Software Design Project Documentation

Engineering 1281H

Autumn, 2017

Nicole Korczak, Seat 27

Keegan Lahm, Seat 28

A. Theiss 10:20 AM

Date of Design: 11/29/17 – 12/06/17

Date of Submission: 12/06/17

1. Introduction

When faced with a problem, engineers consult the design process to formulate a high-quality solution. For our project we incorporated the engineering design process, steps of which may be seen in Appendix A, to create software that satisfies the needs of our target audience. Our game is called The Impossible (but still easier than FEH) quiz Christmas Edition. In this game the user is prompted with a series of riddles which they must solve in order to advance to the next stage. Each level consists of a short instruction statement. The user is given three lives. An incorrect answer for any question results in the loss of a life. Once the user has lost all three lives, the game is over, and the user is asked whether they would like to play again. Although our game uses Ohio State and SpongeBob references, this game is designed to target a wide-ranging audience of pre-teens to adults.

2. Experimental Methodology

After being presented with the game design project, we asked questions to clarify specifications and requirements. A list of possible game ideas, along with their characteristics and several themes were then brainstormed. They are listed in Tables 1 and 2 on the next page. After careful consideration, it was decided that the impossible quiz with a Christmas theme was the most entertaining and time-effective option.

Table 1: Brainstormed game ideas.

Game Idea	Characteristics
Impossible quiz	Series of questions, life system, levels, touchscreen
Snake.io	Buttons, touchscreen, randomly generated block rotation
Flappy bird	Buttons, timing system, avoiding objects, random placement of blocks
Temple run	Multiplayer, buttons and touchscreen, avoid obstacles, graphics
Target shooting	Moving targets, arrows, timing, touchscreen

Table 2: Brainstormed theme ideas.

Theme Ideas
Christmas
Ohio State
Engineering parody

Prior to coding the game, an algorithm representing the "thought process" our game would go through was created. This algorithm can be seen in Appendix B. Next, a blank qtCreator document was commented and then code was written. The code was tested as each section was written and then necessary corrections were made.

3. Results and Description

The Impossible (but still easier than FEH) Quiz Christmas Edition uses only nine variables and three functions which are listen in Tables 3 and 5 below and on the next page. Table 4 shows some assumptions used when creating the game.

Table 3: Variables used and how they were used.

Variables	Uses
x	x-coordinate of last touch
y	y-coordinate of last touch
number	count number of correct questions
ansminx	minimum x value accepted as an answer
ansmaxx	maximum x value accepted as an answer
ansminy	minimum y value accepted as an answer
ansmaxy	maximum y value accepted as an answer
lives	keeps track of user's lives
answer	keeps track if answer is correct or not

Table 4: Assumptions when creating game.

Assumptions
The user wants to play the game at least once if they turn on the Proteus.
The user knows that they must touch the screen with the stylus.
The user is familiar with SpongeBob.

Table 5: Description of functions.

Function	Purpose	Type	# of calling parameters	Example
Test	Receives the coordinates of the user's touch, and compares it to the pre-determined coordinates of the correct answer and returns whether a value for the variable answer, (1 for yes, 0 for no)	Integer	8	test(float x, float y, float xmin, float xmax, float ymin, float ymax, int lives, int number)
Lesslives	Decrements total lives if the answer is incorrect.	Integer	2	lesslives(in tans, int lives)
Correct	Displays a message to the user if the answer is correct.	Void	1	correct(int ans)

4. Discussion

There were problems with reading in the correct answer from the Proteus. At the beginning, coordinates required a lot of guess and check. Also, a day was spent trying to figure out why the Proteus was printing that the answer to a specific question was incorrect when the correct area was touched. Eventually it was realized that the Proteus required a sleep after a touch, otherwise it would read in the same coordinates multiple times. This caused the program to fall into later coordinate checks and caused an error message.

However, after these errors were fixed, the game performed as expected. When the user selected the correct answer, a congratulations message was displayed and when the user made a mistake, constructive criticism was given. For most questions, the area in which the user may touch while still getting a “correct” answer was pretty accurate. For a few of the questions, the stylus must touch a certain part of the correct answer to be recognized as correct. Some additional limitations are: some of the words/phrases are too long and parts of words are cut into two lines and that the Proteus records the last position on the screen before the stylus is lifted off the screen.

