MERGE BLOCKS GAME

Aditya Hegde • 201IT105

CONTENTS

INTRODUCTION

- In recent years, the computer games have made remarkable achievements, especially it has really attracted and seeked attention from kids and youths.
- Logical Based Puzzles are becoming more and more popular these days. These games improve logical ability and Thinking power. Instead of wasting time on games like PUBG ,these kind of logical games have to be played.

PROBLEM DESCRIPTION

- In a fixed size container, blocks of different images are dropped from above. Blocks of same images merge together to form a bigger number block.
- This same process continuous until the container becomes filled with the blocks and there is no place for new block to enter.
- The final score is the overall sum of all the merged blocks.

Novelty

- It’s an advanced version of famous game 2048.
- We will use images while merging the blocks but score will be calculated on the basis of value or rank of that image.
- We have made 3 difficulty modes :- Easy, Medium, Hard.
- Adding Leaderboard on homepage.
- Added background music while playing.
- Added new animations and backgrounds.

METHODOLOGY

- ***FRONT-END :-***

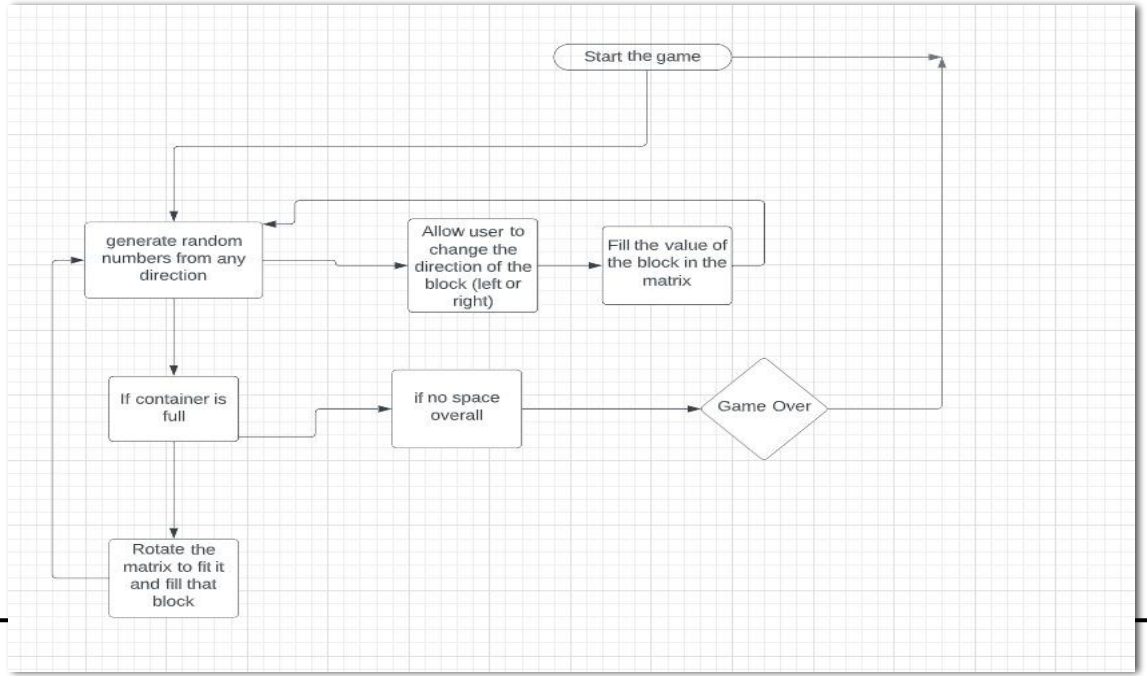
1. We will be creating three UI screens - **Home Screen**, **Play Screen and Leaderboard Screen.**
2. The Home Screen consists of buttons Start Game and Leaderboard.
3. The Play Screen has a board with blocks dropping into the board from top frequently. Also the score is displayed at the top. Best Score will also get updated.
4. The Play Screen will have buttons to new game, select difficulty and undo moves.

