Assessment1: 2D Modelling Project

Author Yifei Luo

ID number 1928155

Module Code CPT205

Module Title Computer Graphics

Programme Information and Computing Science

Design

1. Main scenario
    1) The background (void background())
        - The background will be darker when it is night, and the background will be lighter when it is morning.
    2) FB (void fb())
        - FB has a concave wall in the middle, some windows, as well as the lines between these windows. Each two windows are adjacent to each other.
    3) CB (void cb())
        - CB has a brace, cube, as well as the concave lines
    4) Trees (void tree1(); void tree2(); void tree3(); void tree4())
        - There are 4 trees in front of the building
    5) Bird (void bird())
        - The bird consists of head, eyes, body, wings, and a tail
        - The bird is flying automatically
    6) Clock (void drawClock())
        - The clock consists of the outline, a minute hand, and an hour hand.
    7) Balloon (void balloon1(); void balloon2(); void balloon3())
        - The balloon consists of lines and sphere
        - There are three balloons swinging automatically
    8) "Happy anniversary!" text (void printBitmapString(void* font, const char* s); int TextOut(float x, float y, const char* cstr); void greetingText())
        - This line follows the current time behind the “Happy anniversary” line.
2. Interactive operations
    1) Menu
        - Show all the operations that is set in the keyboard
        - This menu has sub-menus to categorize different operations
    2) Switch building between FB and CB
        - Users can choose which building to show in the screen
    3) Heighten the bird
        - Users can heighten the bird while it is flying
    4) Enlarge or shrink the balloon
    5) Set, quicken or slow down the clock
        - Users can set customized clock, quicken or slow down the clock rotation speed
    6) Change the viewport (zoom in/out, move)
        - Users can zoom in to see closely

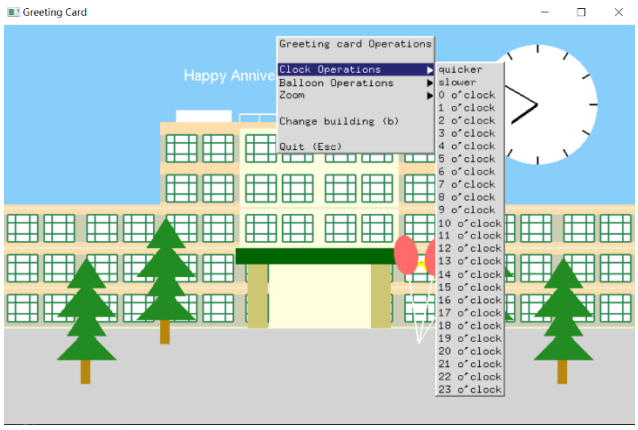
Implementation

1. Menu operation
   - I create a menu, so that mouse and keyboard can control the display altogether.
   - When we click the right button, we can show the menu and click what we want to operate.
     `glutCreateMenu` is to create the menu structure.
     `glutAddMenuEntry` is to add the option.

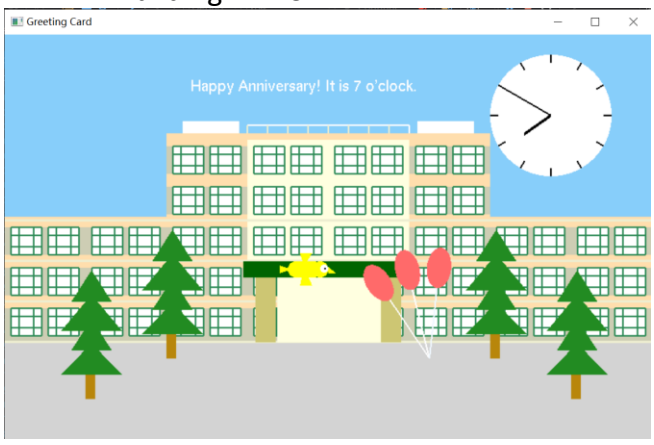
     For example,

```
balloonOp = glutCreateMenu(menu);
glutAddMenuEntry("larger", 'm');
```

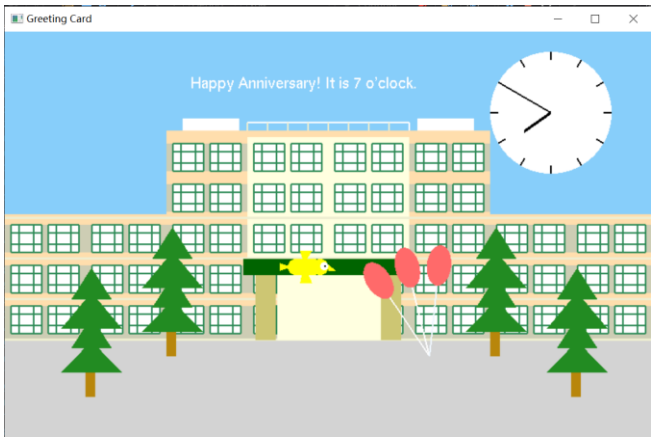
     we create a new menu balloonOp, and a new button in balloonOp menu. When we click this "larger" button, actually we click on 'm' button, which is already set how it will perform in the `keyboardInput()`.

