**Assignment 2**

**Group #14**

**BSCS14044 – M. Abdullah Sajjad**

**BSCS14021 – Anum Rafi**

**BSCS13025 – Muhammad Athar Siddiqui**

**Requirements Engineering**

**Introduction:**

Pair Parallax is a Puzzle type single and multiplayer game. The game revolves around finding matching cards which have been flipped over. Each player gets 2 turns. If they can flip 2 cards which match, they get an extra pair of turns; else the other player gets to take 2 turns and so on until all cards have been flipped. The player with the highest score wins. The game can also be played alone which involves a simple timer where the player has to solve the puzzle in the least time possible. The project is intended to be developed on Unity 3D following proper Software Engineering Techniques which we will study in class.

**Requirements elicitation:**

1. Scenarios
2. Use Cases

**Reasons:** Ours is a game which we have decided to develop. Therefore interviews are not going to be beneficial. We can also not use Ethnography as we are developing a game we came up ourselves. The options left are Scenarios and Use Cases. We can work with scenarios as a game can have multiple states and that can be effectively portrayed by scenarios. For example what can player 1 do while it’s his turn and what he can do when it’s not etc.

**Lenses:**

**Lens of EMOTION:**

We would like to create emotion of self-doubt. In our players. We want their heart beats to rush and feel excited throughout the gameplay. Thrill is what keeps bringing back players.

Since the game has not been created yet, we do not know what emotions it currently gives. However we have played the game in real life with 2 sets of playing cards and it is fun to say the least.

**The Lens of Essential Experience**

We want the player to experience thrill and excitement. The pressure of the ticking time and the desire to win. We simply want the experience to be intense for the players. Our game will capture this essence by limiting time for each turn of the player, allowing cheating in exchange for something valuable and making them believe whatever they are betting is actually valuable.

**The Lens of Fun**

Fun is an essential part of the game. We will allow this by allowing players to cheat and deceive other players by momentarily viewing all hidden cards in exchange for some points. The game is a test of memory and knowing you are better intellectually is always a good boost to the ego directly proportional to fun.

**The Lens of Curiosity**

Our game revolves around this lens. Our game is based on hiding cards hence not knowing where the desired card is going to add an element of curiosity into our players.

**The Lens of Endogenous Value**

We plan to add a class system allowing players to achieve levels and unlock a status. Higher status means more power in the game but we intent to make that position volatile so as to not give unfair advantage to a selected few.

**The Lens of Problem Solving**

As this is a puzzle game, the answer here is self-explanatory

**Functional and non-functional requirements**

**Functional requirements:**

From system point of view our requirements are to maintain the flow of game that it must not break, server where all data will save is firebase database of android studio. Functional system requirements are OS 4.4 kit Kat, processor must be 1GHz and RAM must be 512. Functional user requirements have smart phone with internet connectivity and minimum 4.4 android.

**Non-Functional requirements:**

The game will run for 10 seconds for each turn and every player get two turns and if the player succeeds he/she will get two more turns. For every turn system will take at least 6-byte rate per transfer, game graphics will be stored. IDE will be unity and java script, efficiency and complexity will be maintained by no lagging in the game, internet connectivity and space and performance requirements.

**Types of non-functional requirements:**

Three main types of non-functional requirements are covered in our game are as follows:

* Product requirements
* Organizational requirements:
* External requirements

**Explanation of type with their sub types are written below**

* **Product requirements:**

-Efficiency requirements: handle by no lagging in the game and continues internet availability. Space and complexity also handle using at least 6 bytes’ rate per transfer and 10 seconds for each turn.

-Usability requirements: handle with different screen sizes

-Security requirements: we don’t do anything with security neither secure nor invade it.

-Dependability requirements: dependency handle with supporting OS 4.4 kit Kat and processor of 1GHz with 512 RAM.

* **Organizational requirements**

-Environmental requirements: randomly cards are placed in a grid that shows to the player for a while.

-Operational requirements: Player have to match the same and get points in return if he\she succeeds he got another two turns and so on. There is menu bar too in the game.

-Developments requirements: unity with OS supported and android studio. Person will login first automatically.

* **External requirements**

-Regulatory requirements: rules are that every player must get two turns and if the players plays correctly, get points he/she get extra two turns and plays again.

-Ethical requirements: we will not take personal information of the player and game will run dynamically and fairly. The ethical way of playing is not no use gems or hint option but if anyone has too much eager to win the game in any case for them there will be options.

* **Legislative requirement**

-Accounting requirements: there will be no purchasing method in the game

-Safety/security requirements: there is no way of either securing the security of player or invading security of any player.

**Requirements Validation**

* **Consistency check**:

According to the above stated requirements we don’t see any conflicts in any of them effecting any other requirements. The functional requirements are pretty basic as you can see that minimum OS, Processor and Ram requirements and very low. Smartphones these days have specs far better than the minimum requirements for this game.

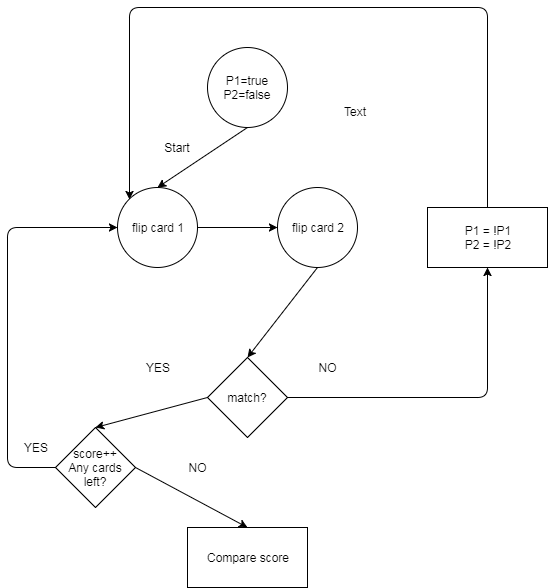
* **Realism Check:**

Yes, the app can be realistically built and all of the above stated requirements can be implemented using the current technology. To be exact we’ll be using Android Studio for the development of the app, Firebase for the real-time database, authentication, hosting etc. and Unity Game Engine for designing the game.

* **Verifiability Check:**

We won’t be using requirements review as our requirements definition has almost been formulated and all of requirements are well defined, realistically testable and understandable. We’ll definitely be using the prototyping technique to understand the flow of the game and for the comparison of how it is supposed to work and how it is working in the prototype state. Also, we’ll be making test cases in order to test if all of requirements have been fulfilled and working properly. Also to check if there are any scenarios where the game could crash or halt and cause any problem to the device.

**DIAGRAM (SCENARIO)**

****