```
//Claire Forrestal and Tammy Nguyen-Huynh
//Software Design Project
//Dr. Schlosser 10:20AM

#include <FEHLCD.h>
#include <FEHIO.h>
#include <FEHUtility.h>
#include <stdlib.h>
#include <FEHBuzzer.h>

// Class Statistics includes all data for keeping track of game statistics
class Statistics
{
    private:
    int all_scores[14];
    int score;
    int game;

    public:
    Statistics();
    void Stats();
    void Game_Counter();
    void Score_Setter();
    void Reset_Score();
    void Display_Current_Score();
    void Display_All_Scores();
};

int main(void)
{
    // Declaring variables used throughout program
    float x, y, x_touch, y_touch, sequence1[10], sequence2[10];
    int i, scarlet = 1, blue = 2, green = 3, yellow =4,  choice, choice1;

    // Declaring class Statistics
    Statistics score_recorder;

    //Create Intro screen
    LCD.Clear(FEHLCD::White);
    LCD.SetFontColor(BLACK);

    //Welcome tone
    Buzzer.Tone( FEHBuzzer::Bf5, 250 );
    Buzzer.Tone( FEHBuzzer::A5,  100 );
    Buzzer.Tone( FEHBuzzer::Bf5, 250 );
    Buzzer.Tone( FEHBuzzer::A5,  100 );
    Buzzer.Tone( FEHBuzzer::Bf5, 250 );

    LCD.WriteAt("Welcome to", 111,20);
    LCD.SetFontColor(SCARLET);
    LCD.WriteAt("S",10,40);
    Sleep(250);

    LCD.SetFontColor(ORANGE);
    LCD.WriteAt("I",75,67.5);
    Sleep(250);
```

```
    LCD.SetFontColor(YELLOW);
    LCD.WriteAt("M",160,95);
    Sleep(250);

    LCD.SetFontColor(GREEN);
    LCD.WriteAt("O",235,122.5);
    Sleep(250);

    LCD.SetFontColor(BLUE);
    LCD.WriteAt("N",300,150);
    Sleep(1000);

    //Create while loop to return player to game menu once play, rules,
statistics or credits option has been chosen.
    while(true)
    {
        // Clear background
        LCD.Clear(FEHLCD::White);

        // Introduction to the Game: Start Menu
        // Declaring menu array
        FEHIcon::Icon start_menu[4];

        // Menu labels
        char start_menu_labels[4][20] = {"Play", "Rules", "Statistics",
"Credits"};

        // Displaying menu on screen
        FEHIcon::DrawIconArray(start_menu, 4, 1, 10, 10, 2, 2,
start_menu_labels, SCARLET, BLACK);

        //Allow user to choose play, rules, statistics or credits by
recording their touch and executing their decision upon release.
        while( !LCD.Touch(&x, &y) )
        {}
        while (LCD.Touch(&x_touch, &y_touch))
        {}

        // Seed with TimeNow() to generate random numbers later on.
        srand(TimeNow());

        // Determines what was selected based on user's input
        if (y > 10 && y < 65)       //Play
        {
            LCD.Clear( FEHLCD::White ); // Clear background

            //Switch-case to allow user to choose Easy or Hard level.
            // Declaring level array
            FEHIcon::Icon level[2];

            //Level menu labels
            char level_labels[2][20] = {"Easy", "Hard"};

            //Displaying menu on screen
            FEHIcon::DrawIconArray(level, 2, 1, 10, 10, 2, 2,level_labels,
GREEN, SCARLET);
```

```
            while( !LCD.Touch(&x, &y) )
            {}
            while (LCD.Touch(&x_touch, &y_touch))
            {}

            //Create if-else if statement to set expression for a switch-
case.
