Course Project Documentation

CS 101 Project
PIANO TILES
TEAM ID : 370

Yash Nimje	14D070052
Pranav Raje	14D070032
Anurodh Mehta	140040042
Keshav Meena	140070002

Table Of Contents

1.Introduction

: The goal of this project is to construct a user-friendly program to enable users to play
the game “PIANO TILES”. It’s a single player-game. Piano Tiles is a game where the player's
objective is to tap on the black tiles as they appear from the top of the screen while avoiding
the white. As player tap black tiles he/she scores a point. In original game, when each black tile
is tapped, it emit a piano sound hence game is names as "PIANO TILES". If the player taps on a
white tile, the player will lose the game.
This is a super fun and highly addictive game suited for everyone. No special skills
needed, all this game asks of you is an attentive mind and fast fingers!
This is actually an Android Game application widely played by smartphone users which
we are trying to create using simple cpp as much similar to the original as possible with all the
knowledge we have.

2. PROBLEM STATEMENT

GOAL1

- The initial step of our project was to have a movement of rectangular tiles in a 4*4 rectangular grid with one black tile per row.
- For this a new row will replace the uppermost row and all the other rows move downwards. The position of tile in uppermost row is completely random from one of the column.

- GOAL 2
- This is certainly the most crucial part of our project . In this part we made a movement of tiles on clicking on correct tile i.e. the black tile in the last row by the user. If user clicks elsewhere in game window the game will end and score will be displayed.

We are successful in making movement of all black tiles on click of user.

3.
Requirements:

A) Hardware Requirements

Only a screen and mouse or touchpad.
No specific requirement

B) Software Requirements

Code blocks: Cross-platform IDE built around wxWidgets, version13.12.0.0

4. Implementation:

A) Functionality:

This is an arcade game played by a single player.The game is all about the chase. In mode created by us you will be brought to a 4×4 screen with black and white rectangles. The object of the game is to tap the black tiles as the screen cascades upwards. Miss a black tile as it passes down the lower end of your screen, or tap a white tile by mistake, and your game is over. The way the screen descends on the bottom of your screen is actually quite fast and it speeds up with time.

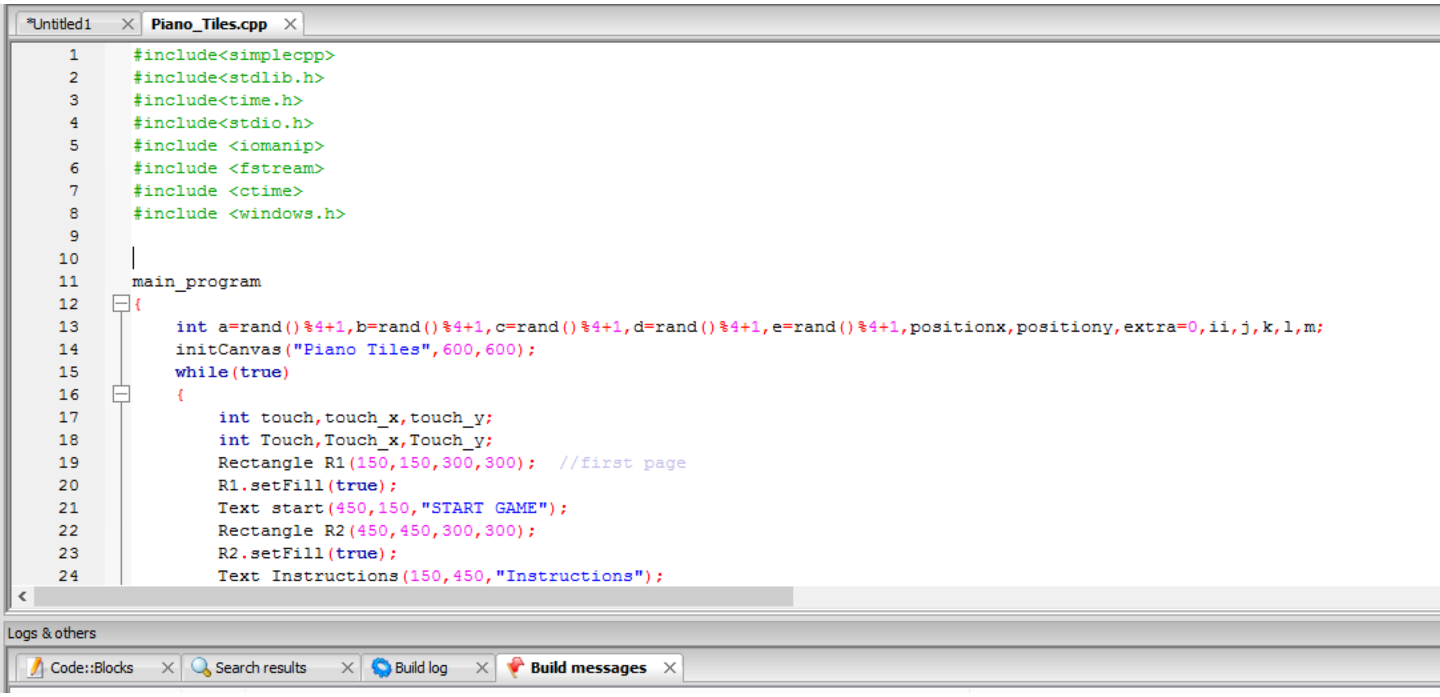
5. Testing Strategy and Data:

