UOW MALAYSIA KDU PENANG UNIVERSITY COLLEGE
PART OF THE UNIVERSITY OF WOLLONGONG AUSTRALIA GLOBAL NETWORK

ASSIGNMENT COVER PAGE

Validated by
UNIVERSITY OF LINCOLN UNITED KINGDOM

Programme	Course Code and Title
Bachelor of Computer Science (Hons)	CCG3013/N Computer Graphics
Student's name / student's id	**Lecturer's name**
CHAN SEOW FEN / 0207368	Mr. Tee Eng Hong

Date issued	Submission Deadline	Indicative Weighting
Week 2 - 23/09/2024	Week 6 - 27/10/2024	30%
Assignment [1] title	Simulation & Report Writing	

This assessment assesses the following course learning outcomes

# as in Course Guide	UOWM KDU Penang University College Learning Outcome
CLO2	Evaluate and draw the graphics elements involved in a storyboard using image manipulation software.
CLO3	Develop graphics program to render scenes, user interfaces, and interactions.

# as in Course Guide	University of Lincoln Learning Outcome
CLO1	Appreciation of a range of different graphics hardware devices and software used.
CLO4	Knowledge and skills in colour models, be familiar with 2D and 3D software programming.

Student's declaration

I certify that the work submitted for this assignment is my own and research sources are fully acknowledged.

Student's signature:

Submission Date: 27/10/2024

Table of Contents

1.0 Conceptual Design 2
1.1 Initial Design Diagram 2
1.2 Design Concept 2
1.2.1 Colours and Themes 2
1.2.2 User Interface Design 2
2.0 Picture Rendering 4
2.1 Rendered Result 4
2.2 Icon Rendering Preprocessing 4
2.3 Code Description 5
2.2.1 colour.h 5
2.2.2 dimension.h 5
2.2.3 object.h 6
2.2.4 object.cpp 6
2.2.5 render.h 7
2.2.6 init.h 7
2.2.7 main.cpp 7
3.0 Bibliography 9