            if (y > 0 && y < 110)   // Easy
            {
                choice1 = 1;
            }
            else if (y > 110 && y < 220)    // Hard
            {
                choice1 = 2;
            }

            //Insert switch case to execute choice
            switch (choice1)
            {
                case (1):   //Easy mode

            {   //Start bracket for switch case 1

            //Set score to 0
            score_recorder.Reset_Score();

            //Create four rectangles
            LCD.Clear( FEHLCD::White ); // Clear background
            LCD.SetFontColor( SCARLET );
            LCD.DrawRectangle(0, 0, 160, 120);
            LCD.SetFontColor( BLUE );
            LCD.DrawRectangle(160, 0, 160, 120);
            LCD.SetFontColor( GREEN );
            LCD.DrawRectangle(0, 120, 160, 120);
            LCD.SetFontColor( YELLOW );
            LCD.DrawRectangle(160, 120, 160, 120);

            // Allows player to view grid before sequence is shown
            Sleep(2000);

            //Declare integer used in do-while loop
            int k = 1;

            //Do-while loop to randomly generate coordinate pair and fill
rectangle corresponding to it (drives the game)
            do
            {

                // Resets k = 1 for a new game (if player decides to play
again)
                k = 1;

                for (i = 0; i < 3; i++) // Loops for length of sequence
                {
                    //Create four rectangles
                    LCD.Clear( FEHLCD::White ); // Clear background
                    // Top-Left of screen
```

```
                    LCD.SetFontColor( SCARLET );
                    LCD.DrawRectangle(0, 0, 160, 120);
                    // Top-Right
                    LCD.SetFontColor( BLUE );
                    LCD.DrawRectangle(160, 0, 160, 120);
                    // Bottom-Left
                    LCD.SetFontColor( GREEN );
                    LCD.DrawRectangle(0, 120, 160, 120);
                    // Bottom-Right
                    LCD.SetFontColor( YELLOW );
                    LCD.DrawRectangle(160, 120, 160, 120);

                    //Randomly generate x and y coordinate pair
                    x = rand() % 321;
                    y = rand() % 241;

                    //Check which rectangle randomly generated pair is in and
convert to a rectangle number
                    if (x < 160 && y < 120) //Check to see if rand generate
pair is in scarlet(value of 1) rectangle
                    {
                        sequence1[i] = scarlet;
                        //Fill rectangle that corresponds to randomly
generated coordinate pair
                        LCD.SetFontColor( SCARLET );
                        LCD.FillRectangle(0, 0, 160, 120);
                        Sleep(1000);    // Allows player to see color
                        // Clear rectangle (flashes the color of rectangle)
                        LCD.SetFontColor( WHITE );
                        LCD.FillRectangle(0, 0, 160, 120);
                        Sleep(500);     // Pause between flashing rectangle
colors in sequence
                    }
                    else if (x > 160 && y < 120) // blue
                    {
                        sequence1[i] = blue;
                        LCD.SetFontColor( BLUE );
                        LCD.FillRectangle(160, 0, 160, 120);
                        Sleep(1000);
                        LCD.SetFontColor( WHITE );
                        LCD.FillRectangle(160, 0, 160, 120);
                        Sleep(500);
                    }
                    else if (x < 160 && y > 120) // green
                    {
                        sequence1[i] = green;
                        LCD.SetFontColor( GREEN );
                        LCD.FillRectangle(0, 120, 160, 120);
                        Sleep(1000);
                        LCD.SetFontColor( WHITE );
                        LCD.FillRectangle(0, 120, 160, 120);
                        Sleep(500);

                    }
                    else if (x > 160 && y > 120) // yellow
                    {
                        sequence1[i] = yellow;
```

```
                        LCD.SetFontColor( YELLOW );
                        LCD.FillRectangle(160, 120, 160, 120);
                        Sleep(1000);
                        LCD.SetFontColor( WHITE );
                        LCD.FillRectangle(160, 120, 160, 120);
                        Sleep(500);
                    }
                }

                // Player input
                for (i = 0; i < 3; i ++)
                {
                    // Wait for player input
                    while(!LCD.Touch(&x, &y) )
                    {}