First test case is to see whether the tile movement is affected by the click of mouse.

Secondly new tiles coming from top of the screen are at random positions.

Thirdly if the program is showing the time duration correctly.

Code overview:

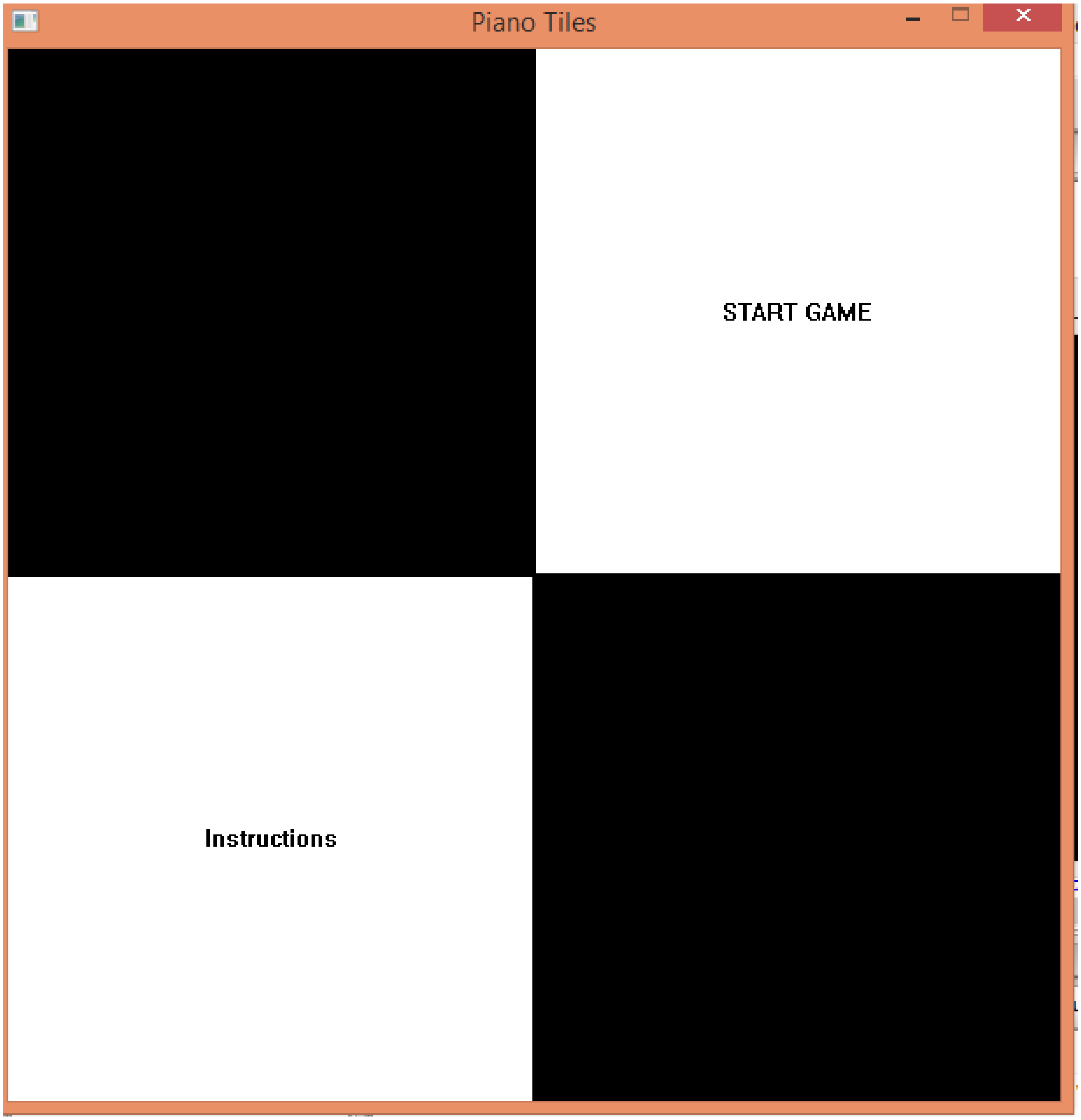


```
#include<simplecpp>
#include<stdlib.h>
#include<time.h>
#include<stdio.h>
#include <iomanip>
#include <fstream>
#include <ctime>
#include <windows.h>


main_program
{
    int a=rand()%4+1,b=rand()%4+1,c=rand()%4+1,d=rand()%4+1,e=rand()%4+1,positionx,positiony,extra=0,ii,j,k,l,m;
    initCanvas("Piano Tiles",600,600);
    while(true)
    {
        int touch,touch_x,touch_y;
        int Touch,Touch_x,Touch_y;
        Rectangle R1(150,150,300,300);  //first page
        R1.setFill(true);
        Text start(450,150,"START GAME");
        Rectangle R2(450,450,300,300);
        R2.setFill(true);
        Text Instructions(150,450,"Instructions");
```

The following libraries have been included in simplecpp.

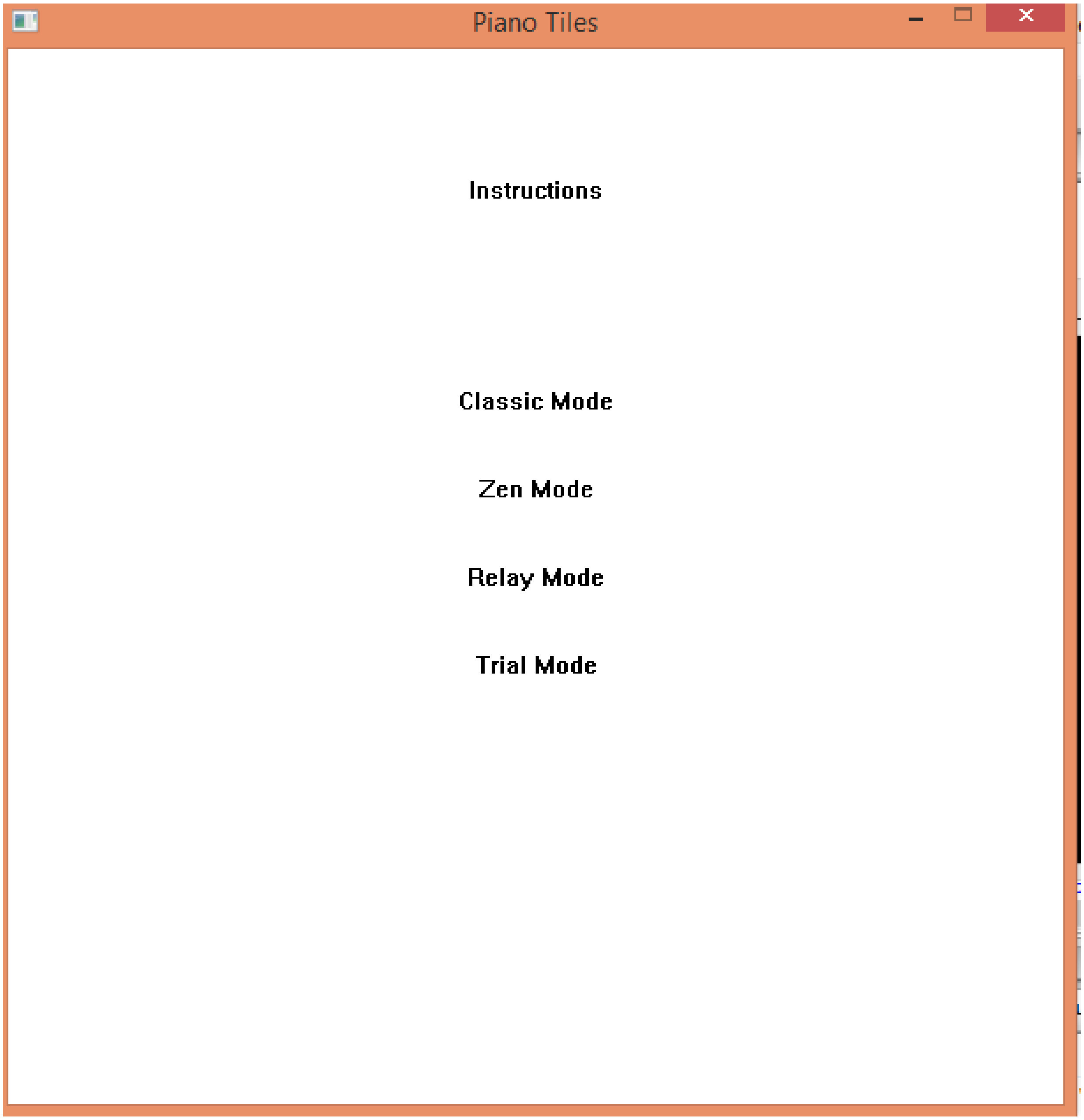
Display After Running The Program



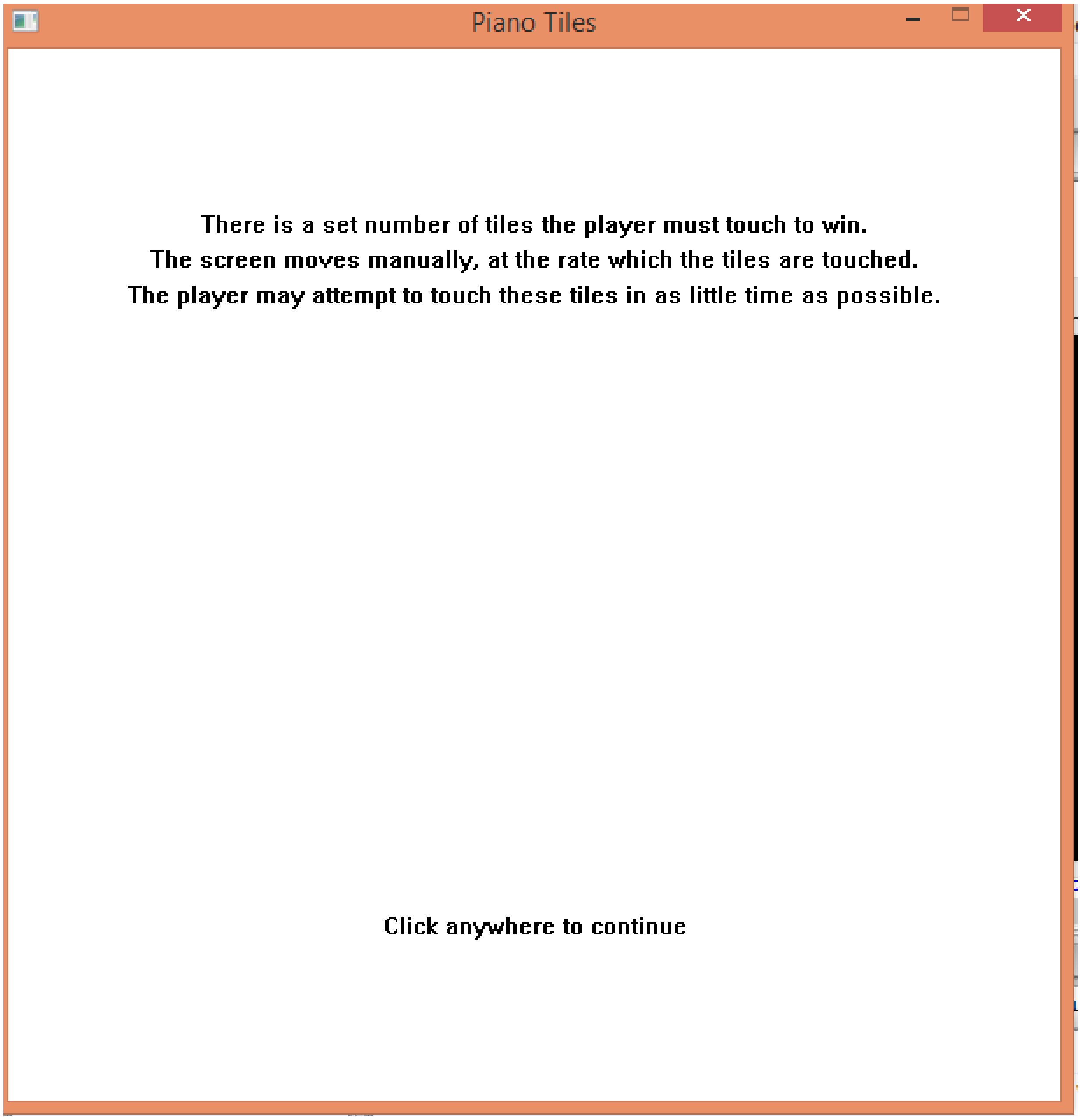
The main screen after running the program.

