderFillEnabled variable.	

8am – 10am	24/09/24	☐Get movement keys working fully.	Added movement keys. Looked at ways to move the helicopter in multiple directions as well as the correct direction based on the yaw.	 Created the basic needed movement for the helicopter and started looking at ways to move the helicopter in multiple directions. One of the ways I found was similar to the way the camera could be manipulated to follow the helicopter in the third-person camera slides. heliZ + cos heading ∗ distanceXZ heliX + sin heading ∗ distanceXZ	

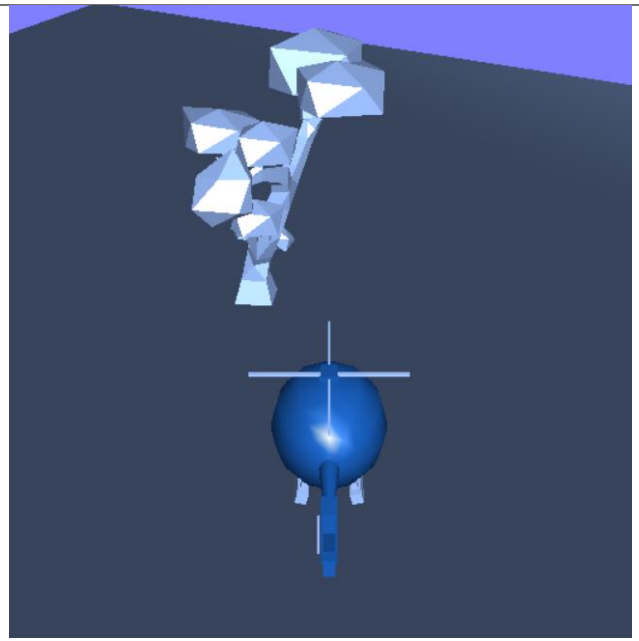
2pm – 5pm	26/09/24	☒Get movement keys working fully.	Added a way to get the helicopter to follow the correct movement when rotated (had to change to -180 as normal 180 was doing the opposite	 I added in the equation within the slides mentioned before and slightly adjusted them to work with the helicopter. The heading was the hYaw changed into radians and the distance became the total movement I wanted for the helicopter to move. I ran into a small issue with the helicopter moving in the opposite direction I needed it to, so to fix that, I changed the 180 to -180 which solved the issue.	

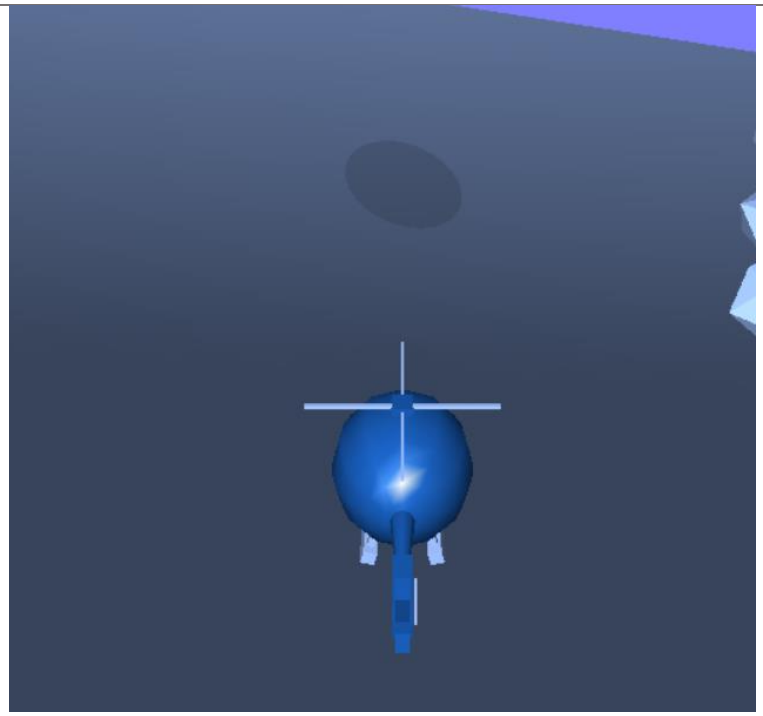
```
}
if (keyboardMotion.Surge != MOTION_NONE) {
    /* TEMPLATE: Move your object backward if .Surge < 0, or forward if .Surge > 0 */
    if (keyboardMotion.Surge < 0) {
        hPos[0] += (float)(movementSpeed * sin(hYaw * PI / -180.0f));
        hPos[2] -= (float)(movementSpeed * cos(hYaw * PI / -180.0f));
    }
    else {
        hPos[0] -= (float)(movementSpeed * sin(hYaw * PI / -180.0f));
        hPos[2] += (float)(movementSpeed * cos(hYaw * PI / -180.0f));
    }
}
if (keyboardMotion.Sway != MOTION_NONE) {
    /* TEMPLATE: Move (strafe) your object left if .Sway < 0, or right if .Sway > 0 */
    if (keyboardMotion.Sway < 0) {
        hPos[0] += (float)(movementSpeed * cos(hYaw * PI / -180.0f));
        hPos[2] += (float)(movementSpeed * sin(hYaw * PI / -180.0f));
    }
    else {
        hPos[0] -= (float)(movementSpeed * cos(hYaw * PI / -180.0f));
        hPos[2] -= (float)(movementSpeed * sin(hYaw * PI / -180.0f));
    }
}
if (keyboardMotion.Heave != MOTION_NONE) {
    /* TEMPLATE: Move your object down if .Heave < 0, or up if .Heave > 0 */
```