5. Conclusion

Overall, by using the engineering design process, we were able to successfully create a game to satisfy our target consumer. We learned that the engineering design process is not performed in a set order but rather steps are revisited as the design progresses.

APPENDIX A

Engineering Design Process

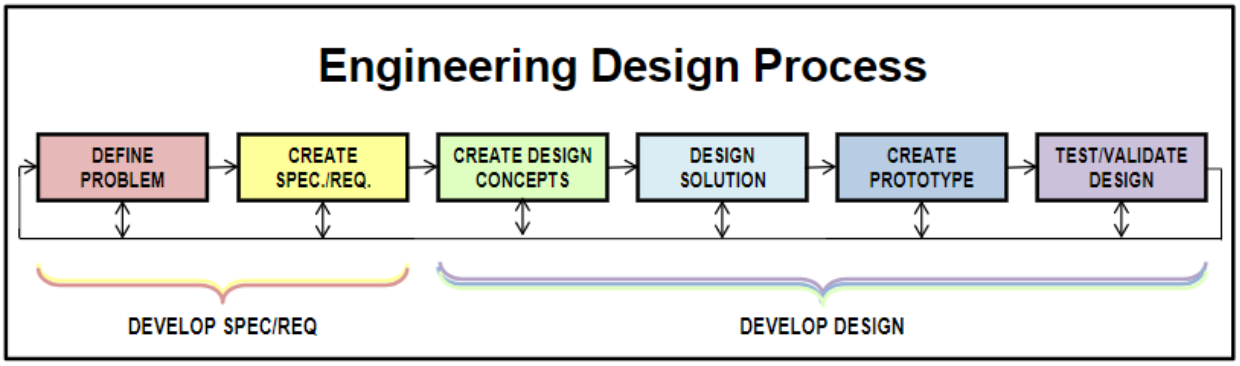

Figure A1: The Engineering Design Process used in this project.

We incorporated each step in this project. The specific steps we took which fall under each category are listed in Table A1 below.

Step in Process	What we did
Define Problem	Problem: To create an entertaining game with a title screen, directions, and credits that can be replayed.
Create Spec./Req.	1. Can the game be multiplayer? Are there any extra points if it is? 2. What must be included in the user manual? 3. How long do the rules and instructions have to be? 4. How to implement graphical interface? 5. How do we utilize the touchscreen?
Create Design Concepts	Brainstormed ideas

	1. Impossible quiz: series of questions, life system, 20 levels, touchscreen. 2. Snake.io: buttons, touchscreen, randomly generated block rotation. 3. Flappy bird: buttons, timing system, avoiding objects, random placement of blocks. 4. Temple run: multiplayer, buttons and touchscreen, avoid obstacles, graphics. 5. Target shooting: moving targets, arrows, timing, touchscreen.
Design Solution	We created an algorithm which stated what the code would do. This can be seen in Appendix B.
Create Prototype	Actual written code.
Test/Validate Design	Testing the code on the Proteus.

Table 1: How we used the engineering design process.

APPENDIX B

Algorithm

Algorithm:

I. Display title page
    a. Button for directions
    b. Button for start
    c. Name of game
II. Display directions
    a. Tells user about lives
    b. Tells user to answer riddle based on specific directions for that question/level
    c. Continue button
III. Directions for specific riddle/question
IV. Wait for user to answer
    a. If correct, advance to next level and display congratulations message
    b. If incorrect, take away one life, and display error message
V. Repeat until all lives are gone or user completes all levels
VI. Display score along with encouraging/discouraging message based on results
VII. Ask if user wants to play again
VIII. Play again/quit game
IX. Show credits when user decides to quit