1.0 Conceptual Design

1.1 Initial Design Diagram

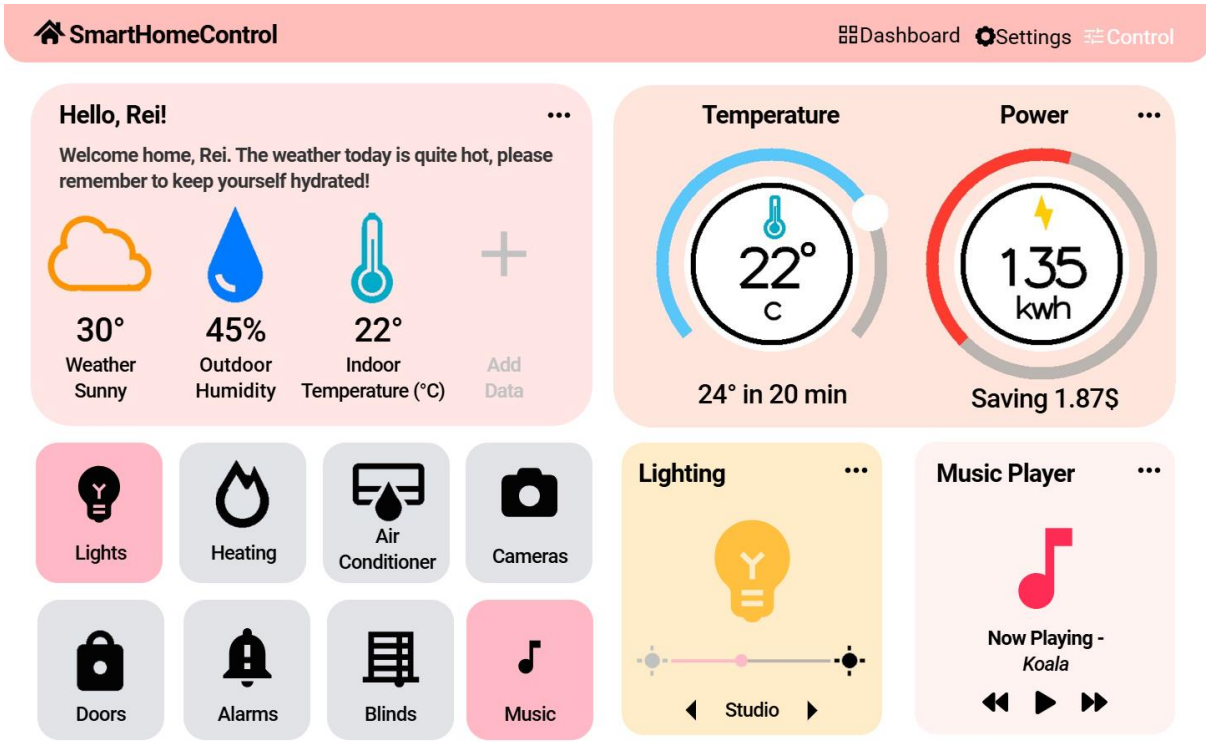

Figure 1 Initial Design

1.2 Design Concept

1.2.1 Colours and Themes

The colour theme selected is mainly pink, white and black. As to present a user interface that is aesthetically and using harmony colour, pastel colour which is soft colour is selected for the entire screen. Moreover, to avoid presenting a boring user interface, more colour like orange, blue, yellow and red is used to lighten up the visual experience. In addition, to reduce user's eye fatigue, cold colour is applied to less priority item like the feature button that is not in used, and warm colour is applied to highlighted feature button to attract user's eye focus. Black colour as the text colour is as a colour contrast to help user with eye disabilities to read easily without obstacles. Overall, the user interface is taking a minimalistic, clean, soft and aesthetic route to enhance user experience.

1.2.2 User Interface Design

In terms of the user interface design, it mainly consists of 5 container: Navigation Bar, 4 Dashboard, with Dashboard 1 to present information such as weather, temperature, humidity and other optional atmospheric data such as wind, precipitation and more for user to add to show on the dashboard. Following that, Dashboard 2 serve as a dashboard for user to have control over their smart home by showing the home information such as indoor temperature, power usage and sub-information such as the temperature changes and electricity cost saved by the smart home control. Subsequently, Dashboard 3 serve as the feature buttons for user to switch as to have control on that particular feature they clicked on. Maximumly only 2 features can be controlled at one time. As shown in the *Figure 1,* Lights button and Music button have been pressed and the button turned into pink colour to indicate these two features is in used in control panel, while other unused features remain as grey. Finally, Dashboard 4 consists of two sub-container as one is Control Panel 1 and another is Control Panel 2. It is used to show the control features user selected at Dashboard 3, it consists of the feature name

and the control interface corresponding to their feature. For example, as shown in *Figure 1* brightness bar and room switching for Lightning Control and previous, next and pause/play for Music Player. In addition, there are button presented as 3 dots to indicate “More” option for user to customise the dashboards. Besides, the Navigation Bar consists of the application name: SmartHomeControl and its application logo at its left side, it also comes with three buttons for user to navigate the page with, which is dashboard for user to customise the dashboard layout, settings for application settings such as theme colour and font settings, and lastly control for user to control their smart home appliances as shown in *Figure 1*.