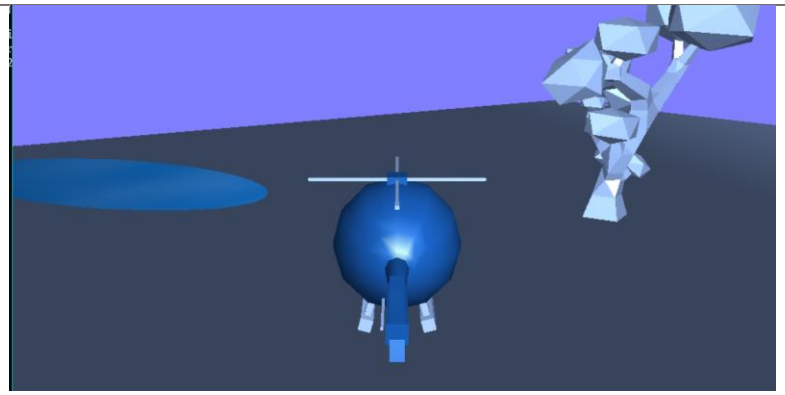
10am – 12pm	27/09/24	☒Create hierarchical Model for helicopter	Fixed issue with rotor skewing	
				After much trialling with trying to return the rotations back to 0 and the transformations, I realised the issue was with the overall scaling of the object. To fix this, I divided each previous scaling factor by 1.0 which returned the scaling back to normal and stopped the rotor from skewing on the helicopter.
				`}`
				`else {`
				`  glutSolidCone(4.0f, 6.0f, 8, 8);`
				`}`
				
				`glScalef(1.0f / 0.3f, 1.0f / 0.3f, 1.0f / 2.0f); //reset scaling back to 1, 1, 1 (fixes back rotor skewing issue)`
				
				`//tail back`
				`glPushMatrix();`
				`  glTranslatef(0.0f, 0.0f, 9.0f);`
				`  glScalef(1.0f, 1.0f, 4.0f);`	

12pm – 4pm	1/10 /24	☐Allow for the helicopter to Take off and land gradually.	Added TO and Landing rotor speeds, ground detection, and a delayed TO for the rotor speeds to increase.	 For the ground, I decided to stop the helicopter at 2.4 at this was when the model was just touching the grid. For the landing of the helicopter, I decided on using the y position of the helicopter to gradually slow down the descent and the rotors as the helicopter landed. This was done with a variable called descentFactor. float descentFactor = (hPos[1] - 2.4f) / (5.0f - 2.4f); This took the y position and the ground position and took those away from each other. Then it took the y position that would start the descent and the ground position and also took those away from each other. The variable then divides these two to give a small	Issue with TO only happening when the program is first run and not when the helicopter lands again during the play through