APPENDIX C

Source Code

```
#include <FEHLCD.h>
#include <FEHIO.h>
#include <FEHUtility.h>

//make a function prototype to check whether the tap is whithin
//the bounds of the answer and returns either a 1 or 0 if it is false
int test(float x, float y, float xmin,float xmax, float ymin, float ymax,int lives,int number);

//Create another function to decrement lives
int lesslives(int answer,int lives);

//Create a function to print to the screen whenever the user gets the correct answer.
void correct(int answer);

int main(void)
{
   //Position of last touch
   float x,y;

   //initialize number correct variable
   int number;


   //Position of solution
   float ansminx,ansmaxx,ansminy,ansmaxy;

   //Number of lives
      int lives;

   //tells the game whether the user was correct or not to exit the loop
      int answer=0;

   LCD.Clear( FEHLCD::Black );
   LCD.SetFontColor( FEHLCD::White );

   //Loop for rerun
   do
   {
      //reset
      lives=3;
      number=0;
      answer=0;

      LCD.Clear(RED);

      LCD.SetFontColor(WHITE);

      //Welcome screen
      LCD.WriteLine("Welcome to the Impossible (But easier than FEH) Test Christmas Edition.");
      LCD.WriteLine("Tap the screen to continue to the directions.");

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

      LCD.Clear(RED);

         //Write the directions to the game
```

```
    LCD.WriteLine( "Just follow the directions for each level/question." );
    LCD.WriteLine( "You get three lives. ");
    LCD.WriteLine("Basically, just don't be stupid." );
    LCD.WriteLine("Tap the screen to continue.");

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

//Question #1- just a riddle
  while(lives!=0&&answer!=1)
  {
      LCD.Clear(WHITE);
      LCD.SetFontColor(RED);

     //print question
     LCD.WriteLine("Where does Santa live");

     LCD.WriteAt("North Pole",10,45);
     LCD.WriteAt("He doesn't exist",10,90);
     LCD.WriteAt("In your heart",10,135);
     LCD.WriteAt("My basement",10,180);

     //set the answer pixels
     ansminx=0; ansmaxx=319; ansmaxy=170; ansminy=120;

     //Wait for touch
     while(!LCD.Touch(&x,&y)){}

    //Call the test function
    answer=test(x,y,ansminx,ansmaxx,ansminy,ansmaxy,lives,number);

    //Call the less lives function
    lives=lesslives(answer,lives);

     //Sleep for touch sensor
     if(answer==1)
     {
        LCD.Clear(GREEN);
        Sleep(1000);
        LCD.WriteLine("That is blasphemy.");
        LCD.WriteLine("I know what you were thinking.");
        Sleep(3000);
     }
  }


  //increment number
  if(answer==1)
  {
     number++;
  }

//reset the answer
  answer=0;


  /*Second question:
```

```
        * Tap the smallest ornament to continue. The smallest one is located on the position of the o in ornament.*/

    //loop until out of lives or done with second question
    while(lives!=0&&answer!=1)
    {

        LCD.Clear(GREEN);

       LCD.WriteLine("Tap the smallest ");
       LCD.WriteAt("ornament",199,3);

       //Draw ornaments
       LCD.SetFontColor(MEDIUMPURPLE);

       LCD.FillCircle(160,120,20);

       LCD.SetFontColor(DEEPSKYBLUE);
       LCD.FillCircle(30,200,15);

       LCD.SetFontColor(DARKRED);
       LCD.FillCircle(280,150,25);


       //set the answer pixels
       ansminx=170; ansmaxx=250; ansmaxy=60; ansminy=0;

       //Wait for touch
       while(!LCD.Touch(&x,&y)){}

      //Call the test function
      answer=test(x,y,ansminx,ansmaxx,ansminy,ansmaxy,lives,number);

      //decrement lives for wrong answer