                    // Allows code to continue execution upon release
                    while (LCD.Touch(&x_touch, &y_touch))
                    {}

                    if (x < 160 && y < 120) //Check to see if touch is in
scarlet rectangle
                    {
                        sequence2[i] = scarlet;
                        //Fill rectangle that corresponds to randomly
generated coordinate pair
                        LCD.SetFontColor( SCARLET );
                        LCD.FillRectangle(0, 0, 160, 120);
                        Sleep(250);
                        // Clear rectangle so that the color appears to have
flashed for a moment
                        LCD.SetFontColor( WHITE );
                        LCD.FillRectangle(0, 0, 160, 120);
                    }
                    else if (x > 160 && y < 120) // blue
                    {
                        sequence2[i] = blue;
                        LCD.SetFontColor( BLUE );
                        LCD.FillRectangle(160, 0, 160, 120);
                        Sleep(250);
                        LCD.SetFontColor( WHITE );
                        LCD.FillRectangle(160, 0, 160, 120);
                    }
                    else if (x< 160 && y > 120) // green
                    {
                        sequence2[i] = green;
                        LCD.SetFontColor( GREEN );
                        LCD.FillRectangle(0, 120, 160, 120);
                        Sleep(250);
                        LCD.SetFontColor( WHITE );
                        LCD.FillRectangle(0, 120, 160, 120);
                    }
                    else if (x > 160 && y > 120) // yellow
                    {
                        sequence2[i] = yellow;
                        LCD.SetFontColor( YELLOW );
                        LCD.FillRectangle(160, 120, 160, 120);
                        Sleep(250);
```

```
                        LCD.SetFontColor( WHITE );
                        LCD.FillRectangle(160, 120, 160, 120);
                    }

                    //Compare each input of user to randomly generated values
                    if (sequence1[i]!= sequence2[i])
                    {
                        LCD.Clear( FEHLCD::White );
                        LCD.SetFontColor( SCARLET );
                        Buzzer.Buzz(500);
                        LCD.WriteLine("Sorry! You lost.");  // on-screen
message
                        LCD.Write("Score: ");


                        // Display score on screen
                        score_recorder.Display_Current_Score();
                        // Count number of games played for statistics
                        score_recorder.Game_Counter();
                        //Record stats from game
                        score_recorder.Stats();

                        LCD.Write("Tap screen");

                        while( !LCD.Touch(&x, &y) )
                        {}
                        while (LCD.Touch(&x_touch, &y_touch))
                        {}

                        Sleep(1000);    // Delay between player input and
displaying replay menu

                        //Clear background
                        LCD.Clear( FEHLCD::White );

                        //Switch-case to allow user to play again or quit.
                        // Declaring replay array
                        FEHIcon::Icon replay[2];
                        // Replay menu labels
                        char replay_labels[2][20] = {"Play Again", "Return to
Menu"};
                        // Displaying menu on screen
                        FEHIcon::DrawIconArray(replay, 2, 1, 10, 10, 2,
2,replay_labels, GREEN, SCARLET);

                        while( !LCD.Touch(&x, &y) )
                        {}
                        while (LCD.Touch(&x_touch, &y_touch))
                        {}

                        //Create if-else if statement to set expression for a
switch-case.
                        if (y > 0 && y < 110)   // Play Again
                        {
                            choice = 1;
                            score_recorder.Reset_Score();   // Reset score
for next game
```

```
                        }
                        else if (y > 110 && y < 220)    // Quit
                        {
                            choice = 2;
                        }

                        //Insert switch-case to execute choice
                        switch (choice)
                        {
                            case (1):
                            //Continue do-while loop to play again.
                                k = 2;
                                break;
                            case (2):
                            //Break out of do-while loop to return to main
menu.