2.0 Picture Rendering

2.1 Rendered Result

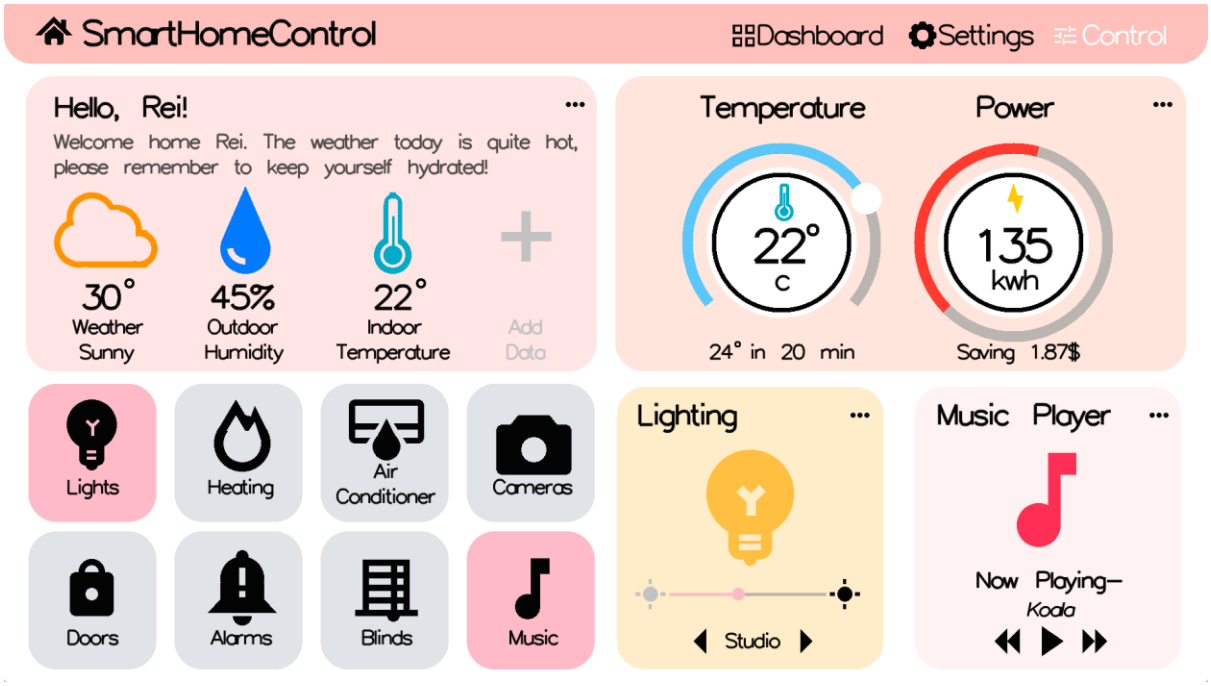

Figure 2 Rendered Result

2.2 Icon Rendering Preprocessing

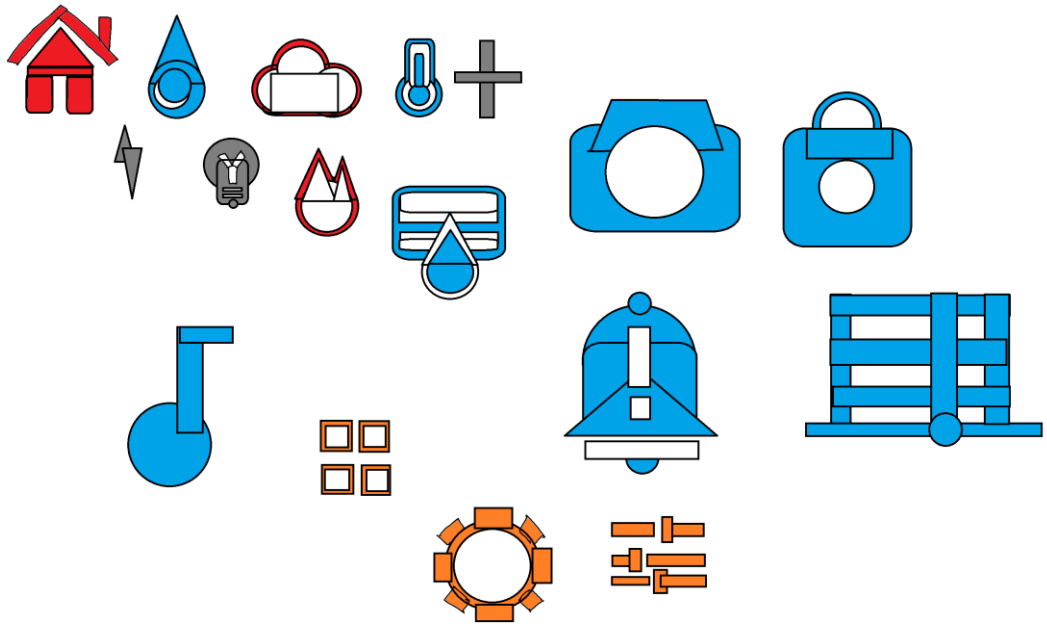

Figure 3 Icon Draft

As to render the icon above using primitive shape including circle, torus, triangle, quadrilateral, rectangle and rounded rectangle, the icon is drafted as shown in *Figure 3*. The image above is drafted using paint with circle, triangle, rectangle and rounded rectangle to ensure that the icon is recreateable using primitive shape and also provide a better sense of rendering it using code.

