

# PROJECT REPORT

## MADE BY:

Sukhmanjit Singh Mann (180050106)

Shubham Gupta (180050101)

Dhananjay Singh (180050029)

## TITLE: QUERENCIA

Submitted-to: Prof. Amitabha Sanyal

CS-154

### PROBLEM-DESCRIPTION AND PROGRAM DESIGN:

**Querencia** is a 'multipurpose' software designed in Drracket which has enjoyable and intellectual games and which enables a user to connect to a LAN/WiFi based network and play a game with another user (who is online) as well as engage in real time interaction (like chatting or sharing data) with the user. This is possible only if the host is online, otherwise the user can play any of the two games on the software with the machine. This software aims at extending the limits of Racket as a language and has done so to some extent. The inspiration of this software is a lack of a platform in IIT-Bombay to share data and messages on the network.

We have given the software a very user friendly GUI and which gives the user two options on running the software:

- 1) Stay offline and play **WORD-SEARCH** or **MANCALA** with the machine.
- 2) Go online after logging-in with a username and password if the user is already registered in the host server, or register to the host servers.



### PART 1: ONLINE REAL TIME INTERACTION

Any number of users can connect with the host and engage in chatting or playing a game called Mancala with the other user. We also maintain a chat-history of two particular users within the user machines and the host maintains a database which contains the registered

users. There is also a part of our code which learns from the chatting-history and the dictionary of the user and gives **possible completed words** while the user is writing. So the machine learns as much as the user chats. We have also provided the feature of **sharing data** (images, files, videos, etc...) along with the messages. The host server is a separate identity from the user-world so the person who is hosting the server can also log in as a user. Any message the user sends is encrypted using a randomly generated permutation of the alphabet known only to the user and the 'partner'. We have used **Secured Hashing Algorithm (SHA-256)** for **password encryption** and **data integrity check**. The user can change his/her password, status, profile-pic or details and can also view another person's account. The function **switch** mainly handles the networking part of the program and has receive-message-handlers giving it instructions.



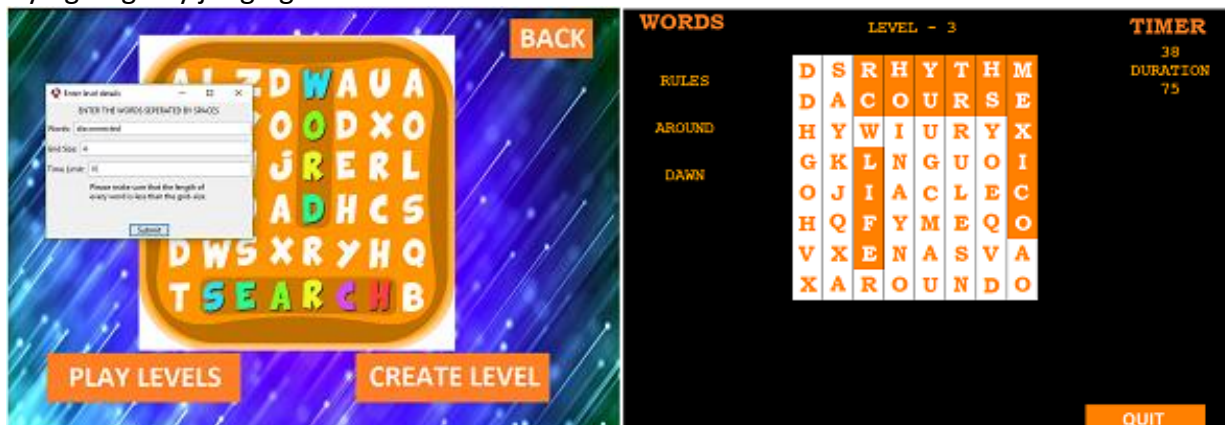
## PART 2: WORD-SEARCH

Word Search is a game that has a grid and various words hidden inside it. The main frame of the game displays a list of words, main-grid (playing-area) and a timer. There are 10 in-built levels and the letters in the word appear in the order somewhere in the grid, and are automatically placed horizontally, vertically or diagonally. The main point of the game is to find all the words hidden inside the grid in the limited period of time.

The game is played Offline and it has two parts:

- Play the in-built levels
- Create your own levels

This game uses an **our-own-algorithm** to generate a random grid having random combinations of letters of words from a given word list and also interlinks the common letters in the words provided. It also has a user-friendly interface. It even attempts to guess which way is the user trying to go by judging the mouse movements.



### PART 3: MANCALA

Mancala is one of the oldest known games to still be widely played today. The objective is usually to capture all or some set of the opponent's pieces.

The game has **two variants** with **two modes** each:

- Offline with AI
- Online 2-Player

The first variant involves a continuous streak of chances where the chance is passed when you land on an empty hole. In the second variant the turn is switched on alternatively. An extra chance is gained when you land on your home. This game uses **Minimax algorithm** and **Alpha-beta pruning**. The game has a user-friendly and customizable environment.

For further details about mancala, refer to this link:

Variant 1: <https://www.youtube.com/watch?v=-A-djjimCcM>

Variant 2: <https://www.youtube.com/watch?v=iSJk6CYsf6c>

The higher-order-function **minimax** gives the best possible move of **both** the variants using the above said algorithms and the current difficulty level which corresponds to the depth of searching. At a certain depth if AI is not able to make decision it will return a random move so that same game is not repeated. The function **turn** modifies the grid according to the move in an animated style.



We have used Microsoft Windows-10 Paint and Paint3D to create buttons and backgrounds and some images have been taken from Google images. A single higher-order-function called **user-update** has 3 slight-modifications in the 3 parts and handles the button-updates. The **button-area** is the main 'object-type' function in the GUI part, which could also be treated as a class but has been used as a lambda and we have made a **timer** using **timer%** class.

The **limitations** of this software are that it does not support group chats and the data-transfer speed is slow (0.7 to 0.8 Mbps). Sometimes it takes a bit of time for the formation of new-levels in word-search.