                                k = 0;
                                break;
                        }
                        break;  // Break out of if statement
                    }

                }
                    if (k == 1)
                    {
                        LCD.Clear( FEHLCD::White );
                        LCD.SetFontColor( SCARLET );
                        Buzzer.Tone( FEHBuzzer::Bf5, 150 );
                        Buzzer.Tone( FEHBuzzer::A5,  75 );
                        Buzzer.Tone( FEHBuzzer::Bf5, 150 );
                        Buzzer.Tone( FEHBuzzer::A5,  75 );
                        Buzzer.Tone( FEHBuzzer::Bf5, 150 );
                        LCD.WriteLine("Nice job! :D");  // on-screen message
                        LCD.WriteLine("Tap screen to continue");
                        while(!LCD.Touch(&x, &y))
                        {}
                        k = 1;
                        score_recorder.Score_Setter();
                        Sleep(1000);
                    }
              } while(k >= 1);

            //Break out of case 1 (Easy mode)
            break;
            } //End bracket for  case 1

            case (2):   //Hard mode (same code as easy mode but for loop runs
through more times)
                //Set score to 0
                score_recorder.Reset_Score();

                //Create four rectangles
                LCD.Clear( FEHLCD::White ); // Clear background
                LCD.SetFontColor( SCARLET );
                LCD.DrawRectangle(0, 0, 160, 120);
                LCD.SetFontColor( BLUE );
                LCD.DrawRectangle(160, 0, 160, 120);
```

```
                LCD.SetFontColor( GREEN );
                LCD.DrawRectangle(0, 120, 160, 120);
                LCD.SetFontColor( YELLOW );
                LCD.DrawRectangle(160, 120, 160, 120);

                // Allows player to view grid before sequence is shown
                Sleep(2000);

                //Declare integer used in do-while loop
               int k = 1;

                //Do-while loop to randomly generate coordinate pair and fill
rectangle corresponding to it (drives the game)
                do
                {
                    // Count number of games played for statistics
                    score_recorder.Game_Counter();

                    // Resets k = 1 for a new game (if player decides to play
again)
                    k = 1;

                    for (i = 0; i < 6; i++) // Loops for length of sequence
                    {

                        //Create four rectangles
                        LCD.Clear( FEHLCD::White ); // Clear background
                        // Top-Left of screen
                        LCD.SetFontColor( SCARLET );
                        LCD.DrawRectangle(0, 0, 160, 120);
                        // Top-Right
                        LCD.SetFontColor( BLUE );
                        LCD.DrawRectangle(160, 0, 160, 120);
                        // Bottom-Left
                        LCD.SetFontColor( GREEN );
                        LCD.DrawRectangle(0, 120, 160, 120);
                        // Bottom-Right
                        LCD.SetFontColor( YELLOW );
                        LCD.DrawRectangle(160, 120, 160, 120);

                        //Randomly generate x and y coordinate pair
                        x = rand() % 321;
                        y = rand() % 241;

                        //Check which rectangle randomly generated pair is in
and convert to a rectangle number
                        if (x < 160 && y < 120) //Check to see if rand
generate pair is in scarlet(value of 1) rectangle
                        {
                            sequence1[i] = scarlet;
                            //Fill rectangle that corresponds to randomly
generated coordinate pair
                            LCD.SetFontColor( SCARLET );
                            LCD.FillRectangle(0, 0, 160, 120);
                            Sleep(1000);    // Allows player to see color
                            // Clear rectangle (flashes the color of
rectangle)
```

```
                            LCD.SetFontColor( WHITE );
                            LCD.FillRectangle(0, 0, 160, 120);
                            Sleep(500);     // Pause between flashing
rectangle colors in sequence
                        }
                        else if (x > 160 && y < 120) // blue
                        {
                            sequence1[i] = blue;
                            LCD.SetFontColor( BLUE );
                            LCD.FillRectangle(160, 0, 160, 120);
                            Sleep(1000);
                            LCD.SetFontColor( WHITE );
                            LCD.FillRectangle(160, 0, 160, 120);
                            Sleep(500);
                        }
                        else if (x < 160 && y > 120) // green
                        {
                            sequence1[i] = green;
                            LCD.SetFontColor( GREEN );
                            LCD.FillRectangle(0, 120, 160, 120);
                            Sleep(1000);
                            LCD.SetFontColor( WHITE );
                            LCD.FillRectangle(0, 120, 160, 120);
                            Sleep(500);

                        }