2.3 Code Description

As shown in *Figure 2*, the rendering result is basically same as the initial design, except for the font. This is due to the default font provided is only Times Roman and Helvetica, hence the font is changed from Roboto to Helvetica while the others remained the same.

Moreover, the code is coded in vscode, hence, due to the difference in configuration, it would not run in Devc++. It is highly recommended to download all the files including .vscode configuration file and run it in vscode that is preconfigured using the same ways like in Devc++ (glut.h etc…). In addition, due to the fact that, unlike Devc++, vscode does not come with the feature to automatically run all files, hence, there will be multiple include files statement that are unnecessary when running in Devc++, but required to make the program work in vscode.

In addition, the code is organised in the way of grouping the codes into 7 files, according to their function to make the code more organised, manageable and readable. The files included “colour.h”, “dimension.h”, “object.h”, “object.cpp”, “render.h”, “init.cpp” and “main.cpp”. The detailed description of each file is included in the section below. Overall, the key strengths of the code structures included:

1. Object-Oriented Design: Uses classes to encapsulate drawing and transformation functionality
2. Modular Architecture: Separates concerns into different files (colours, dimensions, objects, rendering)
3. Comprehensive UI Components: Supports various smart home controls
4. Professional Graphics: Implements anti-aliasing and transparency
5. Responsive Design: Uses screen dimensions for layout calculations
6. Rich Visual Elements: Supports various shapes, text, and icons
7. Technical Implementation:
    - Uses OpenGL for rendering
    - GLUT for window management
    - Implements custom shape drawing algorithms
    - Supports transformations and complex compositions
    - Handles text rendering with scaling and styling options
    - Manages OpenGL state for optimal rendering performance

2.2.1 colour.h

A header file containing colour definitions used throughout the UI rendering. Contains:

- Default colour constants (white, black, various greys)
- Theme colour constants (pink, orange, yellow, blue variants)
- All colours are defined in RGBA format (red, green, blue, alpha)
- Colours are used for UI elements like buttons, text, and backgrounds

2.2.2 dimension.h

A header file defining window dimensions and positioning constants:

- Gets screen dimensions using GLUT
- Defines window width and height

- Calculates centre coordinates for window positioning
- Used for responsive layout calculations

2.2.3 object.h

Header file defining the Object class interface:

- Class declaration with public methods
- Mathematical constant definitions
- 2D primitive drawing method declarations
- Shape drawing functions:
    - Basic shapes (point, line, triangle, quad)
    - Complex shapes (circle, torus)
    - Special shapes (rounded rectangles)
- Transformation methods:
    - Translation
    - Rotation
    - Scaling
    - Mirroring
- Utility functions for text rendering
- Position tracking variables

2.2.4 object.cpp

Implementation file for the Object class that provides:

- Basic 2D primitive drawing functions
- Constructor and destructor implementations
- Methods for drawing:
    - Points
    - Lines
    - Triangles
    - Quadrilaterals
    - Rectangles
    - Rounded Rectangles
    - Circles
    - Torus
- OpenGL state management
- Matrix transformation handling

2.2.5 render.h

Rendering implementation file that:

- Defines UI component objects
- Contains rendering functions for:
    - Navigation bar
    - Application icon
    - Application name
    - Dashboard icon and text
    - Settings controls
    - Various UI components
- Implements colour and positioning for each component
- Handles complex shape composition for icons
- Manages UI layout and component relationships

2.2.6 init.h

Initialisation and rendering setup file containing:

- Main render function that calls individual component renders
- Renders UI elements like:
    - Light controls
    - Heat controls
    - Air conditioner controls
    - Camera controls
    - Door controls
    - Alarm controls
    - Blinds controls
    - Music controls
- Window initialisation function that:
    - Sets up GLUT display mode
    - Configures window properties
    - Initialises OpenGL rendering context
    - Sets up transparency and anti-aliasing
    - Configures matrix modes and buffer clearing

2.2.7 main.cpp

The entry point of the application that:

- Includes necessary headers
- Initialises the window system
- Sets up the render callback
- Starts the GLUT main loop
- Handles program termination

3.0 Bibliography

Benstead, L. (2014). *NeHe Productions: FreeType Fonts in OpenGL*. Gamedev.net. https://nehe.gamedev.net/tutorial/freetype_fonts_in_opengl/24001/

in. (2013, December). *Drawing torus in opengl*. Stack Overflow. https://stackoverflow.com/questions/20312697/drawing-torus-in-opengl

