COMP612 Assignment 1 Logbook

Myles Hosken 20124720

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Time Spent | Work Done | Detail | Errors Encountered |
| 8/03/2023 | 2hrs | -Set up Assignment Project  -Linking main library  -Creating reusable circle function | -Creating a window for the scene  -Decided to go with the snow globe view for the project  Worked on a function to draw a circle at any specified position. Created a struct to hold the fields of a circle.  Created a struct to hold the values of a Colour object. Not entirely sure if this approach will be used as opposed to an array going forward. | Drawing a circle with the calculation worked only when adding the x y coordinates to the vertices drawn.    Drawing large circles left a small line inside the shape, this was fixed by increasing the value of PI to its full floating-point value. #define PI 3.141592653589793238 |
| 9/03/2023 | 1hr | -Increased functionality of the draw circle function  -Added a Boolean for the function to draw a circle at full alpha or the stated transparency | -Defined true and false Booleans  #define TRUE 1  #define FALSE 0  Inside the DrawCircle function I included a parameter called ‘isClear’, when passing in either TRUE or FALSE, will dictate whether the circle will be drawn with transparency.  float alphaValue = isClear ? color.alpha : 1;  glColor4f(color.r, color.g, color.b, alphaValue);  If the boolean (isClear) is TRUE (1), then I will take the alpha value stored inside the color struct being passed in. If it false I will render it as 1, being full transparency. | Before using the ternary operator, I tried to use if else conditions to render the circle with either  **glColor3f** and **glColor4f**. This gave me a lot of flickering artifacts. I changed the logic to always render the colour as **glColor4f**, and only change the alpha value. |
| 10/03/2023 | 2hr | Setting up particle system  Keyboard input to work  Trying to make globe transparent | typedef struct Position2xy {  float x;  float y;  }Position2xy;  typedef struct Particle\_t {  Position2xy pos;  float size;  float dy;  int timer;  int active;  }Particle\_t; | I decided to remove the snowglobe circle and work on the particle system from here on as hiding objects outside the circle proved to be quite hard without calculating the points around the circle. |
| 10/03/2023 | 2hr | Debugging particle system  Setting up particle initializers | void spawnParticles(void){  for (inti=0; i<MAX\_PARTICLES;  i++){  Particle\_t\* snow = &particleSystem[i];  spawnParticle(snow,snow->pos);}}  void spawnParticle(Particle\_t \*p, Position2xy pos){  glPointSize(p->size);  glColor4f(  p->color.r,  p->color.g,  p->color.b,  p->color.alpha);  glBegin(GL\_POINTS);  glVertex2f(pos.x, pos.y);  glEnd();  } | I wrote some functions to spawn particles and pre allocate values into the particle fields before spawning them.  Timing the snow to fade in and out naturally is still in need of debugging. |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |