TREASURE HUNT

NIRMIT ARORA (200010034)
ISHIKA SHARMA (200010020)
HRISHIKESH PABLE (200010037)
RAHUL PRAJAPAT(200010043)

JUNE 12

IIT Dharwad
Created by: Group 4

Problem Statement

The Gold Detector (Group 4)

(a) Consider a grid of 4x5.

(b) Use the random function to position the 3 bombs in the grid every time the game is started.

(c) Use the random function to position 1 box of gold coins in the grid every time the game is started.

(d) Use the scanf function to take player input.

(e) If the number entered by the user is equal to the number where the bomb is positioned then the player loses and the game ends.

(f) If the number entered by the user is equal to the number where the box of gold coins is positioned then the player wins and gets 100 points. A new game can be started with the player score retained.

(g) If the number entered by the user is not equal to the number where the bomb is positioned or the box of gold coins is positioned then the player must input a new number and gets 5points as a bomb is not encountered. After 5 chances the game must be restarted but the player score must be retained.

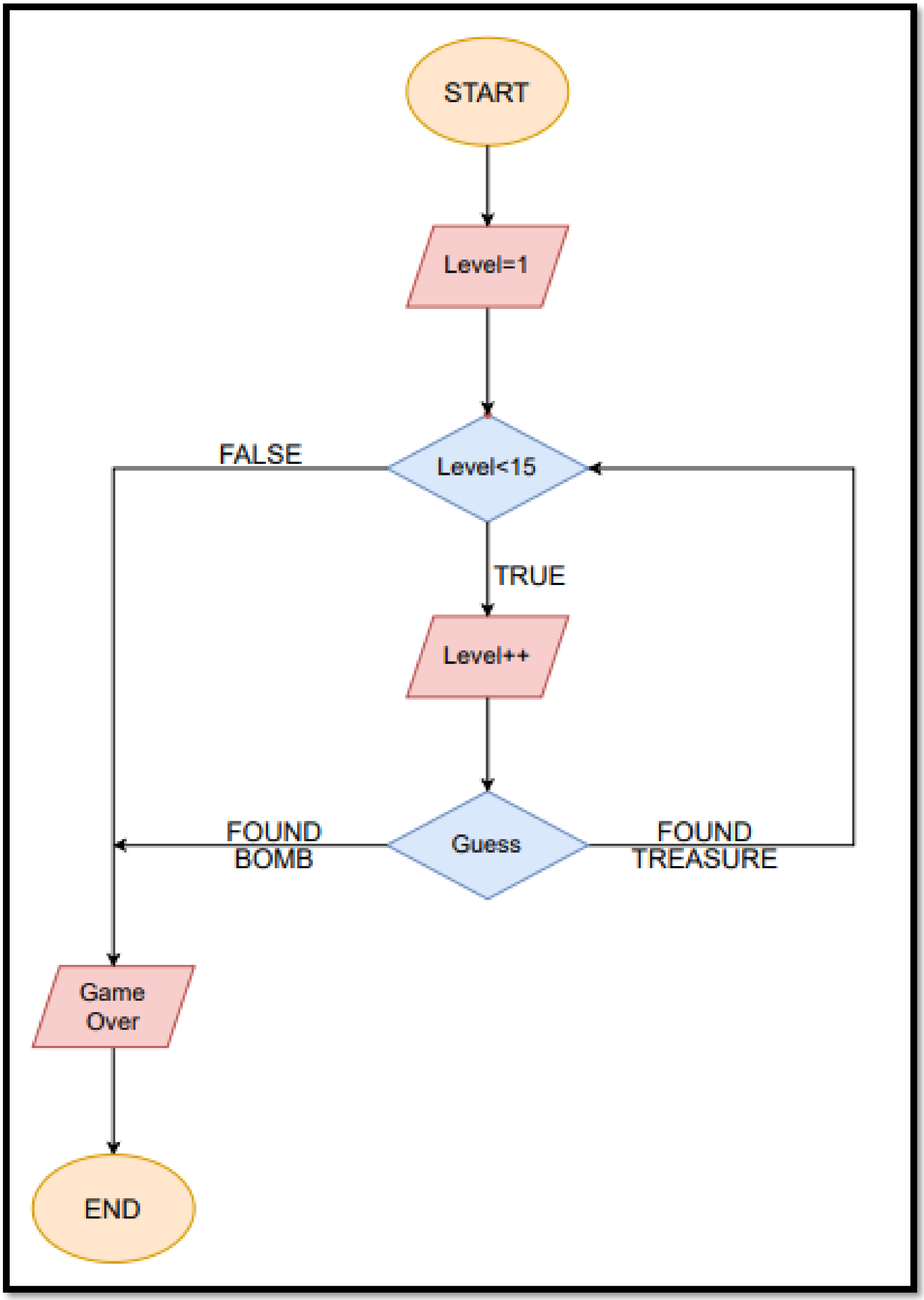
(h) When the player wishes to terminate the game, the final score must be displayed.