      lives=lesslives(answer, lives);

      //Display the correct statement
      if(answer==1)
      {
         correct(answer);
      }
    }


    //increment number
    if(answer==1)
    {
       number++;
    }

//reset answer
answer=0;


//Question #3- just a riddle
   while(lives!=0&&answer!=1)
   {
```

```
        LCD.Clear(WHITE);
        LCD.SetFontColor(RED);

       //print question
       LCD.WriteLine("Last Christmas I gave you...");

      LCD.WriteAt("A trip to the hospital",10,45);
      LCD.WriteAt("My heart",10,90);
      LCD.WriteAt("Im a broke...",10,135);
      LCD.WriteAt("Nothing, Im a proteus.",10,180);

       //set the answer pixels
       ansminx=0; ansmaxx=300; ansmaxy=210; ansminy=150;

       //Wait for touch
       while(!LCD.Touch(&x,&y)){}

     //Call the test function
     answer=test(x,y,ansminx,ansmaxx,ansminy,ansmaxy,lives,number);

     //Call the less lives function
     lives=lesslives(answer,lives);

       //Sleep for touch sensor
       if(answer==1)
       {
          LCD.Clear(GREEN);
          Sleep(1000);
          LCD.WriteLine("This was obvious.");
          Sleep(3000);
       }
    }

   //increment number
   if(answer==1)
   {
      number++;
   }


//reset the answer
   answer=0;


//Question #4- just a riddle
   while(lives!=0&&answer!=1)
   {
       LCD.Clear(WHITE);
       LCD.SetFontColor(RED);

       //print question
       LCD.WriteLine("How much does a polar bear weigh?");

      LCD.WriteAt("Enough to break the ice ;)",10,45);
      LCD.WriteAt("A lot",10,90);
      LCD.WriteAt("Idk but I can take em",10,135);
```

```
  LCD.WriteAt("10000 Buckeyes",10,180);

  //set the answer pixels
  ansminx=0; ansmaxx=300; ansmaxy=160; ansminy=100;

  //Wait for touch
  while(!LCD.Touch(&x,&y)){}

 //Call the test function
 answer=test(x,y,ansminx,ansmaxx,ansminy,ansmaxy,lives,number);

 //Call the less lives function
 lives=lesslives(answer,lives);

  //Sleep for touch sensor
  if(answer==1)
  {
     LCD.Clear(GREEN);
     Sleep(1000);
     LCD.WriteLine("You definitely can't take him");
     Sleep(2000);
  }


}

//increment number
if(answer==1)
{
   number++;
}


//reset answerand add one answer completed
answer=0;


 //Fifth question: Instructions- "Merry Christmas" answers in order - "and a" "happy" "new" "year"
 while(lives!=0&&answer!=1)
 {
     LCD.Clear(WHITE);
     LCD.SetFontColor(RED);

    LCD.WriteLine("Merry Christmas");

   LCD.WriteAt("New",20,100);
   LCD.WriteAt("Happy",200,100);
   LCD.WriteAt("and a",20,200);
   LCD.WriteAt("Year",200,200);


    //set the first answer pixels
    ansminx=0; ansmaxx=150; ansmaxy=239; ansminy=180;

    //Wait for touch
    while(!LCD.Touch(&x,&y)){}
```

```
  //Call the test function
  answer=test(x,y,ansminx,ansmaxx,ansminy,ansmaxy,lives,number);


  //decrement lives for wrong answer
  lives=lesslives(answer,lives);

  //Sleep for touch sensor
  if(answer==1)
  {
     LCD.Clear(GREEN);
     Sleep(1000);
  }
}

//reset answer
answer=0;


//keep in the game at same point until run out of lives or answer is correct
while(lives!=0&&answer!=1)
{

  //Reprint
  LCD.Clear(WHITE);
  LCD.SetFontColor(RED);

 LCD.WriteLine("Merry Christmas");

LCD.WriteAt("New",20,100);
LCD.WriteAt("Happy",200,100);
LCD.WriteAt("and a",20,200);
LCD.WriteAt("Year",200,200);



    //set the second answer pixels
    ansminx=180; ansmaxx=300; ansmaxy=170; ansminy=50;

    //Wait for touch
    while(!LCD.Touch(&x,&y)){}


   //Call the test function
   answer=test(x,y,ansminx,ansmaxx,ansminy,ansmaxy,lives,number);