                        else if (x > 160 && y > 120) // yellow
                        {
                            sequence1[i] = yellow;
                            LCD.SetFontColor( YELLOW );
                            LCD.FillRectangle(160, 120, 160, 120);
                            Sleep(1000);
                            LCD.SetFontColor( WHITE );
                            LCD.FillRectangle(160, 120, 160, 120);
                            Sleep(500);
                        }
                    }

                    // Player input
                    for (i = 0; i < 6; i ++)
                    {
                        // Wait for player input
                        while(!LCD.Touch(&x, &y) )
                        {}
                        // Allows code to continue execution upon release
                        while (LCD.Touch(&x_touch, &y_touch))
                        {}

                        if (x < 160 && y < 120) //Check to see if touch is in
scarlet rectangle
                        {
                            sequence2[i] = scarlet;
                            //Fill rectangle that corresponds to randomly
generated coordinate pair
                            LCD.SetFontColor( SCARLET );
                            LCD.FillRectangle(0, 0, 160, 120);
                            Sleep(250);
```

```
                        // Clear rectangle so that the color appears to
have flashed for a moment
                        LCD.SetFontColor( WHITE );
                        LCD.FillRectangle(0, 0, 160, 120);
                    }
                    else if (x > 160 && y < 120) // blue
                    {
                        sequence2[i] = blue;
                        LCD.SetFontColor( BLUE );
                        LCD.FillRectangle(160, 0, 160, 120);
                        Sleep(250);
                        LCD.SetFontColor( WHITE );
                        LCD.FillRectangle(160, 0, 160, 120);
                    }
                    else if (x< 160 && y > 120) // green
                    {
                        sequence2[i] = green;
                        LCD.SetFontColor( GREEN );
                        LCD.FillRectangle(0, 120, 160, 120);
                        Sleep(250);
                        LCD.SetFontColor( WHITE );
                        LCD.FillRectangle(0, 120, 160, 120);
                    }
                    else if (x > 160 && y > 120) // yellow
                    {
                        sequence2[i] = yellow;
                        LCD.SetFontColor( YELLOW );
                        LCD.FillRectangle(160, 120, 160, 120);
                        Sleep(250);
                        LCD.SetFontColor( WHITE );
                        LCD.FillRectangle(160, 120, 160, 120);
                    }

                    //Compare each input of user to randomly generated
values
                    if (sequence1[i]!= sequence2[i])
                    {

                        LCD.Clear( FEHLCD::White );
                        LCD.SetFontColor( SCARLET );
                        Buzzer.Buzz(500);
                        LCD.WriteLine("Sorry! You lost.");  // on-screen
message
                        LCD.Write("Score: ");

                        // Display score on screen
                        score_recorder.Display_Current_Score();
                        // Count number of games played for statistics
                        score_recorder.Game_Counter();
                        //Record stats from game
                        score_recorder.Stats();

                        LCD.Write("Tap screen");

                        while( !LCD.Touch(&x, &y) )
                        {}
                        while (LCD.Touch(&x_touch, &y_touch))
```

```
                        {}

                        Sleep(1000);    // Delay between player input and
displaying replay menu

                        //Clear background
                        LCD.Clear( FEHLCD::White );

                        //Switch-case to allow user to play again or
quit.
                        // Declaring replay array
                        FEHIcon::Icon replay[2];
                        // Replay menu labels
                        char replay_labels[2][20] = {"Play Again",
"Return to Menu"};
                        // Displaying menu on screen
                        FEHIcon::DrawIconArray(replay, 2, 1, 10, 10, 2,
2,replay_labels, GREEN, SCARLET);

                        while( !LCD.Touch(&x, &y) )
                        {}
                        while (LCD.Touch(&x_touch, &y_touch))
                        {}

                        //Create if-else if statement to set expression
for a switch-case.
                        if (y > 0 && y < 110)   // Play Again
                        {
                            choice = 1;
                            score_recorder.Reset_Score();   // Reset
score for next game
                        }
                        else if (y > 110 && y < 220)    // Quit
                        {
                            choice = 2;
                        }

                        //Insert switch case to execute choice
                        switch (choice)
                        {
                            case (1):
                            //Continue do-while loop to play again.