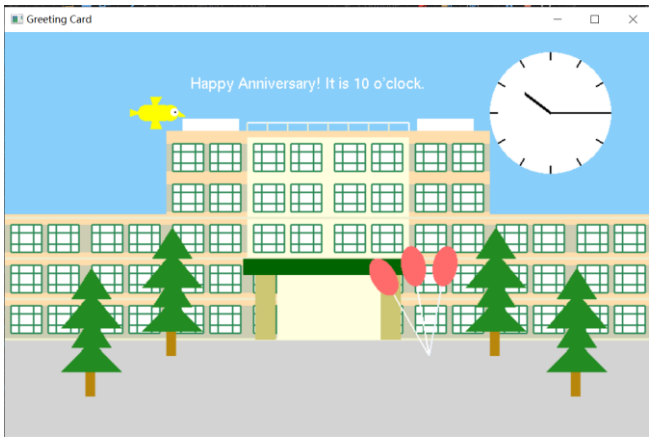


2. Switch building between FB and CB
   - When we press 'b' button/'Change building' in the menu, the building will be changed.
   - This is easy to perform by just changing the variable 'building' to switch.
     Building==0: FB
     Building==1: CB

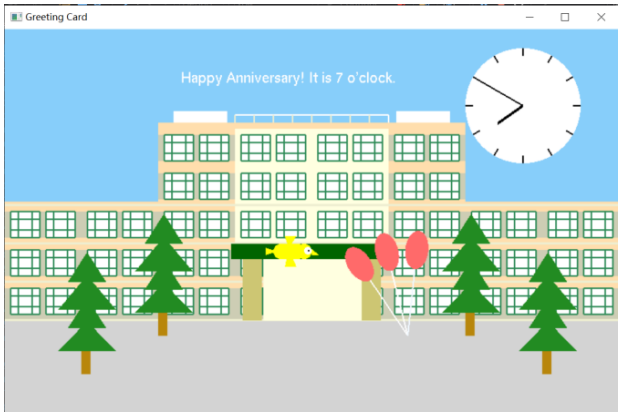


3. Heighten the bird
   - When we click the left mouse, the bird will fly higher.
   - Variable 'birdy', which indicates the y position of the bird, is larger once we press the left mouse.





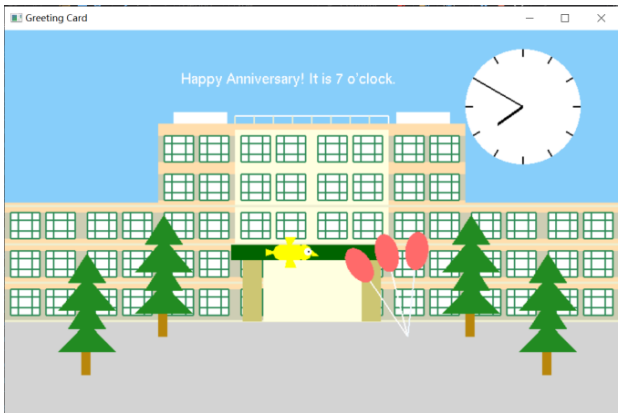
4. Enlarge or shrink the balloon
   - When we click the ‘n’, the balloon will be smaller
     When we click the ‘m’, the balloon will be larger
   - Variable ‘bp’, which indicates the scale of balloon, is larger once we press ‘m’, or smaller once we press ’n’.



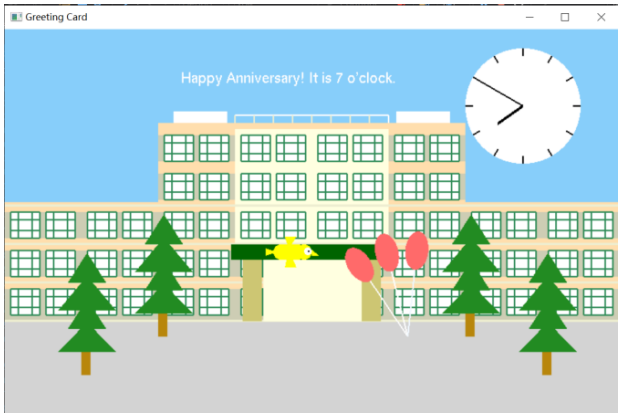
5. Set, quicken or slow down the clock
   - When we choose the ‘clock operations’ in the menu, the clock will be changed correspondingly.
   - Variable ‘clock’ can set the current o’clock.
   - We can use ‘k’ ‘l’ button or ‘quicker’ ‘slower’ in the menu to change the clock rotation speed

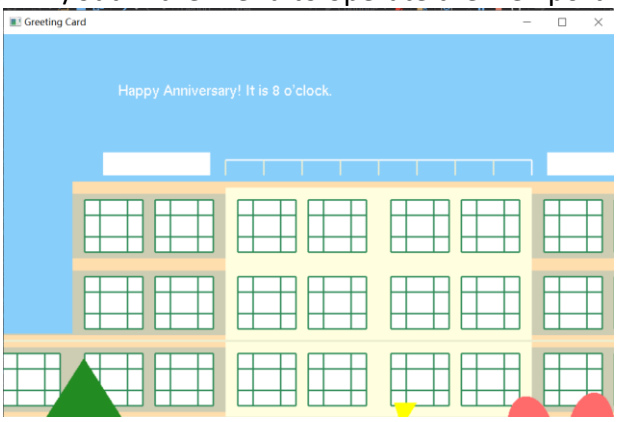
6-18 is daytime, while 18-24-6 is nighttime.

   - Variable ‘cp’, which indicates the rotation angle of clock, is quicker once we press ‘k’, or slower once we press ’l’.



6. Change the viewport (zoom in/out, move)
   We can use ‘w’ ’a’ ’s’ ’d’ ‘-’ ’+’ button or zoom in/out in the menu to operate the viewport





7. Simplified methods (`glArc, glCircle, glLines, glRect, glTri`)
   I create a series of method, including `glArc, glCircle, glLines, glRect, glTri`, in substitute of long code of creating these components every time.