Click on start game to start or click on instructions for help.

INSTRUCTIONS INTERFACE:



Click the text of the mode user wants to read instruction of.

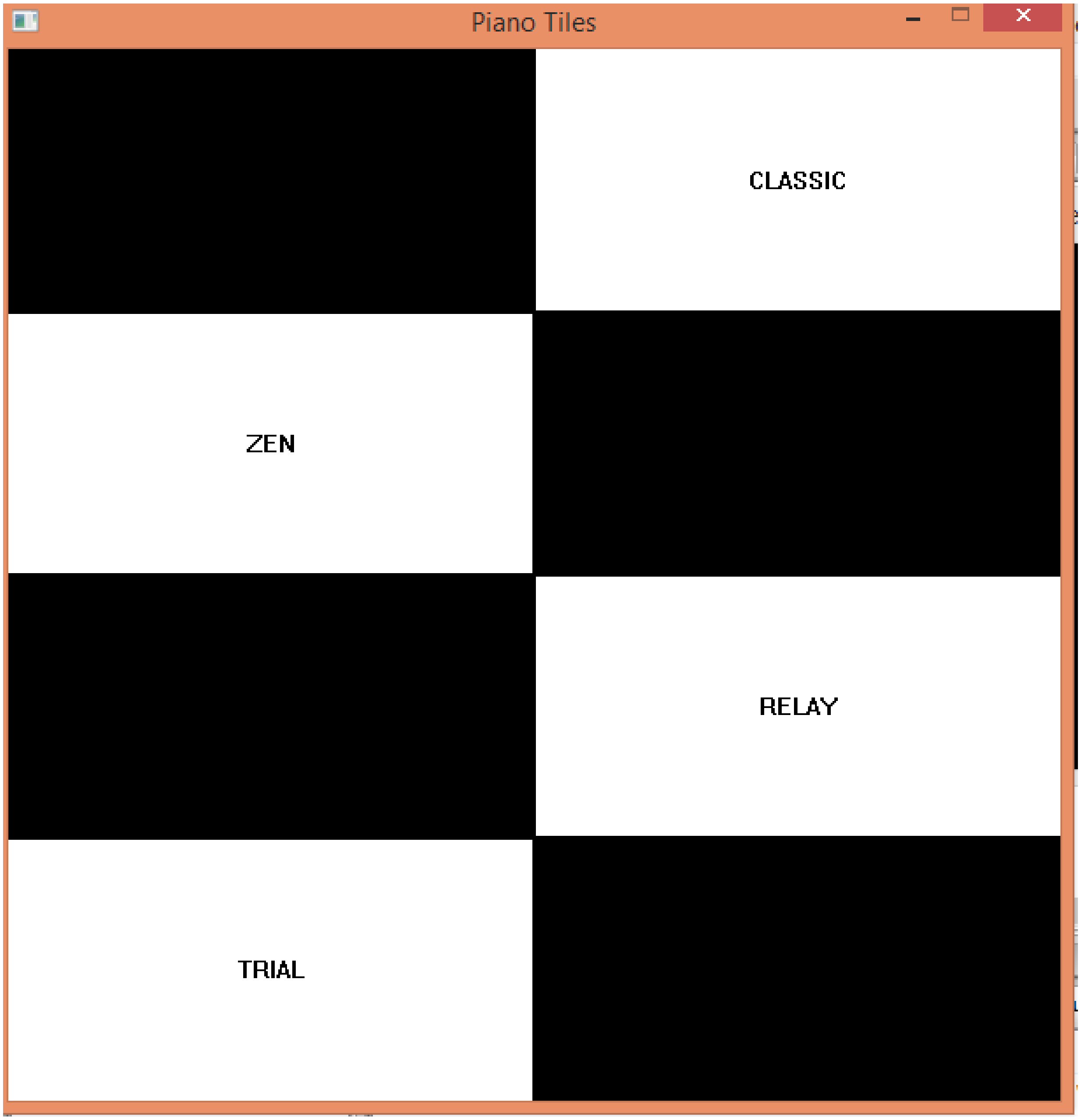


There is a set number of tiles the player must touch to win.
The screen moves manually, at the rate which the tiles are touched.
The player may attempt to touch these tiles in as little time as possible.