(i) The player's last 3 scores can be displayed.

(j) Single user game.

There are some new interesting developments too…

FLOWCHART


START
Level=1
Level<15
FALSE
TRUE
Level++
Guess
FOUND BOMB
FOUND TREASURE
Game Over
END

DESIGN STEPS

1. level Function: This function initiates the level of game while being called in the main function when the program starts and is also a gateway to the next level once the user finds treasure at some level.
2. Drawing game board and background on screen:
   - 2.1. mat Function: It makes the background design.
   - 2.2. grid Function: Creates the grid where the game will be played.
   - 2.3. border Function: It sets the border of the screen.
3. set_up_game_board Function: It places the bombs and treasure on the multidimensional array named game_board using the random function.
4. Input from user and then checking of bomb or treasure at the location:
   - 4.1. guess Function: It takes the input from user when the user executes a left click on the game board on screen. That input is then turned into the corresponding location in game_board array.
   - 4.2. check Function: It checks if the chosen location has a bomb, a treasure or if it is empty. According to each case a different parameter is passed to scoring Function.
5. scoring Function: If the location chosen by user is empty then it adds 5 points to score. If the location chosen by user has the treasure, then it adds the level number times 500 points to score.
6. After updating score:
   - 6.1. If a **BOMB** is encountered at the location selected by the user:
     - 6.1.1. boom Function: It reveals all the bombs that were there on the grid on the display screen.
     - 6.1.2. game_over Function: It puts up the game over screen which will lead to the high-score screen.
   - 6.2. If **TREASURE** is encountered at the location selected by the user:

6.2.1. coins Function: It displays the coins stacking up on the screen.

6.2.2. steps Function: It displays steps with an upward arrow indicating that now the game is going to level up!

6.2.3. treasure_found Function: It calls the coins and steps function along with writing “YOU WON” on screen.

7. Saving the score in a text file:

7.1. save_score Function: It creates a text file to store the scores if the user plays the game for the first time.

7.2. retrieve_score Function: It fetches the scores from the text file.

7.3. put_score Function: It stores the new score in the text file if it is one of the high scores.

ADDITIONAL FEATURES

1. treasure_sensor Function: It tells the direction in which the treasure is present on the grid board.
2. blaze Function: It reveals all the boxes on the grid which are diagonal to the previously chosen box. If there is bomb on the diagonals then it reveals them and if there is treasure on the diagonal then it is awarded to the user leveling up the game further if possible.
3. high_score_screen Function: It checks if the current score is the highest ever achieved by the user and puts a high score screen, which has the current score also.

