```
// generated by Fast Light User Interface Designer (fluid) version 1.0110

#include "labfin.h"
//
```

1 Specification

```
// This is a soccer shoting game. Players need to shoot the ball without
// letting the ball be captured by the goalie so the player could score.
// The goalie will move horizontally from left side of the goal to the
// right. Platers click the screen to shoot the ball. If they miss the
// goal, the player would have to start over again. If they score,
// there is a rank system to cout their rank level. The goalie will move
// faster and faster each time the player scores

//
```

2 Analysis

```
// There are two buttons on the homepage of this game:the "instructions"
// button and the "start" button. by clicking the "instructions" button,
// it willlead the player to another window which shows instructions of
// the game and how to play it. then they will click the "return"
// button to return to the home screen. Then they click the "start"
// button to start the game. The player needs to click on the screen
// to shoot the ball. If they score, their rank level will get higher.
// If the goalie catches the ball. the rank level will drop all the
// way down to the beginning again

//
```

3 Design

//

```
  - The game logo is on top of the home screen.
  - the two buttons are on the left side of the home screen called
  "press start" and "instructions".
  -when you click on "instructions", it will take you to another
  window which would show the instructions, and there would be a
  return button to go back to the home screen.
  -the press start button will take you to another window which the
  game exists.
  -the game window, would have a soccer ball and a goal, and near the
  goal would be a zombie goalie running left and right.
```