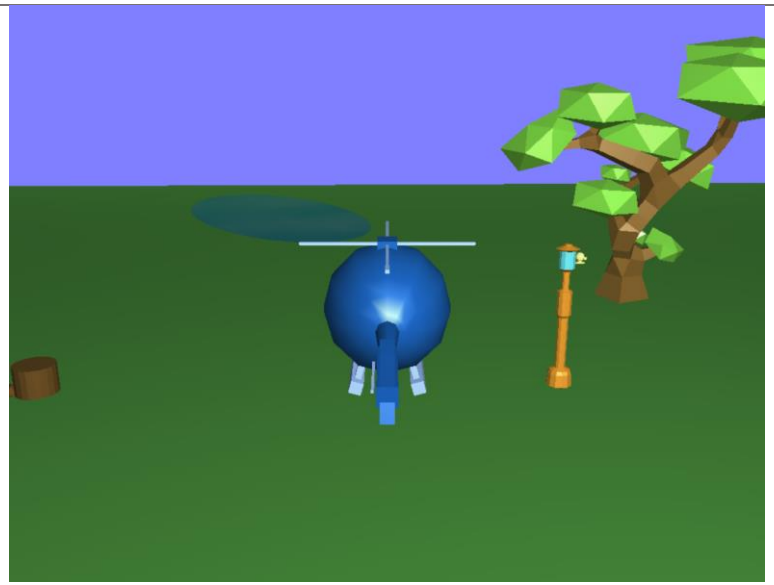
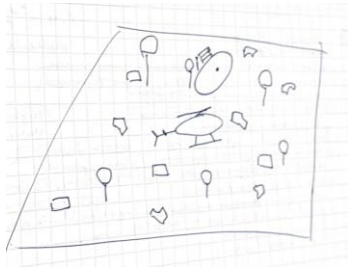
				decrement for the rotors and movementspeed. When the helicopter gets to a certain position (5.0), the rotors and movement gets slower - rotorAngle += 360.0f * FRAME_TIME_SEC * descentFactor; - until the helicopter reaches the ground (2.4) I attempted to use the same for the take off, but the descent factor wont work with this as it relied on the y position of the helicopter to gradually decrease the number. Instead I created a rotorLiftOffSpeed variable that defined how fast the rotors needed to be turning to allow the helicopter to move up. Based on this variable, in the up movement code, I added a section for when the y position of the helicopter was less than or equal to 2.4(ground) and also the rotorAngle of one of the top rotors was less than the rotorLiftOffSpeed. If these conditions were met, the rotors would increase by 50 * FRAME_TIME_SEC every frame until those conditions were no longer met. This seemed to work as the helicopter would start on the ground. However, when the user would land the helicopter any other time and try and take off again, the helicopter would ignore that code and immediately lift off.	

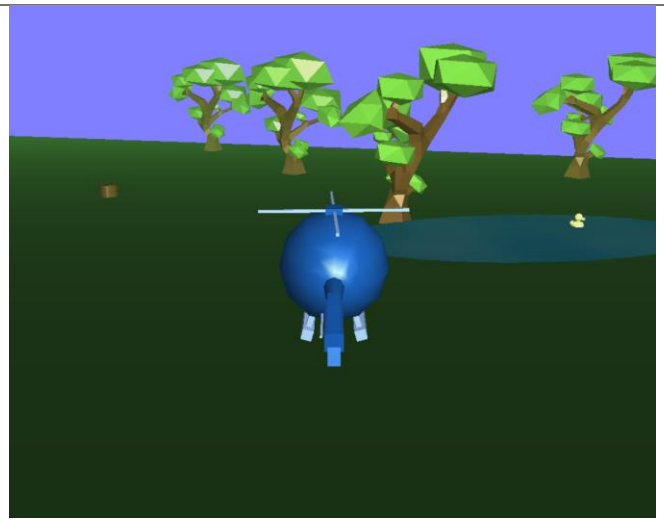
3pm – 5:30pm	5/10 /24	☐Create the environment	Added the OBJ file parsing code to allow for OBJ files to be rendered in the scene. Added a tree obj file to test.	 As I have experience with creating obj models, I decided to use some for the scene. I used the OBJ file parsing code within the lectures with some parts changed to render the obj files in. I tested this with an obj file of a low polygon tree I had made and found the code had worked.	

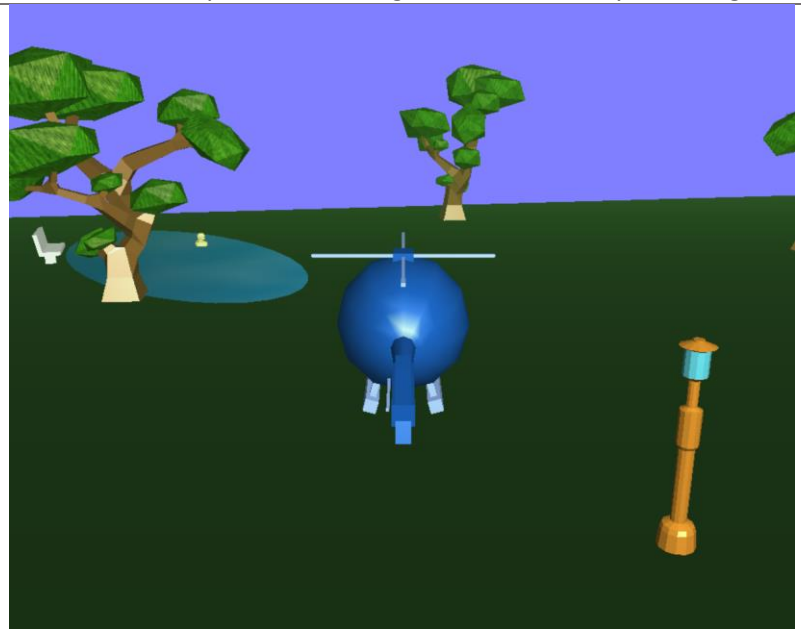
10am – 11:30am	7/10/24		Want to add a pond, lamp post, bench, rocks, trees, and logs to the scene. Looking at using the same code used for circle drawing in last assignment and changing it to draw an oval instead.	 For the complete scene, I wanted to add in a small pond. I found a parametric equation for an ellipse online (https://www.mathopenref.com/coordparamellipse.html) and imported that into my previous circle function from the snowman project. The parametric equation of an ellipse is: $x=h+a\cos t$ $y=k+b\sin t$	