ROLES OF TEAM MEMBERS

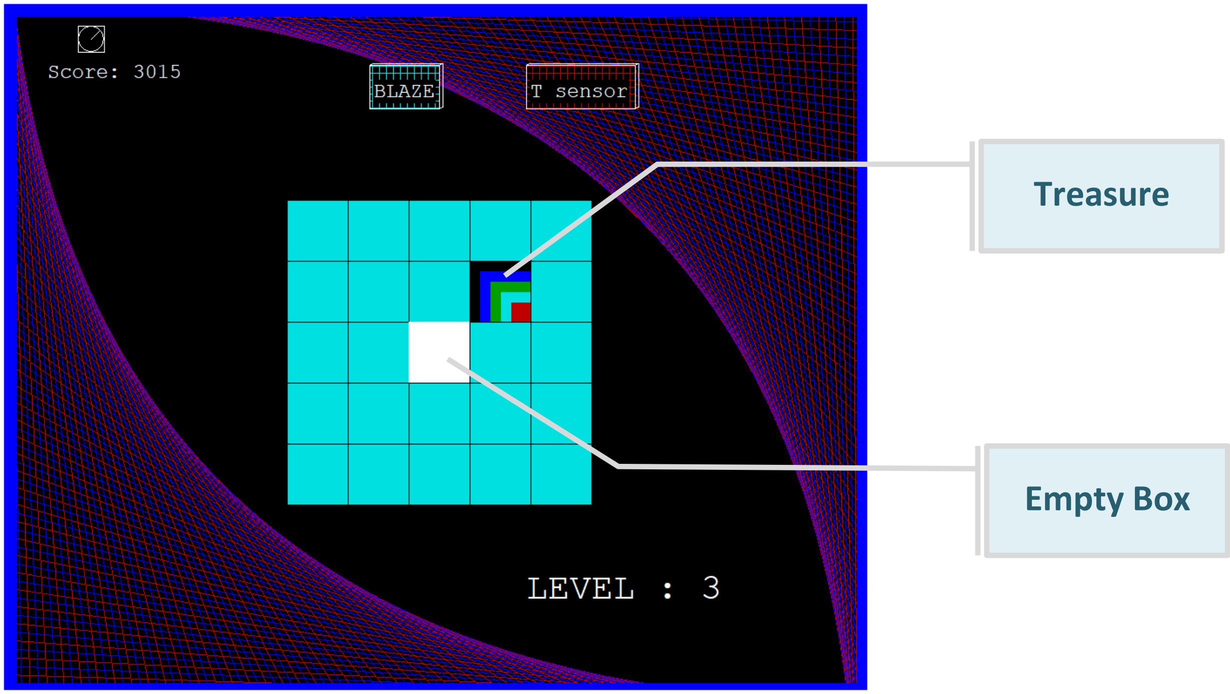
Student Name	ROLE
Ishika Sharma (200010020)	Main code layout, continuous evaluation, integrating, upgradation of code, functions designed and implemented: level, mat, grid, border, set_up_game_board, guess, check, boom, game_over, treasure_found, coins, steps, scoring.
Nirmit Arora (200010034)	Final report writing, functions designed and implemented: retrieve_score, put_score, save_score, H_Scoremat, new_high_score, high_score_screen.
Hrishikesh Pable (200010037)	Report 1 and report 2 writing, <graphics.h> setup, functions designed and implemented: treasure_sensor, Blaze, blaze_check.
Rahul Prajapat (200010043)	Read me file, report formatting, flow chart making, designed and implemented mat function.

REFERENCES

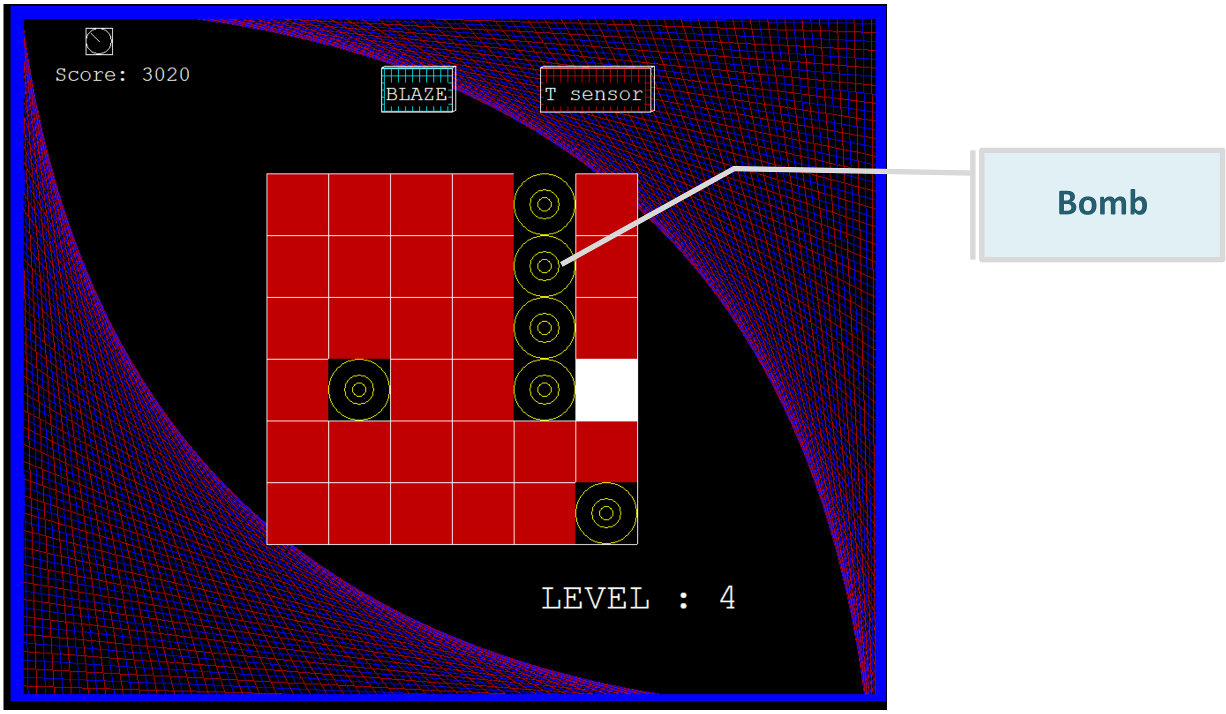
- VCR Tutorials: For <graphics.h> and mouse pointer for taking the input. [https://youtube.com/playlist?list=PL5UFsTza4wWSNhe0xuO6ELw7ORU-UHNDO]
- Source code of the Minesweeper game. [https://stackoverflow.com/questions/50320998/minesweeper-in-c/50321042]
- Stack Overflow: [https://stackoverflow.com/questions/143174/how-do-i-get-the-directory-that-a-program-is-running-from]
  - different functions such as save_score.
  - Regarding other warnings and error.
- Geeks for Geeks:
  - for saving the file in the current directory.
  - for errors regarding <graphics.h>. [https://www.geeksforgeeks.org/setfillstyle-floodfill-c/]

RESULTS:

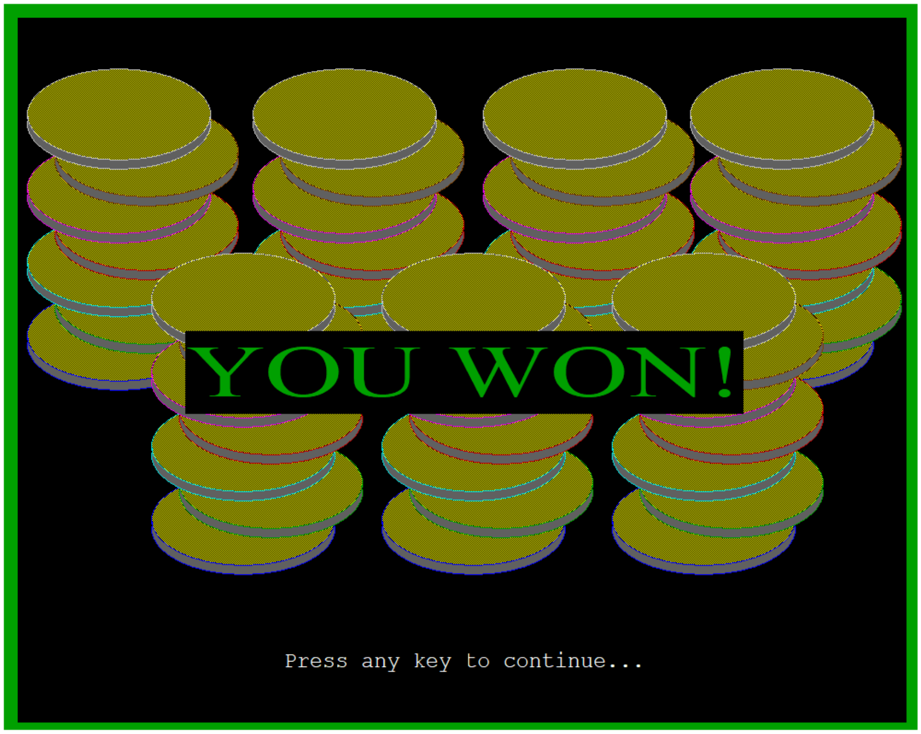
- The grid when user wins a level:



- The grid when user loses a level:



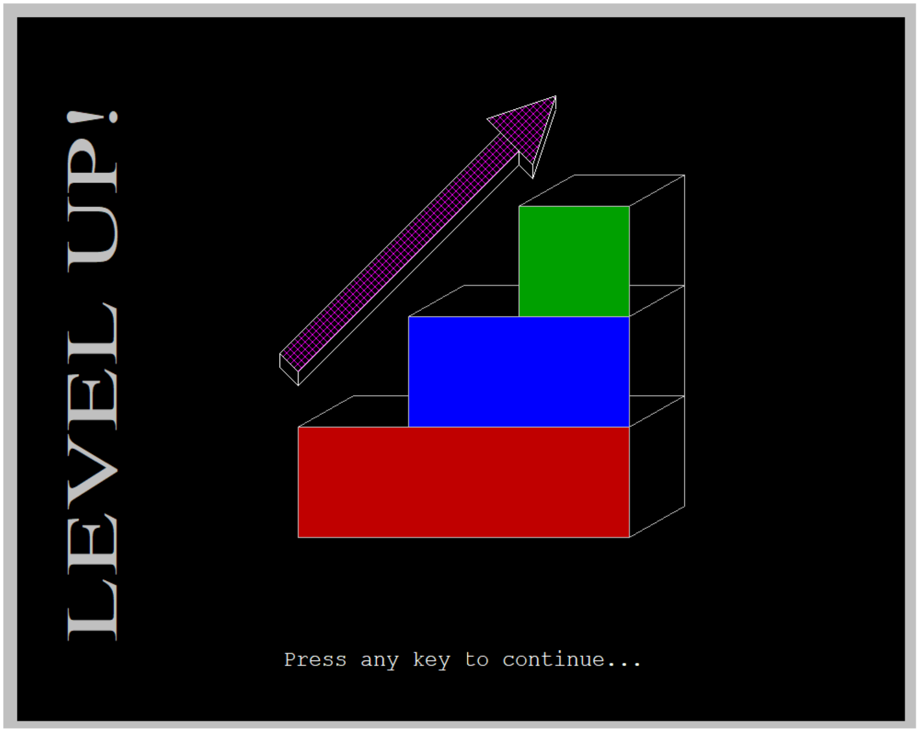
- “YOU WON” screen:



YOU WON!

Press any key to continue...

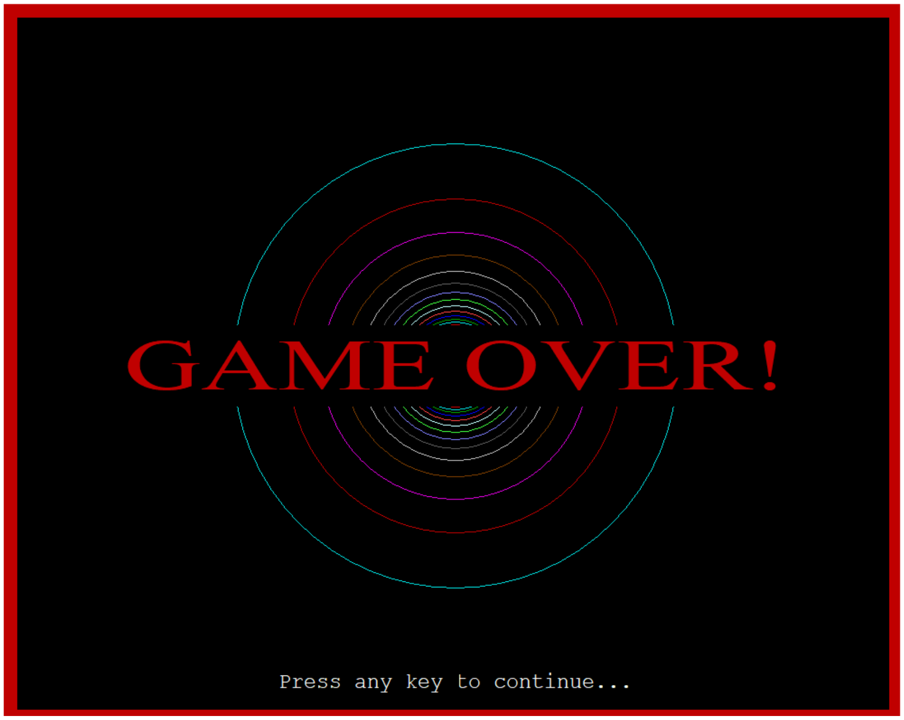
- “Level Up” screen:



LEVEL UP!

Press any key to continue...