- ***BACK-END :-***

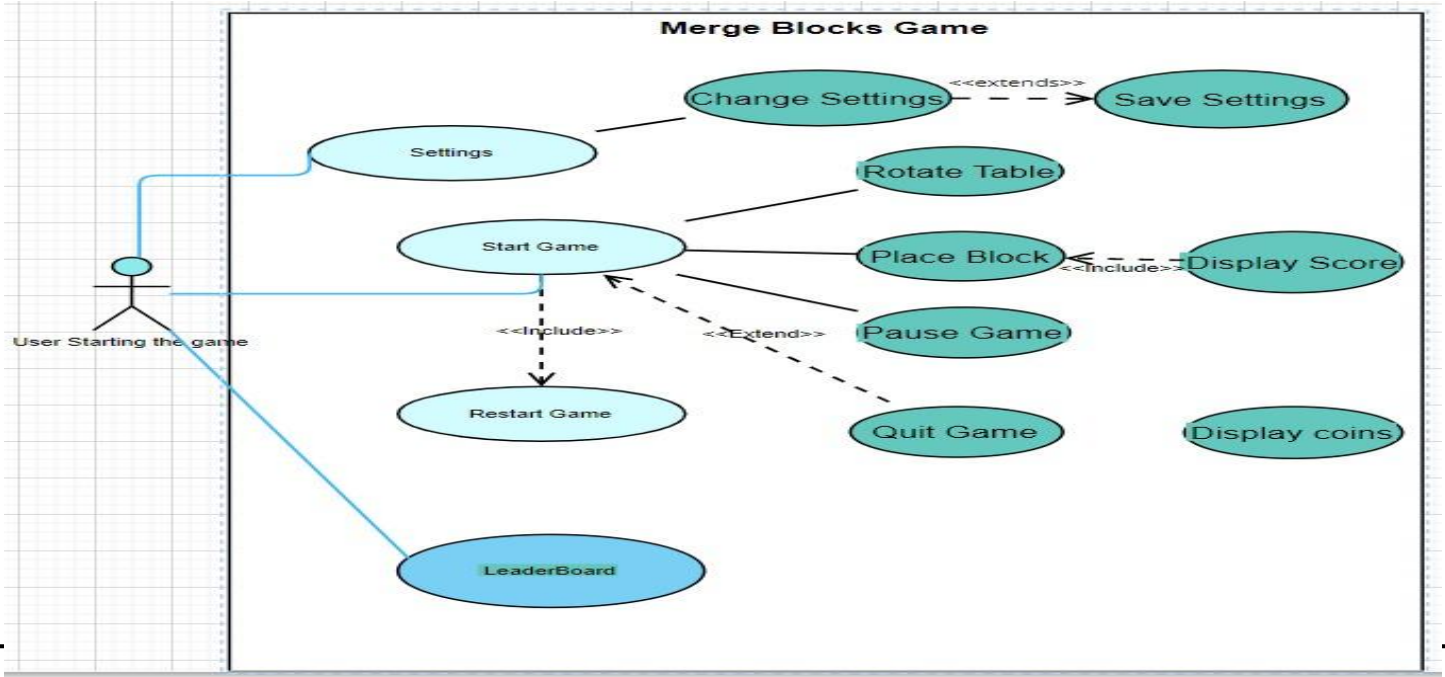
1. In the back-end, we will have random generation of values that appear as blocks in front-end.
2. As soon as the dropping block touches one or more blocks, a method is triggered to check for possibility of any merges. If merges are possible , they are reflected on the screen.
3. The state of the board is changed after each block drops.
4. The score and best score is updated after each drop.
5. User score is stored in a universal database that is displayed in the leaderboard.