   //decrement lives for wrong answer
   lives=lesslives(answer,lives);

   //Sleep for touch sensor
   if(answer==1)
   {
```

```
        LCD.Clear(GREEN);
        Sleep(1000);
    }


  }//end while

//reset answer
answer=0;


while(lives!=0&&answer!=1)
{

   //Reprint
   LCD.Clear(WHITE);
   LCD.SetFontColor(RED);

  LCD.WriteLine("Merry Christmas");

 LCD.WriteAt("New",20,100);
 LCD.WriteAt("Happy",200,100);
 LCD.WriteAt("and a",20,200);
 LCD.WriteAt("Year",200,200);

      //set the third answer pixels
     ansminx=0; ansmaxx=160; ansmaxy=170; ansminy=50;

      //Wait for touch
      while(!LCD.Touch(&x,&y)){}

     //Call the test function
     answer=test(x,y,ansminx,ansmaxx,ansminy,ansmaxy,lives,number);


     //decrement lives for wrong answer
     lives=lesslives(answer,lives);

     //Sleep for touch sensor
     if(answer==1)
     {
        LCD.Clear(GREEN);
        Sleep(1000);
     }


}//end of while

//reset answer
answer=0;


while(lives!=0&&answer!=1)
{
   //Reprint
```

```
  LCD.Clear(WHITE);
  LCD.SetFontColor(RED);

 LCD.WriteLine("Merry Christmas");

LCD.WriteAt("New",20,100);
LCD.WriteAt("Happy",200,100);
LCD.WriteAt("and a",20,200);
LCD.WriteAt("Year",200,200);


      //set the fourth answer pixels
      ansminx=180; ansmaxx=300; ansmaxy=239; ansminy=180;

      //Wait for touch
      while(!LCD.Touch(&x,&y)){}

     //Call the test function
     answer=test(x,y,ansminx,ansmaxx,ansminy,ansmaxy,lives,number);

     //decrement lives for wrong answer
     lives=lesslives(answer,lives);

     //Sleep for touch sensor
     if(answer==1)
     {
        LCD.Clear(GREEN);
        Sleep(1000);
     }

     //display that they got the question correct
     correct(answer);


}

//increment number
if(answer==1)
{
   number++;
}


//reset answer and add one answer completed
answer=0;


//Question #6- just a riddle
  while(lives!=0&&answer!=1)
  {
      LCD.Clear(WHITE);
      LCD.SetFontColor(RED);

      //print question
      LCD.WriteLine("HO HO HO");
```

```
  LCD.WriteAt("Stop it Patrick you're -",0,30);
  LCD.WriteAt("scaring him",0,45);
  LCD.WriteAt("Santa",0,90);
  LCD.WriteAt("Chris K.",0,135);
  LCD.WriteAt("HO",0,180);

  //set the answer pixels
  ansminx=0; ansmaxx=300; ansmaxy=70; ansminy=20;

  //Wait for touch
  while(!LCD.Touch(&x,&y)){}

 //Call the test function
 answer=test(x,y,ansminx,ansmaxx,ansminy,ansmaxy,lives,number);

 //Call the less lives function
 lives=lesslives(answer,lives);

  //Sleep for touch sensor
  if(answer==1)
  {
     LCD.Clear(GREEN);
     Sleep(1000);
  }

  //correct
  correct(answer);

}

//increment number
if(answer==1)
{
   number++;
}


//reset answer
answer=0;


//Question #7- just a riddle
  while(lives!=0&&answer!=1)
  {
      LCD.Clear(WHITE);
      LCD.SetFontColor(RED);

     //print question
     LCD.WriteLine("What is brighter than rudolph's nose");

    LCD.WriteAt("Your Future",10,45);
    LCD.WriteAt("The Clock Tower",10,90);
    LCD.WriteAt("2 rudolph's noses",10,135);
    LCD.WriteAt("A buttload of lightbulbs",10,180);

    //set the answer pixels
```

```
      ansminx=0; ansmaxx=300; ansmaxy=240; ansminy=160;

      //Wait for touch
      while(!LCD.Touch(&x,&y)){}

     //Call the test function
     answer=test(x,y,ansminx,ansmaxx,ansminy,ansmaxy,lives,number);

     //Call the less lives function
     lives=lesslives(answer,lives);

      //Sleep for touch sensor
      if(answer==1)
      {
         LCD.Clear(GREEN);
         Sleep(1000);
         LCD.WriteLine("Your Future? LOL");
         Sleep(3000);
      }
   }