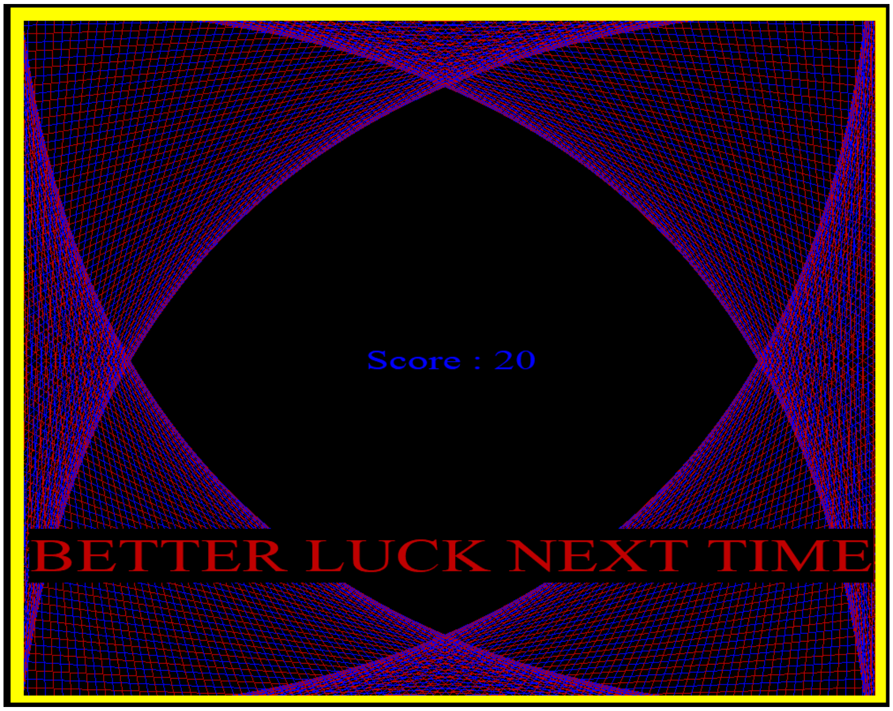
- Game over screen:



GAME OVER!

```
Press any key to continue...
```

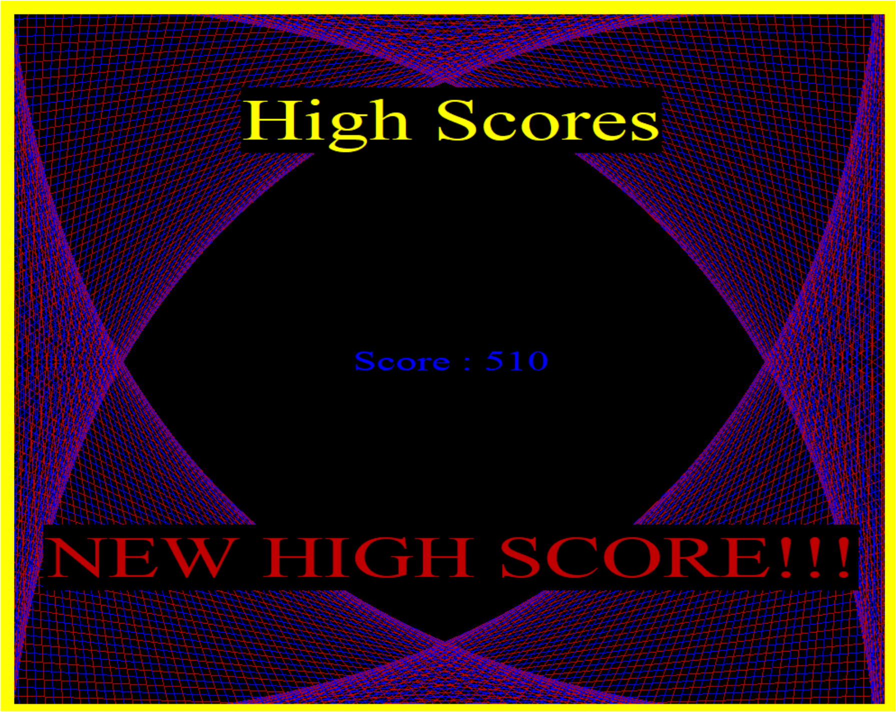
- Screen for score display:



Score : 20

BETTER LUCK NEXT TIME

- Screen for new high score:



High Scores

Score : 510

NEW HIGH SCORE!!!