Click anywhere to continue

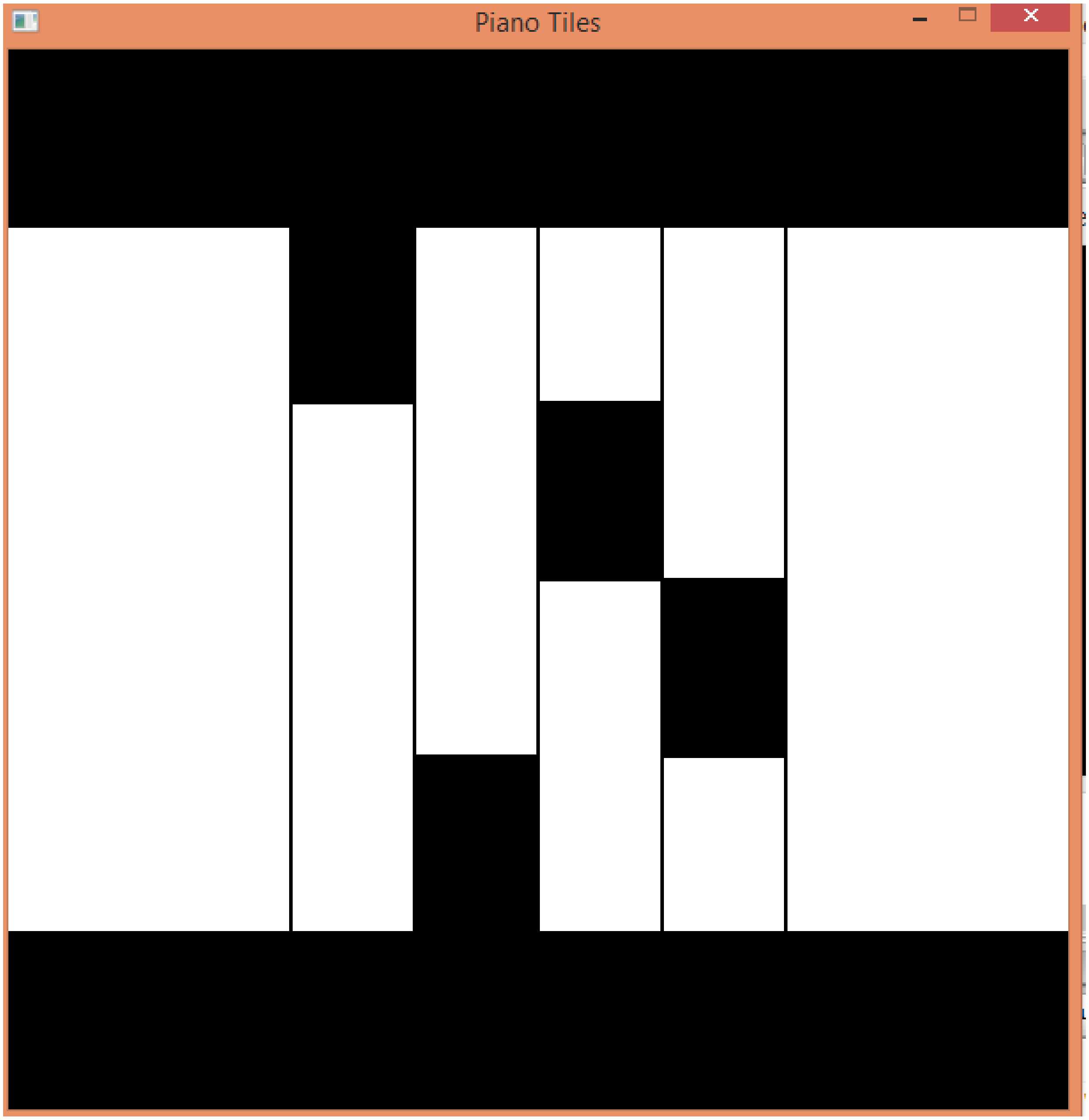
Instructions for classic mode. Clicking on anywhere will bring the user to start screen.

Mode displaying interface:



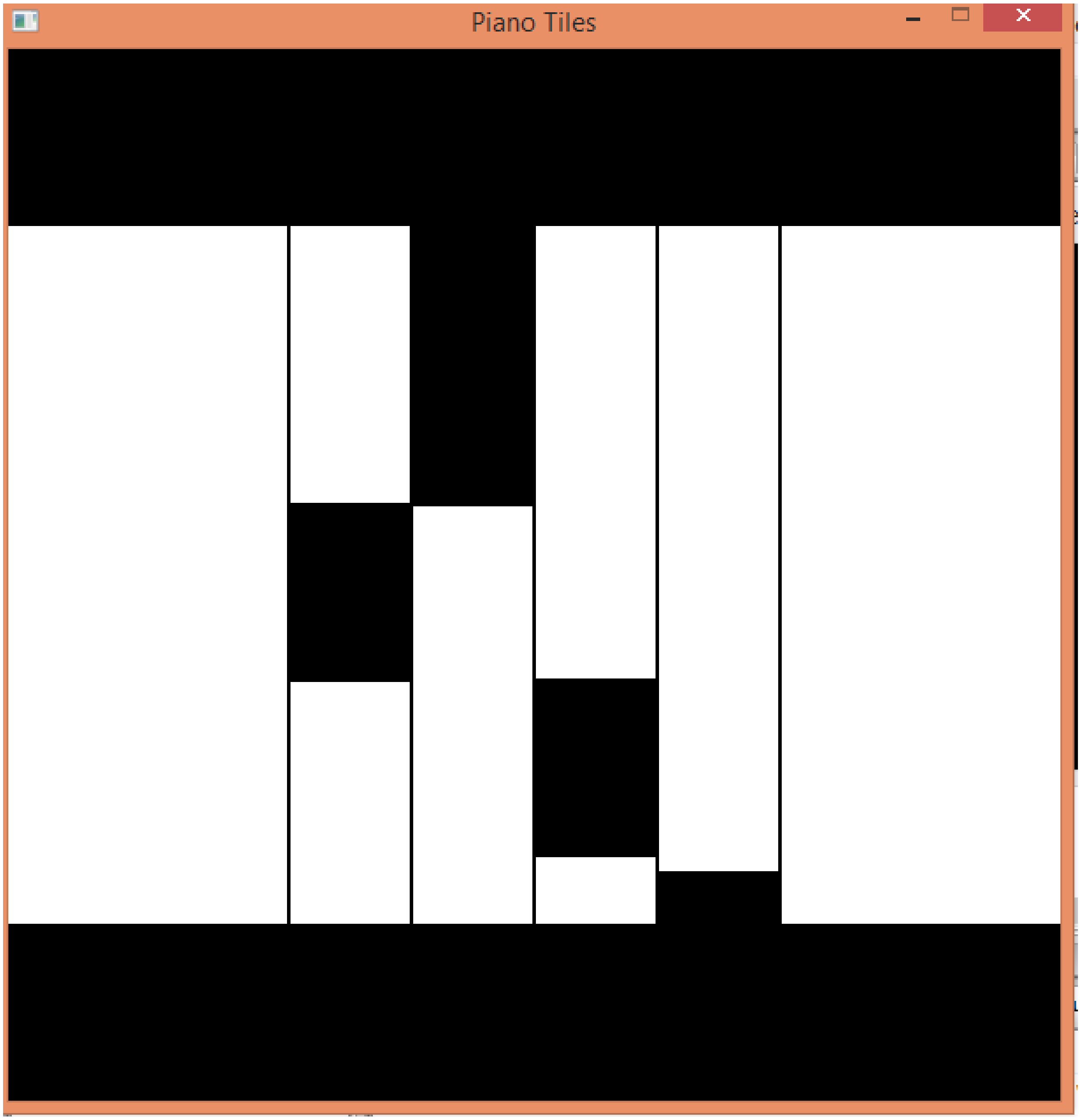
Different modes showing interface. Clicking on any of the modes will open the game.

