CSD1401 - Assignment 2

Purpose

This assignment will help students develop:

- Familiarity with the C programming language
- Familiarity with the Visual Studios IDE
- Familiarity with the CProcessing engine
- Familiarity with developing an application
- Familiarity with the concept of states

Overview

In this assignment, we will implement a program with a simple main menu will transition to an interactive application.

Task

Create the Visual Studio Project

1. In the `src` folder, you should see `mainmenu.h` and `mainmenu.c`, `carlevel.h` , `carlevel.c`, `utils.c`, `utils.h`, `game.c` and `main.c`
2. Use these files to create a CProcessing Visual Studio Project.

View and understand what's required from the sample program

Open the `sample` folder and run `main.exe`. You should see a window pop out on your screen. Observe what is happening in the window. In this assignment, you will be implementing a similar program to the sample program given to you.

You can think of the program as having the following separate states that you will need to implement:

1. A main menu state.
2. A car level state.

Specifications of each state is detailed in their respective sections below. Your final program does not need to be exactly the same as the example program given, but it must at least have those features.

Implementing utility functions

Open `utils.c` and `utils.c`. To help with the implementation of the features in each state, we can write simple reusable functions that can be used throughout the program. To get you started, you are provided with the function prototypes of is `IsAreaClicked`, `IsCircleClicked` and `AngleToVector` functions, which you can implement their definitions in `utils.c`.

- `IsAreaClicked` will check if a point is within a rectangular area.

    - `area_center_x` and `area_center_y` represents the X and Y point of the center of the rectangle respectively.
    - `area_width` and `area_height` represents the width and height of the rectangle respectively.
    - `click_x` and `click_y` represents the X and Y position of the point to test.
    - Returns 1 if the point given by `click_x` and `click_y` is within the rectangle given by `area_center_x`, `area_center_y`, `area_width` and `area_height`. Otherwise it will return 0.
- `IsCircleClicked` will check if a point is within a circle.
    - `circle_center_x` and `circle_center_y` represents the X and Y point of the center of the circle respectively.
    - `diameter` represents the diameter of the circle.
    - `click_x` and `click_y` represents the X and Y position of the point to test.
    - Returns 1 if the point given by `click_x` and `click_y` is within the circle given by `circle_center_x`, `circle_center_y` and `diameter`. Otherwise it will return 0.
- `AngleToVector` will convert an angle with respect to x-axis to a 2D vector that is rotated counter-clockwise from x-axis.
    - `radian_angle` represents the angle with respect to x-axis in radian.
    - It will return a `CP_Vector` representing the 2D vector that is rotated `radian_angle` counter-clockwise from x-axis.

Implement the Main Menu State

The main menu state will be the first state that the users see when your program launches. It will have two rectangular buttons that users can click on. On click, one button will exit the program, while another button will send the user to the Car Level State.

This main menu state must be implemented in the functions given in `mainmenu.h` and `mainmenu.c`

Implement the Car Level State

This section contains the specifications of the Car Level State.

The car level state will have three triangles that represents 3 cars. The user can only control one car at a time. They can change which car they want to control by clicking on the car. The controlled car's orientation will turn counter-clockwise with the A key and clockwise with the D key. The controlled car will move forward and backwards in the orientation they are facing with the W and S key respectively.

This Car Level state must be implemented in the functions given in `carlevel.h` and `carlevel.c`

Looking for the CProcessing functions to use

The CProcessing wiki here contains documentation on what functions it has available for you to use. Below are some of the functions you might want to look up and use in order to complete this assignment:

- CP_Graphics_DrawCircle
- CP_Graphics_DrawTRiangleAdvanced
- CP_Engine_SetNextGameState
- CP_Settings_RectMode

- CP_Settings_TextAlignment

Check your work

Check that your program's Main Menu State and Car Level State are both working according to their specifications given.

Again, your final program does not need to be exactly the same as the example program given, but it must at least have those features.

Grades

For this assignment, grades will be roughly be given as follows:

What was achieved	Grade
C Satisfied with menu and state transitions	A
Cars can be displayed, selected and moved	C
CSG not even functional	F

Grades in between can also be given (eg. D and B) depending on quality of implementation.

A+ will be considered for students who achieved A **and** have a clean submission (check **Deliverables** section below).

Deliverables

1. Zip your completed `mainmenu.h`, `mainmenu.c`, `carlevel.h` , `carlevel.c`, `utils.c` and `utils.h` using the convention `<DigiPen student id>_2.zip`. For example: If my DigiPen student e-mail is `cheng.dingxiang@digipen.edu`, my DigiPen student id is `cheng.dingxiang`. My zipped folder will hence be named `cheng.dingxiang_2.zip`
2. Submit `<DigiPen student id>_2.zip` file to the appropriate assignment submission link in Moodle.
3. After submitting, do a sanity check by re-downloading the file that you submitted and ensure that it is indeed the file that you submitted.