   //increment number
   if(answer==1)
   {
      number++;
   }


//reset answer and add one answer completed
answer=0;



//Question #8- unwrap a present by tapping the right hand side of the screen
while(lives!=0&&answer!=1)
    {
      //Clears screen and sets it to color "SNOW"
      LCD.Clear(AZURE);
      LCD.SetFontColor(RED);

      //Prompts user to unwrap the present the right way
      LCD.WriteLine("Unwrap the present the right way.");

      //sets font color to "NAVY"
      LCD.SetFontColor(MEDIUMBLUE);

      //Creates present
      LCD.FillRectangle(120,100,60,40);

      LCD.SetFontColor (FUCHSIA);

      LCD.FillRectangle(120,110,60,20);
      LCD.FillRectangle(140,100,20,40);

      //Changes font color to "VIOLET" and draws ribbon lines
```

```
        LCD.SetFontColor(VIOLET);
        LCD.DrawLine(150,100,110,70);
        LCD.DrawLine(150,100,190,70);

        LCD.DrawLine(150,100,120,60);
        LCD.DrawLine(150,100,180,60);

        LCD.DrawLine(150,100,100,50);
        LCD.DrawLine(150,100,200,50);

        LCD.DrawLine(150,100,110,80);
        LCD.DrawLine(150,100,190,80);

        //sets min and max acceptable answer values
        ansminx=150; ansmaxx=230; ansmaxy=110; ansminy=0;

        //wait for touch
        while(!LCD.Touch(&x,&y)){}

        //calls the test function
        answer=test(x,y,ansminx,ansmaxx,ansminy,ansmaxy,lives,number);

        //decrements lives for wrong answer
        lives=lesslives(answer,lives);
        Sleep(1.0);
        correct(answer);
    }

//increment number
if(answer==1)
{
   number++;
}


//reset answer
answer=0;


//Question #9- last riddle
  while(lives!=0&&answer!=1)
  {
      LCD.Clear(WHITE);
      LCD.SetFontColor(RED);

      //print question
      LCD.WriteLine("What do you want for Christmas?");

     LCD.WriteAt("A New Car",10,45);
     LCD.WriteAt("To pass FEH",10,90);
     LCD.WriteAt("Alabama to burn",10,135);
     LCD.WriteAt("Coal- I know my worth",10,180);

     //set the answer pixels
     ansminx=0; ansmaxx=300; ansmaxy=110; ansminy=70;
```

```
  //Wait for touch
  while(!LCD.Touch(&x,&y)){}

 //Call the test function
 answer=test(x,y,ansminx,ansmaxx,ansminy,ansmaxy,lives,number);

 //Call the less lives function
 lives=lesslives(answer,lives);

  //Sleep for touch sensor
  if(answer==1)
  {
     LCD.Clear(GREEN);
     Sleep(1000);
     LCD.WriteLine("Yeah, Good Luck...");
     Sleep(3000);
  }
}

//increment number
if(answer==1)
{
   number++;
}

//reset answer
answer=0;


//Win message, ask if they would like to replay/quit
if(number==9)
{
   LCD.Clear(BLACK);
   LCD.SetFontColor(RED);
   LCD.WriteLine("GAME OVER");
   Sleep(1500);
   LCD.WriteLine("JK");
   Sleep(1000);
   LCD.WriteLine("");
   LCD.WriteLine("You spent that long trying");
   LCD.WriteLine("to memorize the answers?");
   LCD.WriteLine("Get a life.");
   LCD.WriteLine("You get a 9/10.");
   LCD.WriteLine("Just because.");
   Sleep(5000);

}

else
{
   LCD.Clear(BLACK);
   LCD.SetFontColor(RED);
   LCD.WriteLine("GAME OVER");
   LCD.WriteLine("Your future is darker than this screen");
   Sleep(2000);
   LCD.Write("You scored a ");
```

```
        LCD.WriteLine(number);
        LCD.WriteLine("...");
        Sleep(2000);
        LCD.WriteLine("Oh yeah, you suck.");
        Sleep(1000);
      }

      LCD.Clear(WHITE);
      LCD.WriteLine("Would you like to play again?");
      LCD.WriteAt("Yes",0,50);
      LCD.WriteAt("No",0,100);


      //set the answer pixels
      ansminx=0; ansmaxx=300; ansmaxy=80; ansminy=20;

      //Wait for touch