The Game



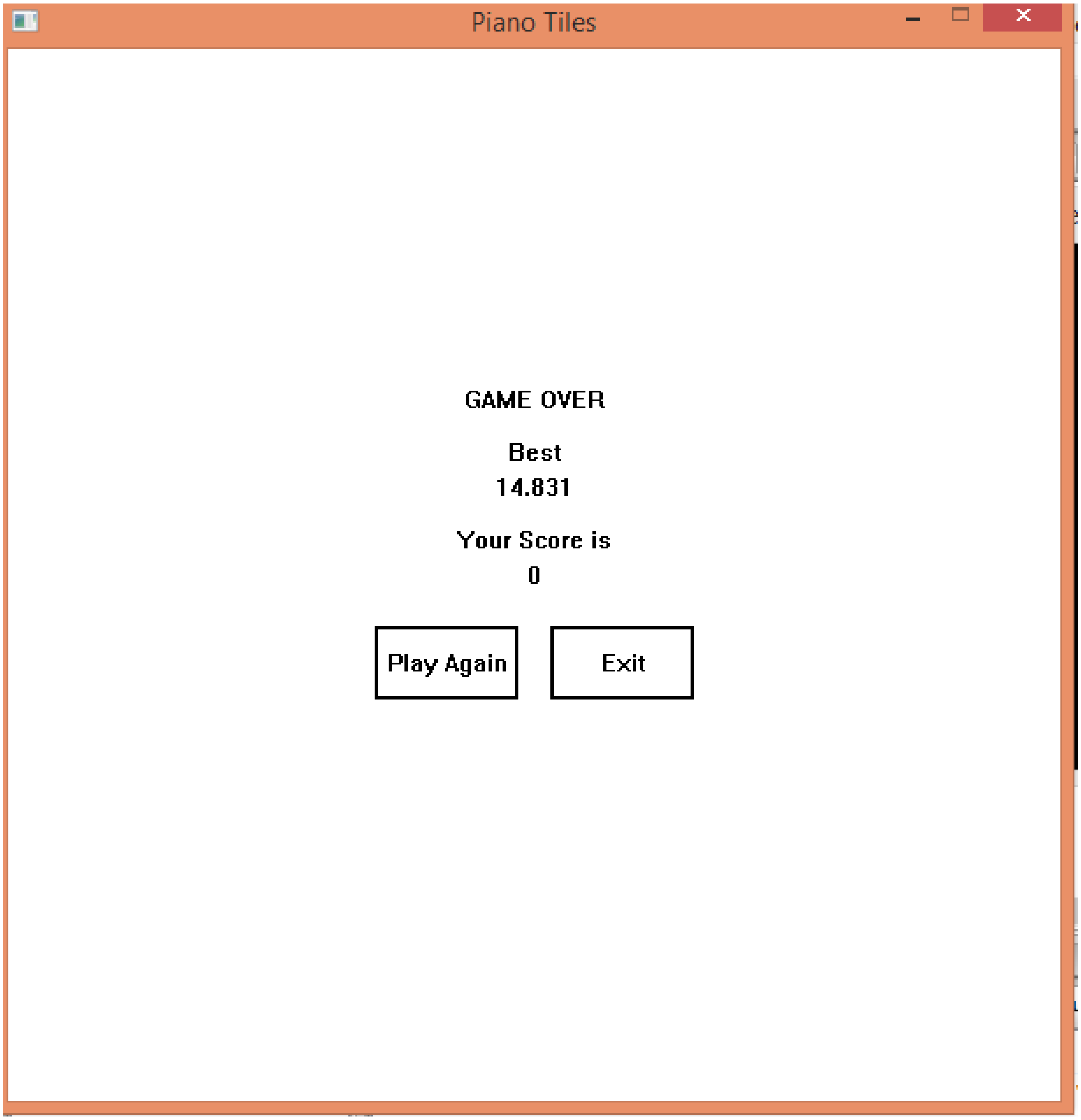
This screen appears if the user clicks on any of the modes. Tapping on the bottommost black tile will run the game.

Tiles Moving:



A screenshot to show that the tiles are in motion after clicking on a right tile.

Final Score Display:



```
GAME OVER

Best
14.831

Your Score is
0
```

After end of the game the user will see his final score and highscore of that mode. Click on Play Again to play the game again or click Exit.

Discussion:

1) What worked as plan?

Continuous motion of tiles after the click of user is successfully accomplished by us.

Random positions of all tiles worked as planned.

Calculation of time taken by the user during the game was updated successfully

Making of four different modes was successfully completed

2) What we added more than discussed in srs?

Display and update of highscore everytime the user plays any of the mode.

We included sound library and now everytime user click on a black tile, a beep sounds.

Instructions about the game has been added on the start page

3) Changes made in plan:

An additional arcade mode was thought to be added, in which the tiles moves continuously irrespective of the click of the user.
Reason: We were unable to do it in simple cpp

We thought to use openGL for our game.
Reason :There was not much need of it.

7. FUTUREWORK:

The game can be further improvised by making a better interface using other libraries like sfml.

Another mode i.e. arcade mode can be made in which tiles move continuously all the time with some specific speed increasing with time . Game ends if the user clicks on wrong tile or the last black tile reaches the end point of game window.

Further tiles can be made of different random colours.

8. Conclusions:

Piano Tiles is a perfect mix of simplicity and addictive gameplay that's easy to learn, but incredibly challenging to get high scores.
It has simple controls, a charming soundtrack, and incredibly addictive gameplay. Although simplecpp does not have enough graphics, we tried to make this game as user friendly as possible.

References:

1) For downloading the Code blocks novice version:
http://www.cse.iitb.ac.in/~ranade/simplecpp/

2) To find sound and time functions:
www.stackoverflow.com

3) Some additional informations:
www.wikipedia.com

4) To create other documents and format of code:
www.moodle.iitb.ac.in

5) All explaination of our project has been on
https://www.youtube.com/watch?v=BqhyPS3wJQw