|  |  |  |  |
| --- | --- | --- | --- |
| **WBS** | | | |
| **TASK** | **DESIGNATION** | **STUDENT NAME** | **STUDENT ID** |
| 1. Setting up the implementation environment | Reuse Model | Mustafa Gamal | 41520002 |
| 1. Implementation Of Shader Class | Agile Model | Mustafa Gamal | 41520002 |
| 1. Implementation Of Camera Class | Agile Model | Mustafa Gamal | 41520002 |
| 1. Implementation Of GLFW window Class | Reuse Model | Mustafa Gamal | 41520002 |
| 1. Implementation Of Primitives Header File | Agile Model | Mustafa Gamal | 41520002 |
| 1. Create a testing prototype | Test | Mustafa Gamal | 41520002 |
| 1. Study bloom effect and deferred shading |  | Mustafa Gamal | 41520002 |
| 1. Study particles system generator |  | Mustafa Gamal | 41520002 |
| 1. Study AntTweakBar |  | Mustafa Gamal | 41520002 |

|  |  |  |  |
| --- | --- | --- | --- |
| **TASK** | **DESIGNATION** | **STUDENT NAME** | **STUDENT ID** |
| 1- Setting up the implementation environment | Reuse Model | Mustafa Gamal | 41520002 |
| Start criteria | GLFW & Glad libraries should be downloaded ,compiled & ready for linking.  Up & running Compiler (Visual Studio 2017) | | |
| Inputs | -GLFW windowing Library  -Glad Library for opengl functions  -Visual studio 2017 compiler | | |
| Activities | -Download and compile GLFW & Glad  -Create Lib & Includes Folders  -Create VisualStudio empty project and link (GLFW - Glad)  -Include libraries in the project  -build and run the project ( linking errors check ) | | |
| Outputs(deliverables) | An environment ready for ( OpenGl Programming ) | | |
| Ending criteria | The project is running without linking or any other errors | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **TASK** | **DESIGNATION** | **STUDENT NAME** | **STUDENT ID** |
| 2- Implementation Of Shader Class | Agile Model | Mustafa Gamal | 41520002 |
| Start criteria | -Task no.1 “Setting up the implementation environment” should be ended | | |
| Inputs | -glm library  -glad library  -fstream & sstream libraries  -string & iostream libraries | | |
| Activities | -include the specified libraries  -create “shader” class  -create constructor takes shader files as parameters and read from it  -check for reading errors  -compile shaders  -check for compilation errors  -delete unnecessary shader objects  -create “use()” function for activating shader  -create utility uniform functions | | |
| Outputs(deliverables) | Compiled and activated shader of any type ( vertex, fragment or geometry shader) | | |
| Ending criteria | * Start of TASK no.6 “Create a testing prototype” | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **TASK** | **DESIGNATION** | **STUDENT NAME** | **STUDENT ID** |
| 3- Implementation Of Camera Class | Agile Model | Mustafa Gamal | 41520002 |
| Start criteria | -Task no.1 “Setting up the implementation environment” should be ended | | |
| Inputs | -glm library  -glad library  -vector library | | |
| Activities | -include the specified libraries  -create “Camera” class  -create enum defines several possible options for camera movment  -create default camera values variables  -create a constructor takes new positions and attributes as parameters  -create “GetViewMatrix()” function returns the view matrix calculated using eular angles and the LookAt matrix  -create “ProcessKeyBoard()” function that processes input received from any keyboard-like input system  -create “ProcessMouseMovement()” function that processes input received from a mouse input system  -create “ProcessMouseScroll()” function that processes input received from a mouse scroll-wheel event | | |
| Outputs(deliverables) | Full camera movement/scrolling processing ( using keyboard and mouse as control units ) | | |
| Ending criteria | * Start of TASK no.6 “Create a testing prototype” | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **TASK** | **DESIGNATION** | **STUDENT NAME** | **STUDENT ID** |
| 4-Implementation Of GLFW window Class | Reuse Model | Mustafa Gamal | 41520002 |
| Start criteria | -Task no.1 “Setting up the implementation environment” should be ended  - Task no.2 “Implementation of Shader Class” should be ended  - Task no.3 “Implementation of Camera Class” should be ended | | |