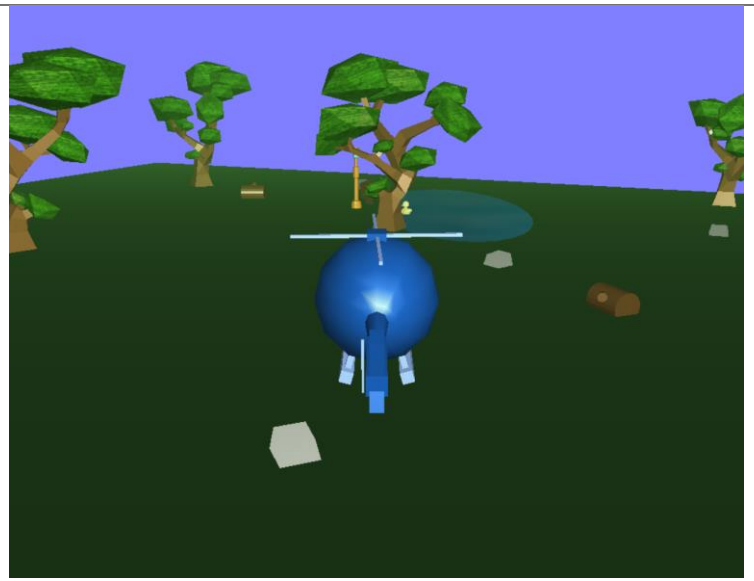
				where h and k are the offsets for the position of the ellipse, a and b are the radius's, and t is an angle used for the ellipse. I inputted this equation into a for loop with 30 parts for a round edge and used the triangle fan option to draw the oval	
10am – 4pm	11/10/24	☐Add texturing	Added texturing to oval	 With the oval drawn, I then attempted to texture the oval. This was difficult as many of the images I had found online wouldn't convert properly to a ppm file. As well as this, I found that the way I had decided to do the image texturing was taking up a massive amount of memory which was causing the program to get many errors. I had initially used this - GLubyte myTexture[woodW][woodH][3]; - to map the image to the texture using a similar code to the ones shown in class. However, I kept getting issues saying: Severity Code Description Project File Line Suppression State Details Warning C6262 Function uses '1440016' bytes of stack. Consider moving some data to heap.	

				Because of this, the program would often crash without loading anything. To fix this, I searched what I could do to switch it to a heap and found https://www.codecademy.com/resources/docs/c/memory-management/malloc Using this, I switched the way I loaded the texture to *GLubyte* myTexture = (GLubyte*)malloc(woodW * woodH * 3 * sizeof(GLubyte));* which stopped the errors and the crashing of the program. However, I was still having issues with converting the images to ppm. I ended up downloading infranview to convert them and ran into the same issue until I change the covert settings to ASCII encoding. Once this way done, the images would all load correctly and would texture properly to the oval.	

1pm – 2:30pm	12/10/24		Added materials into scene and additional obj files	 I started adding simple materials into the scene as a base to build on later once the lighting had been added. All other obj file meshes were rendered in and I started looking at the placement of each item so that it would be more like a foresty garden 	

1pm – 3pm	15/10/24	☒ Animate an object in the scene	Looked at ways to animated the duck on the pond to move in an oval
			
			https://www.mathopenref.com/coordparamellipse.html
			
			Added the equations in and edited them to fit the oval pond.
			
			Added a display list for the trees and a floating aspect for the duck to change the y to mimic floating.
				
				
				For an animated object, I wanted the duck mesh to float around the lake.
				To do this, I used the same equations I used to draw the ellipse.
				duckX = (float)(a * cos(t) + 30.0f);
				duckZ = (float)(b * sin(t) - 7.5f);
				In these equations, t was the FRAME_TIME_SEC divided by 5 and a and b were the radiuses.
				
				To get the duck facing the right way when moving, I made the ducks angle t converted into radians (-180 / PI).
				
				I also wanted to make the duck look like it was floating on the ponds surface, so I wanted to alter the y axis every now and again.
				
				To do this, I created a variable called duckFloating, which checked if the duck was up((1)currently down) or down((0)currently up).
				If the duck was floating, I changed the Y by adding FRAME_TIME_SEC /10 and then checked if the duck was at the y	

				limit. If it was, duckFloating changes to 0 and takes FRAME_TIME_SEC / 10 away from the y axis. As well as animating the duck, I also added in a display list for the multiple trees that would be needed in the scene. I created the display list and called it within init and then drew all the trees. Each tree has its own position and angle that could easily be changed.	
10am-12pm	16/10/24	☒Add texturing	Added in texture mapping for the leaves on the trees	 With the trees in position, I decided to change the leaves material to a texture instead. I created the leaf texture and bound it to the tree leaves in the draw tree function. As well as this, I also make the colour underneath a brighter green as the original texture was quite dull.	

12pm – 4:30pm	18/10/24	☒Allow for the helicopter to Take off and land gradually. ☒Create the environment	Fixed take off and landing by adding a rotor running speed. Added wood texture to bench Added display lists for rocks and logs.	 By using a rotorRunningSpeed variable, I was able to fix the issue with the helipcopter only taking off when the scene first launches. At the beginning of the code, I have a variable that states how fast the rotors need to be turning to be able to take off. Underneath, I have the code from before for the helicopters descent snd normal rotor spinning. After this, I added a check within the same if else statement to see if either of the up or down keys were being pressed. If they weren't, the rotorRunningSpeed resets to 0 and all 3 of the other rotors do as well. under this, I added a small if statement to check if the helicopter was under or equal to 3.0 as well as a check to see if the movement was down. If these are met, the rotorRunningSpeed resets again. To fix the takeoff, I added a new if else statement to the up movement key. If the rotorRunningSpeed was less than the needed	

				rotor speed, the rotorRunningSpeed increases by 50.0f * FRAME_TIME_SEC and all the rotors angles increase as well. Under this, I added a check to see if beacsue of the addition to the rotorRunningSpeed, it was now equal or more thsn the needed rotor speed. If so, the rotorRunningSpeed is clamped to the rotorLiftOffSpeed and the helicopter starts to move up. The else of this statement just allows the helicopter to move up. Adding the check against the rotorRunningSpeed stopped the helicopter from not taking off after launch and fixed the issue. As well as fixing the issue of the takeoff, I also added in more display lists for the rocks and logs, and mapped a wood texture to the bench like I had done previously.	
1:30pm – 3pm	21/10/24	☒Add in all lighting	Added sunset lighting and animated spotlight to the bottom of the helicopter	 I changed the global ambient lighting to a more sunset colour as well as a directional light for the sunset as well. I also added an animated spotlight to the bottom of the helicopter pointed slightly forward towards positive Z to highlight the objects below.	

				To do the spotlight, I simply changed the x, y, and z positions to the helicopters positions every frame. As well as this, I also decreased the y axis as the spotlight wouldn't work correctly if it wasn't further down.	
10am – 4pm	24/10/24	☒Add in fog	Fixed materials and finalised specular and shininess. Added in fog.	 Each material created had its own ambient, diffuse, specular, and shininess values like these: GLfloat groundA[4] = { 0.18f, 0.35f, 0.15f, 1.0f }; GLfloat groundD[4] = { 0.23f, 0.45f, 0.19f, 1.0f }; GLfloat groundS[4] = { 0.05f, 0.05f, 0.05f, 1.0f }; GLfloat groundShine = 5.0f; I determined each by how the object would look in real like (matte, metal, wet) and converted that into the 4 options. Once these had all been created, I changed each material for each object and ensured it all worked correctly.	

				I also added in an orangeish fog into the scene for a more sunset/dusk scene and changed the colour of the sky to match  **Critical Evalutation:** During this project, I successfully implemented several key features like hierarchical modelling, rotor animations, and interactive helicopter movement, takeoff, and landing. The most challenging part for me was texturing as I encountered multiple issues, including difficulties applying textures correctly. Additionally, animation the helicopters smooth takeoff and landing was complex and challenging to do. Next time, I would focus on adding more variety to the scene and aim to texture a wider range of objects. A significant shortcoming of the application was the white screen the appears while the textures are loading, which I was unable to resolves but I hope to figure out in the future.	

Total time spent: 44 hours					