//

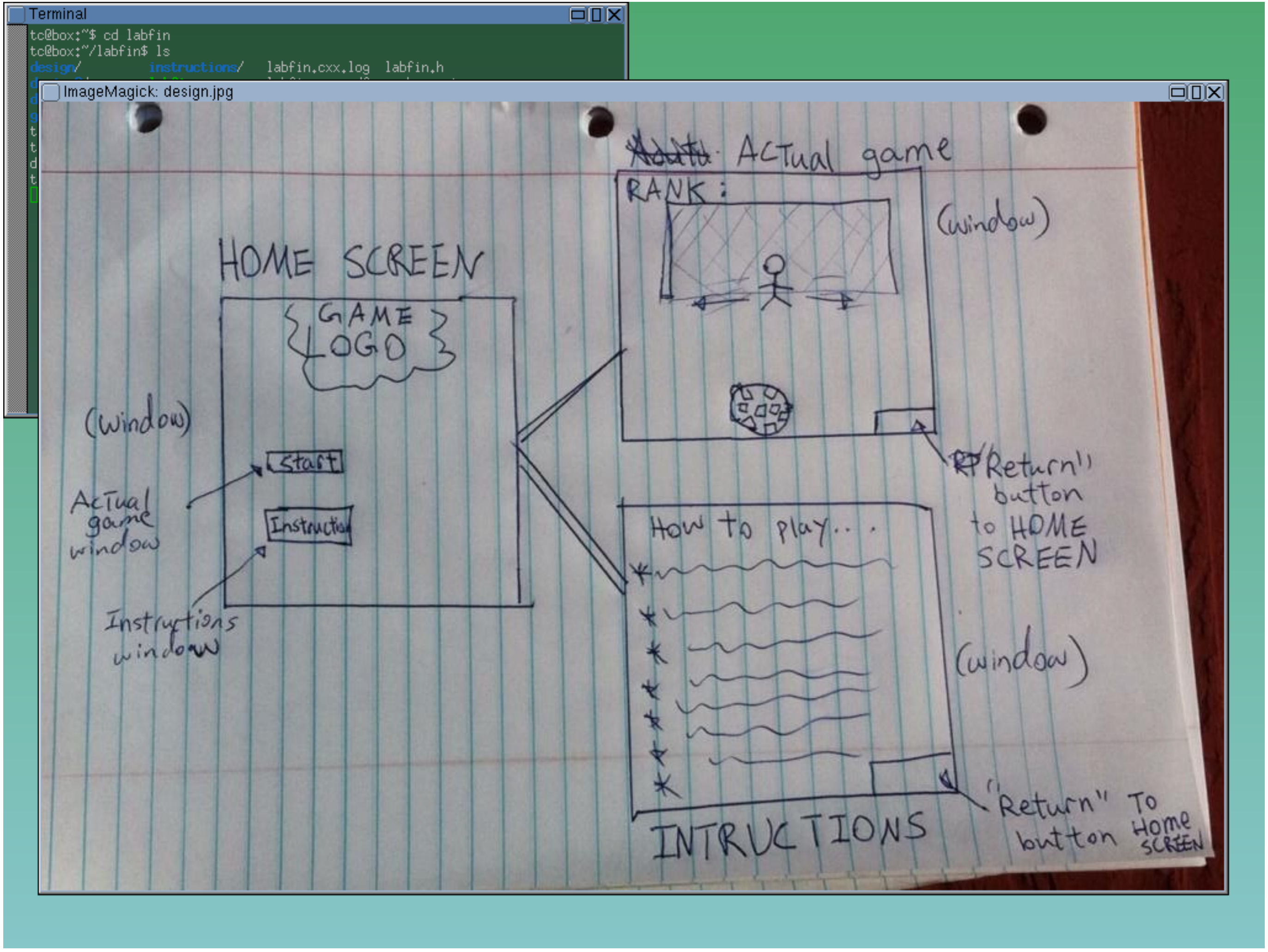
4 Implementation

```
#include <sstream>
#include <iomanip>
#include <sys/unistd.h>
using namespace std;
#include <FL/Fl_GIF_Image.H>
#include <FL/Fl_JPEG_Image.H>
#include <FL/Fl_PNG_Image.H>
static const int N1 = 8; // dribble (intro)
static Fl_GIF_Image* dribble_images[N1];

void load_images() {
  for(int i=0; i < N1; ++i)
{
 std::ostringstream oss; oss << i;
 std::string s = "dribble/dribble0"+oss.str()+".gif";
 dribble_images[i] = new Fl_GIF_Image(s.c_str());
}
}

void dribble_animate(void*) {
  static int i = 0;
Dribble->image(dribble_images[i]);
i = (i + 1) % N1;
Dribble->parent()->redraw();
double t = 1.0/8;
Fl::repeat_timeout(t,dribble_animate);
}

Fl_Double_Window *IScreen=(Fl_Double_Window *)0;

Fl_Double_Window *MainScreen=(Fl_Double_Window *)0;

Fl_Box *Dribble=(Fl_Box *)0;

Fl_Box *Gamelogo=(Fl_Box *)0;

Fl_Button *Instructions=(Fl_Button *)0;

static void cb_Instructions(Fl_Button*, void*) {
  MainScreen->hide();
IScreen->show();
}

Fl_Button *Startgame=(Fl_Button *)0;

int main(int argc, char **argv) {
  { IScreen = new Fl_Double_Window(535, 500);
```

```
    IScreen->end();
  } // Fl_Double_Window* IScreen
  { MainScreen = new Fl_Double_Window(535, 500);
    MainScreen->color((Fl_Color)FL_GRAY0);
    { Dribble = new Fl_Box(350, 350, 74, 101);
    } // Fl_Box* Dribble
    { Fl_Box* o = Gamelogo = new Fl_Box(125, 25, 300, 180);
      o->image(new Fl_PNG_Image("gamelogo/gamelogo.png"));
    } // Fl_Box* Gamelogo
    { Fl_Button* o = Instructions = new Fl_Button(100, 350, 120, 40);
      Instructions->callback((Fl_Callback*)cb_Instructions);
      o->image(new Fl_JPEG_Image("instructions/instructions.jpg"));
    } // Fl_Button* Instructions
    { Fl_Button* o = Startgame = new Fl_Button(100, 285, 120, 40);
      o->image(new Fl_JPEG_Image("startgame/startgame.jpg"));
    } // Fl_Button* Startgame
    MainScreen->end();
  } // Fl_Double_Window* MainScreen
  IScreen->hide();
load_images();
Fl::add_timeout(0,dribble_animate);
  MainScreen->show(argc, argv);
  return Fl::run();
}
//
```

5 Test

// For test, we only need to compile the progam using fltk and open it to see if any error has occured. if no error occured, we would do ./labfin to see if it has compiled the way we intended it to. and that the things we want there is displaying. The orginal image for some reason is not displaying. so i took a screenshot of it. design.png



```
tc@box:~$ cd labfin
tc@box:~/labfin$ ls
design/          instructions/   labfin.cxx.log  labfin.h
```