UML DIAGRAMS

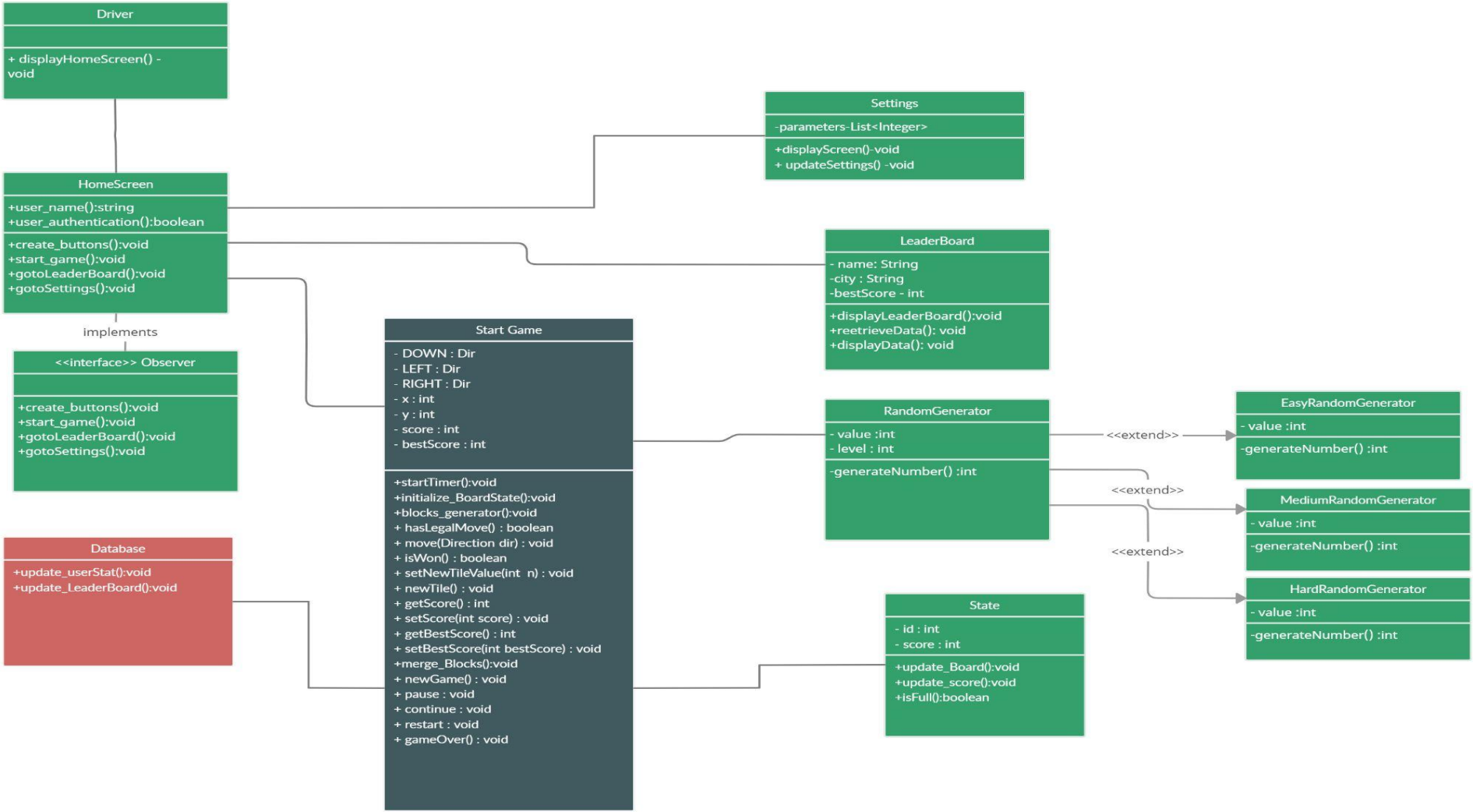
Activity Diagram :-



Use-Case Diagram



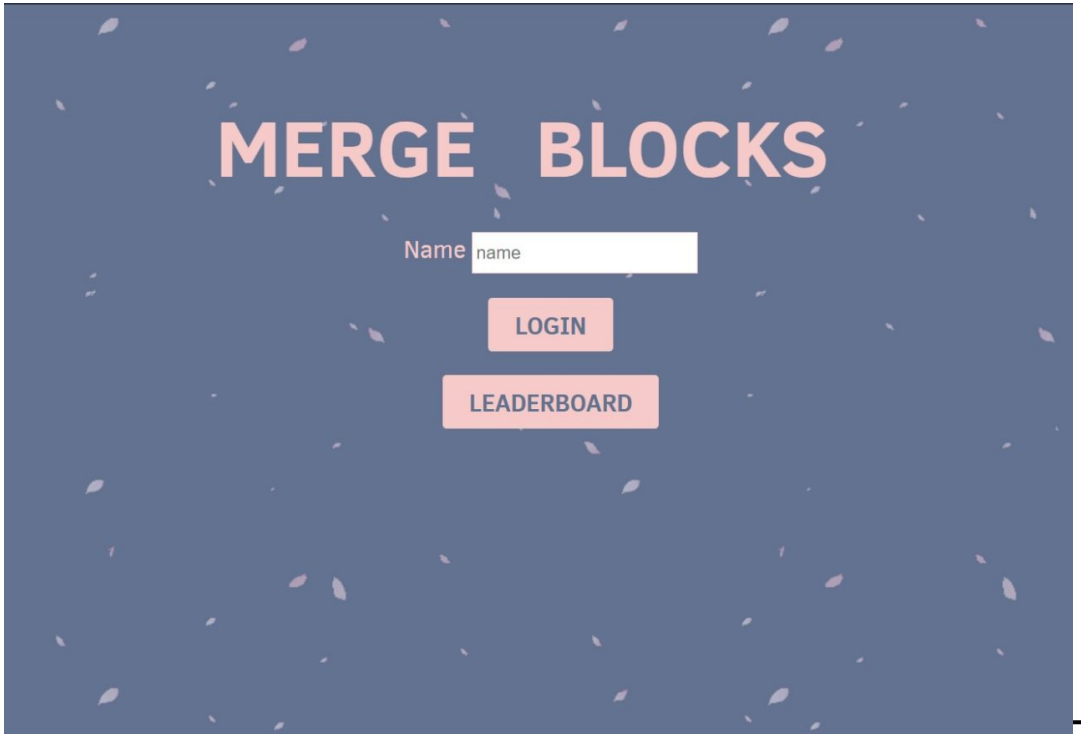
Class Diagram


Driver
+ displayHomeScreen() -
void
Settings
-parameters-List<Integer>
+displayScreen()-void
+ updateSettings() -void
HomeScreen
+user_name():string
+user_authentication():boolean
+create_buttons():void
+start_game():void
+gotoLeaderBoard():void
+gotoSettings():void
LeaderBoard
- name: String
-city : String
-bestScore - int
+displayLeaderBoard():void
+reetrieveData(): void
+displayData(): void
implements
<<interface>> Observer
+create_buttons():void
+start_game():void
+gotoLeaderBoard():void
+gotoSettings():void
Start Game
- DOWN : Dir
- LEFT : Dir
- RIGHT : Dir
- x : int
- y : int
- score : int
- bestScore : int
+startTimer():void
+initialize_BoardState():void
+blocks_generator():void
+ hasLegalMove() : boolean
+ move(Direction dir) : void
+ isWon() : boolean
+ setNewTileValue(int n) : void
+ newTile() : void
+ getScore() : int
+ setScore(int score) : void
+ getBestScore() : int
+ setBestScore(int bestScore) : void
+merge_Blocks():void
+ newGame() : void
+ pause : void
+ continue : void
+ restart : void
+ gameOver() : void
RandomGenerator
- value :int
- level : int
-generateNumber() :int
<<extend>>
<<extend>>
<<extend>>
EasyRandomGenerator
- value :int
-generateNumber() :int
MediumRandomGenerator
- value :int
-generateNumber() :int
HardRandomGenerator
- value :int
-generateNumber() :int
Database
+update_userStat():void
+update_LeaderBoard():void
State
- id : int
- score : int
+update_Board():void
+update_score():void
+isFull():boolean