      while(!LCD.Touch(&x,&y)){}

     //Call the test function
     answer=test(x,y,ansminx,ansmaxx,ansminy,ansmaxy,lives,number);


      Sleep(1000);

    }while(answer==1);//loop if they click yes

    //Credits
    LCD.Clear(BLACK);
    LCD.SetFontColor(WHITE);
    LCD.WriteLine("Game Inspiration:");
    LCD.WriteLine("Impossible Quiz");
    Sleep(4000);


    LCD.Clear(BLACK);
    LCD.WriteLine("Sarcastic Comments ");
    LCD.WriteLine("courtesy of:");
    LCD.WriteLine("Keegan Lahm");
    Sleep(4000);
    LCD.Clear(BLACK);
    LCD.WriteLine("Harsh put-downs ");
    LCD.WriteLine("courtesy of:");
    LCD.WriteLine("Nicole Korczak");
    Sleep(4000);

    //instruct user to turn of the proteus
    LCD.Clear(WHITE);
    LCD.SetFontColor(BLACK);
    LCD.WriteLine("Turn off the proteus.");
    LCD.WriteLine("This is the only time I'll say please.");
    Sleep(3000);



return 0;
```

```
}


   //Accept the touch points and the coordinates of an acceptable answer. It returns whether or not the answer was
correct
   int test(float x,float y, float xmin, float xmax, float ymin, float ymax,int lives,int number)
   {

      //initialize answer variable
      int answer;

      //if else to determine if it was inside acceptable range
      if(x<=xmax&&x>=xmin&&y<=ymax&&y>=ymin)
      {
          //Set variable to return that it was correct
          answer=1;
      }

      else
      {
         LCD.Clear(RED);
         LCD.SetFontColor(WHITE);

         if(lives-1>=0&&number<9)
         {
          //display message
          LCD.WriteLine("Wow, that was ");
          LCD.WriteLine("your guess?");


         Sleep(3000);


         //Print lives
         LCD.Write("You have ");
         LCD.Write(lives-1);
         LCD.WriteLine(" live(s) left.");
         Sleep(1000);
         }

         //answer is wrong
         answer=0;

      }


         //return answer
         return answer;
      }

//subtract a life for a wrong answer
int lesslives(int ans,int lives)
{
   if(ans==0)
   {
     lives--;
```

```
    }

    return lives;
}

//Function to print that the user has gotten the correct answer
void correct(int ans)
{

    if(ans==1)
    {

        LCD.Clear(WHITE);
        LCD.SetFontColor(RED);

        LCD.WriteLine("Congrats, you're not ");
        LCD.WriteLine("terribly stupid!");
        Sleep(1000);
    }


        Sleep(3000);
}
```

THE IMPOSSIBLE (but still easier than FEH) GAME
CHRISTMAS EDITION USER GUIDE

Overview: This game is a series of questions and riddles designed to trick and annoy the player. To play, you must answer the questions correctly and not lose all three of your lives. A wrong answer results in the loss of a life. Reach the end of the game with one life to be competent. Reach the end with two lives to be kind of smart. Reach the end of the game with three lives and either you're a try hard or a genius (Not sure which).

Goal: The goal of our game is to pass all nine levels while maintaining as many lives as possible.

How to answer questions: To answer all questions you must touch the correct coordinates of the right answer. If you do this then you will receive a compliment message. If not, you will receive a harsh comment. Do not touch the compliment or harsh comment screens, as they are set to sleep for a few seconds. Read the directions carefully and make sure to think outside the box.

Quitting the game: To quit the game simply press no when the question is prompted at the end of the game. Then you will see the credits and be prompted to turn off the Proteus.