Khronos. (2010, January 10). *How to draw circle?* Khronos Forums. https://community.khronos.org/t/how-to-draw-circle/59661/11

large, D. (2015, April 25). *Drawing large text with GLUT?* Stack Overflow. https://stackoverflow.com/questions/29872095/drawing-large-text-with-glut

LearnOpenGL - Text Rendering. (2020). Learnopengl.com. https://learnopengl.com/In-Practice/Text-Rendering

OpenGL, in. (2009, May 7). *Smooth lines in OpenGL*. Stack Overflow. https://stackoverflow.com/questions/837060/smooth-lines-in-opengl

OpenGL, in. (2013, January 30). *Applying font styles in OpenGL and FreeType library*. Stack Overflow. https://stackoverflow.com/questions/14607306/applying-font-styles-in-opengl-and-freetype-library

OpenGL, in. (2022, December 30). *How to draw a rectangle in OpenGL with rounded corners?* Stack Overflow. https://stackoverflow.com/questions/74960029/how-to-draw-a-rectangle-in-opengl-with-rounded-corners

CCG3013/N Computer Graphics
ASSIGNMENT [1] Scene rendering (Weighted marks: 30%)

Part 1: Conceptual design (Score: 30%)

LEARNING OUTCOME	MARKING CRITERIA	SCALE					
		Fail (0-49)	**3rd Class (50-59)**	**2nd Lower Class (60-69)**	**2nd Upper Class (70-79)**	**1st Class (80-100)**	**YOUR MARKS/COMMENTS**
CLO2: Evaluate and draw the graphics elements involved in a storyboard using image manipulation software.	**1. Creativity in the use of geometrical objects (20%)**	No primitive shape has been drawn.	Limited primitive shapes have been drawn.	Sufficient primitive shapes have drawn. However, lack of composition of primitive shapes have been drawn.	There is a composition of primitive shapes have been drawn	There is a comprehensive of primitive shapes have been drawn.	
	2. Creativity in the use of colours (10%)	No colour has used.	Colours have used, however too vivid or blurry.	Appropriate colours are applied on the graphic objects. However, no cohesion of colours is used.	Appropriate colours are applied on the graphic objects. Certain cohesion of colours is used, which is either hot or cold colours,	Harmony colours were applied, which separate hot and cold colours	
						Total (30%)	

Part 2: Picture Rendering (Score: 70%)							
LEARNING OUTCOME	**MARKING CRITERIA**	**SCALE**					
		Fail (0-49)	**3rd Class (50-59)**	**2nd Lower Class (60-69)**	**2nd Upper Class (70-79)**	**1st Class (80-100)**	**YOUR MARKS/COMMENTS**
CLO3: Develop graphics program to render scenes, user interfaces, and interactions	**1. Programming code (10%)**	Unable to compile the code.	Relatively hard to interpret the code.	Code able to interpret, but either lack of comments, some lines of unused code, or inappropriate indentations.	Code able to interpret with appropriate indentations, however certain lines of code not encapsulated.	Program code is well managed which includes proper indentations. All lines of code are encapsulated	
	2. Window Setting (20%)	OpenGL window could not be rendered.	Inappropriate configuration of OpenGL window is set.	An appropriate configuration of OpenGL window is set. However, inappropriate handling of frame buffer.	An appropriate configuration of OpenGL window is set with proper handling of frame buffer.	An appropriate configuration of OpenGL window is set with proper handling of frame buffer. Good contrast settings between object and background.	
	3. Drawing functions (20%)	No drawing function is used.	Only limited drawing functions are used.	Good varieties of drawing functions are used. Unable to combine primitive objects to produce a shape.	Good varieties of drawing functions are used. Able to combine primitive objects to produce a shape.	Good varieties of drawing functions are used. Able to combine primitive objects to produce an aesthetics shape	
	4. Matrix transformation (20%)	No matrix transformation is applied.	Inappropriate application of matrices for transformation.	Appropriate application of matrices for transformation.	There is a mastery in the application of matrices for transformation.	There is a mastery in the application of matrices for transformation as well as matrix stacks	
	Total (70%)						
	Overall (100%)						
	Weighted Marks (30%)						