                                k = 2;
                                break;
                            case (2):
                            //Break out of do-while loop to return to
main menu.
                                k = 0;
                                break;
                        }

                        //Break out of if statement
                        break;
                    }

                }
                    if (k == 1)
```

```
                        {
                            LCD.Clear( FEHLCD::White );
                            LCD.SetFontColor( SCARLET );
                            Buzzer.Tone( FEHBuzzer::Bf5, 150 );
                            Buzzer.Tone( FEHBuzzer::A5,  75 );
                            Buzzer.Tone( FEHBuzzer::Bf5, 150 );
                            Buzzer.Tone( FEHBuzzer::A5,  75 );
                            Buzzer.Tone( FEHBuzzer::Bf5, 150 );
                            LCD.WriteLine("Nice job! :D");  // on-screen
message
                            LCD.WriteLine("Tap screen to continue");
                            while(!LCD.Touch(&x, &y))
                            {}
                            k = 1;
                            score_recorder.Score_Setter();
                            Sleep(1000);
                        }
                    } while(k >= 1);

                break;  //Break case 2

            }   //End switch case

        } //End if statement
        else if (y > 65 && y < 120) // Rules
        {
            LCD.Clear( FEHLCD::White );
            LCD.SetFontColor( BLACK );
            LCD.WriteLine("Rules");
            LCD.WriteLine("A sequence is given by the change in button
colors.");
            LCD.WriteLine("Tap the buttons in the ");
            LCD.WriteLine("same sequence.");
            LCD.WriteLine("Watch how long you can ");
            LCD.WriteLine("last. Good luck!");
            LCD.WriteLine(" ");
            LCD.WriteLine("Tap screen for menu");
            while(!LCD.Touch(&x, &y))
            {}
            while (LCD.Touch(&x_touch, &y_touch))
            {}
        }
        else if (y > 120 && y < 175)    // stats
        {
           LCD.Clear( FEHLCD::White );
           LCD.SetFontColor( BLACK );
           score_recorder.Display_All_Scores();
           while(!LCD.Touch(&x, &y))
           {}
           while (LCD.Touch(&x_touch, &y_touch))
           {}
        }
        else if (y > 175 && y < 230)    //Credits
        {
            LCD.Clear( FEHLCD::Black );
            LCD.SetFontColor( FEHLCD::White );
            LCD.WriteLine("Credits");
```

```
            LCD.WriteLine("Creators:");
            LCD.WriteLine("Tammy Nguyen-Huynh & ");
            LCD.WriteLine("Claire Forrestal");
            LCD.WriteLine("The Creators would like to thank:");
            LCD.WriteLine("Dr. Schlosser, Sean Dunphy");
            LCD.WriteLine("Jennifer Scott and Matt ");
            LCD.WriteLine("Polatas");
            LCD.WriteLine(" ");
            LCD.WriteLine("Tap screen for menu");

            while(!LCD.Touch(&x, &y))
            {}
            while (LCD.Touch(&x_touch, &y_touch))
            {}
        }
    }
    return 0;

}
Statistics::Statistics()    // Constructor function for class Statistics
{
    int i;
    for (i = 0; i < 14; i++)
    {
        all_scores[i] = 0;
    }
    score = 0;
    game = 0;
}

void Statistics::Game_Counter() // Counts number of games played to keep
track of scores from each game
{
    game = game + 1;
}

void Statistics::Score_Setter() // Adds to score after player successfully
completes a sequence
{
    score = score + 10;
}

void Statistics::Stats()  // Stats function stores scores
{
    all_scores[game-1] = score;
}

void Statistics::Reset_Score()  // Resets score for a new game
{
    score = 0;
}

void Statistics::Display_Current_Score()    // Displays score on screen
{
    LCD.WriteLine(score);
}
```

```
void Statistics::Display_All_Scores()   // Displays scores from multiple
games
{
    int i;
    for (i = 0; i < 14; i++)
    {
        LCD.Write("Game ");
        LCD.Write(i+1);
        LCD.Write(" ");
        LCD.WriteLine(all_scores[i]);
    }
}
```