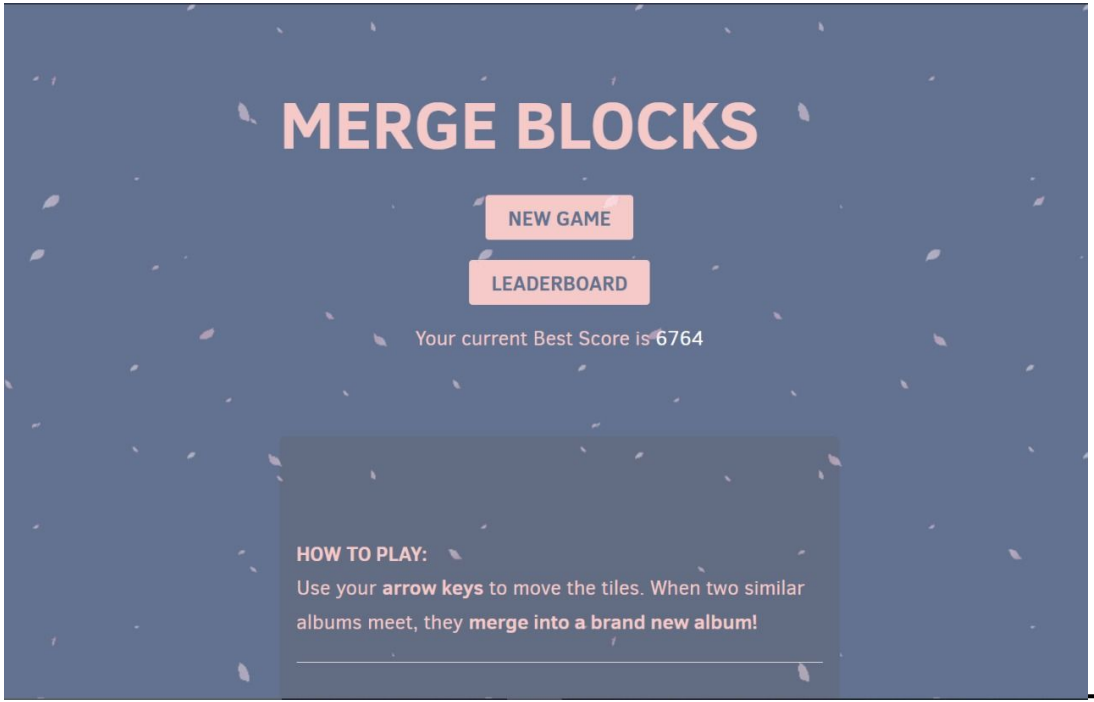
Results

1. After designing the game, it was played many times to check the correctness and efficiency of the project.
2. It can be declared that the code is error-free after a large number of trials. Few screenshots of the website are attached for reference Leaderboards and settings pages were working as expected , with music being played during the game.