| Inputs | -glm library  -glad library  -vector library  -string library  -Camera Class | | |
| Activities | -include the specified libraries  -create “MyWindow” class  -create object from Camera Class  -create a constructor takes window name/dimensions as parameters and initializes the glfw window  -create “processinput()” function the processes any received keyboard input from the user  -create “framebuffer\_size\_callback()” function handles any change in window size (by OS or user resize)  -create “mouse\_callback()” function that handles mouse movments  -create “scroll\_callback()” function that handles scrolling wheel events | | |
| Outputs(deliverables) | GLFW display window that accepts user interactions (mouse – keyboard) | | |
| Ending criteria | * Start/End of TASK no.6 “Create a testing prototype” | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TASK** | **DESIGNATION** | | **STUDENT NAME** | | | **STUDENT ID** | |
| 5-Implementation Of Primitives Header File | Agile Model | | Mustafa Gamal | | | 41520002 | |
| Start criteria | -Task no.1 “Setting up the implementation environment” should be ended  - Task no.2 “Implementation of Shader Class” should be ended  - Task no.3 “Implementation of Camera Class” should be ended | | | | | | |
| Inputs | -class Shader  -class Camera  -screen width & height  -vertices coordinates | | | | | | |
| Activities | -implement class Cube  -implement class Quad | | | | -create constructor that generate buffer, vertex array and vertex attribute pointer data  -create another constructors with extra feature (texturing) takes texture path as a parameter  -create “perspective\_Trans()” that adjust the perspective view  -create “renderCube()”/”renderQuad()” function that binds the assigned texture & vertex array to render the vertices  -create “renderCube()”/”renderQuad()” this takes vector of color info to render the shape without texture  -create “loadTexture()” handles the loading texture process  -create “freeCube()/freeQuade()” to delete unnecessary objects from the memory | | |
| -implement class Point  -implement class Line | | | | -create constructor that generate buffer, vertex array and vertex attribute pointer data  -create “perspective\_Trans()” that adjust the perspective view  -create “renderPoint()”/”renderLine()” function that binds vertex array to render the vertices & takes vector of color info  -create “freePoint()/freeLine()” to delete unnecessary objects from the memory | | |
| Outputs(deliverables) | The ability to construct/render any of the 4 implemented primitive shapes (Cube - Quad - Point - Line) | | | | | | |
| Ending criteria | * Start/End of TASK no.6 “Create a testing prototype” | | | | | | |
| **TASK** | | **DESIGNATION** | | **STUDENT NAME** | | | **STUDENT ID** | |
| 6- Create a testing prototype | | Test | | Mustafa Gamal | | | 41520002 | |
| Start criteria | | * The pervious tasks No.(1,2,3,4 & 5) should be implemented/ready for integration and testing | | | | | | |
| Inputs | | -glm library  -glad library  -glfw library  -stb\_image library  -Camera class  -Shader class  -GLFWwidnow.h  -Primitives.h | | | | | | |
| Activities | | -include the specified libraries  -create “Main()” Function  -initialize new window using Mywindow class  -load all opengl function pointers using gladLoadGLLoader()  -initialize object from each of the 4 primitive shape classes  -implement the rendering loop  -call “processInput()” from the main window object  -set the glClear and glClearColor parameters  -call perspective\_Trans() & rendering functions from each shape object  -adjust the transformation matrix and pass it to perspective\_Trans()  - call swap buffer function glfwSwapBuffers()  - free all the unnecessary objects  -terminate glfw window using glfwTerminate() | | | | | | |
| Outputs(deliverables) | | GLFW display window contains one or more primitive shapes  In the scene with proper texture/colors  -no shader errors  -the window should respond seamlessly to the user interactions(keyboard-mouse) while exploring the scene | | | | | | |
| Ending criteria | | * Start/End of TASK no.6 “Create a testing prototype” | | | | | | |