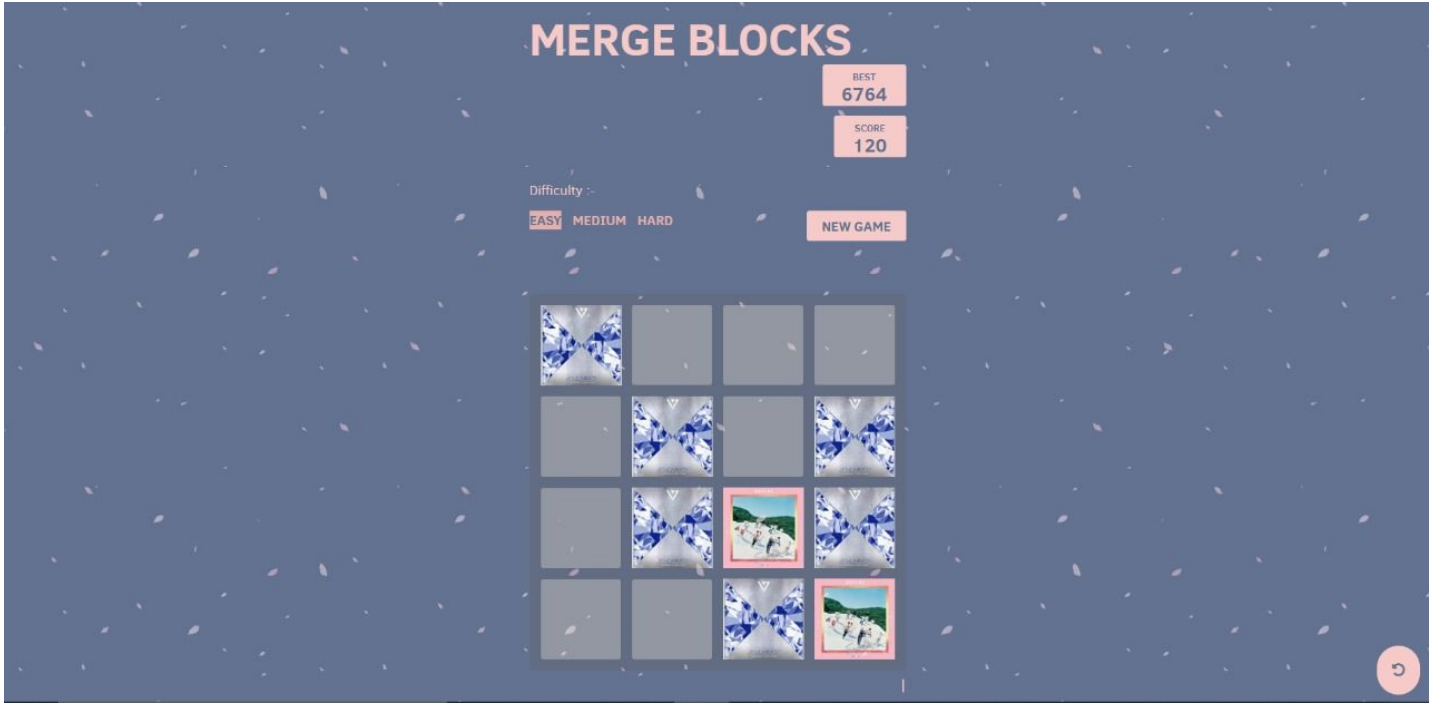
Home Page



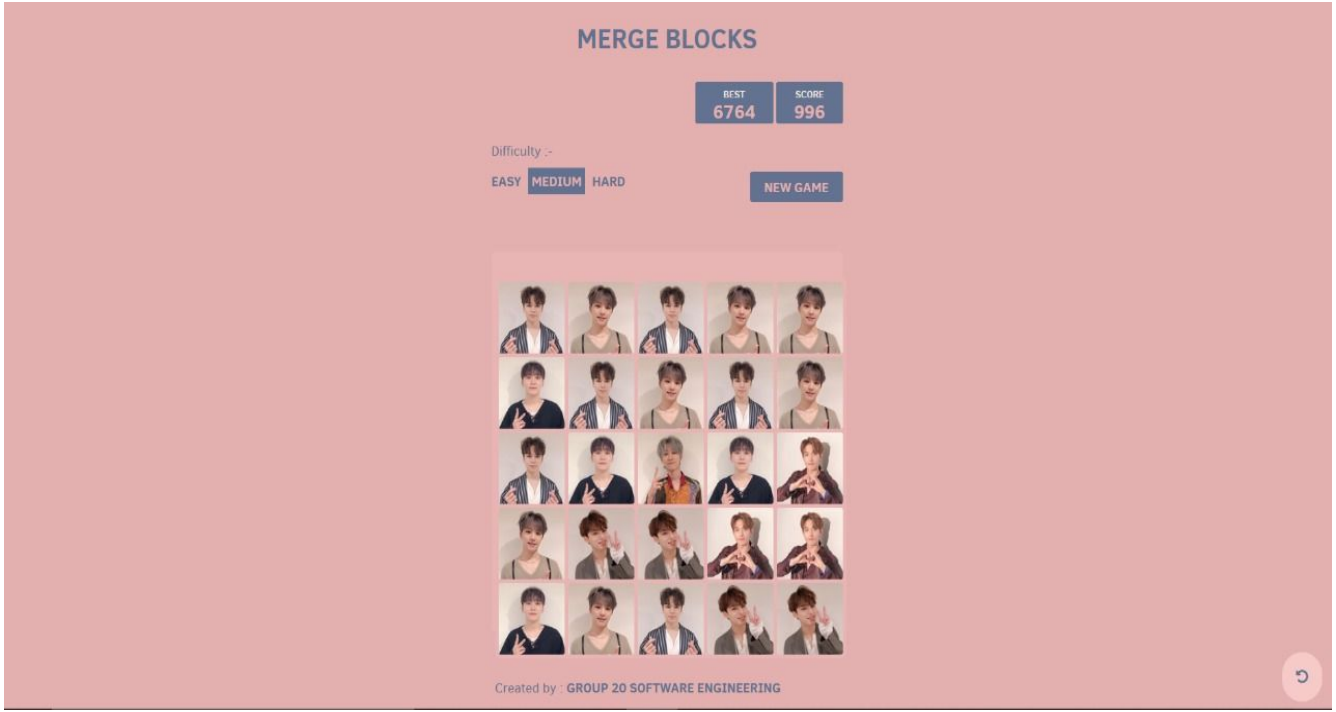
Dashboard Page



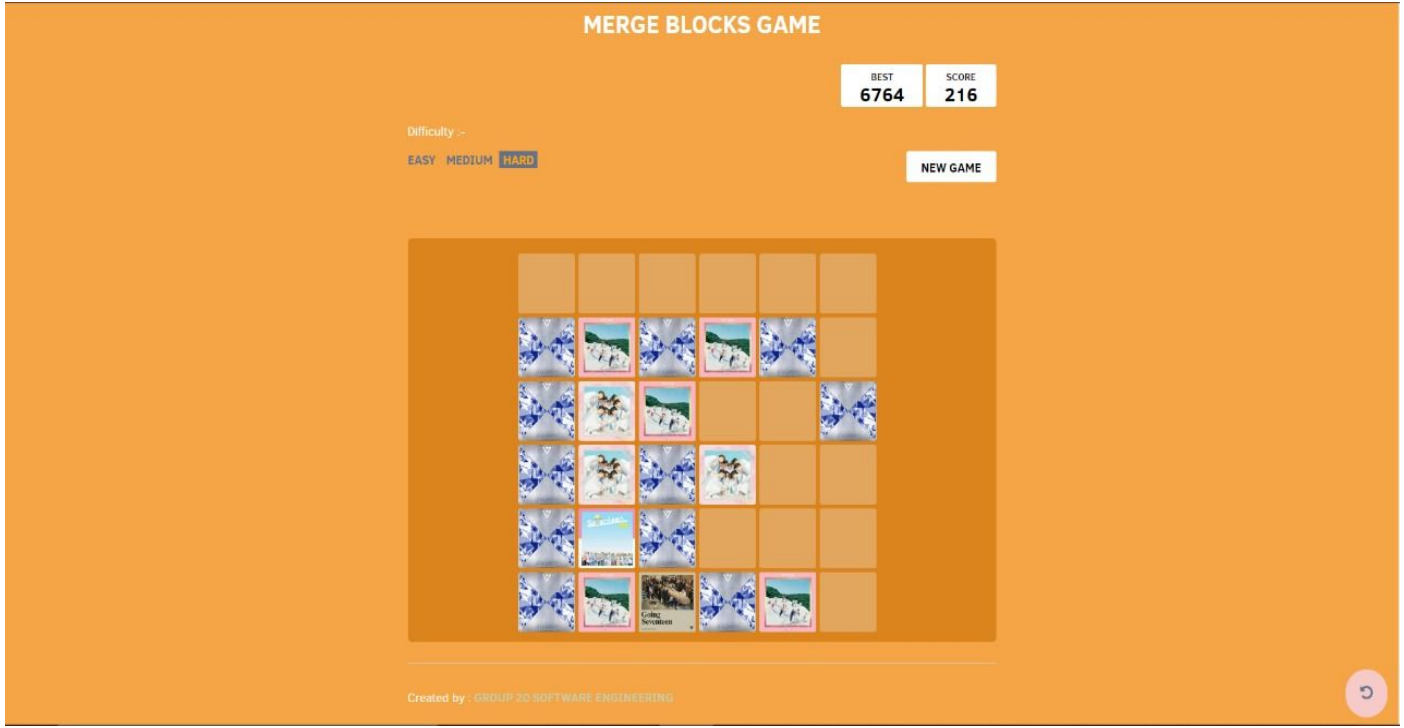
Easy Game Page



Medium Game Page



Hard Game Page



Game Over Page

Conclusions

1. Project was made an advanced version of 2048 game based on merging of blocks. This game will be played on a website .Game was added with some interesting features such as leaderboard of the users ,best score of the user ,difficulty level of the game i.e. easy,medium ,hard.
2. To make the Game to more interesting we used the pictures in frontend for calculating the score rather than the same number but numbers will be used for calculating the score in the backend. This game also contains the feature of playing music.
3. This Game gets a better user experience if it is developed for android and can be available at different platforms such as google play store so that maximum people can enjoy the Game.

REFERENCES

- use case diagram for 2048 game - Bing images
- Software Engineering-A Practitioner's Approach, *Pressman*
- 10 Use Case Diagram Examples (and How to Create Them) - Venngage

THANK YOU