A Project Report On

SCRABBLE GAME

Submitted by

Divija Nagaraju – 14IT112

Mukta Kulkarni – 14IT220

Pooja Soundalgekar – 14IT230

Yogitha A N – 14IT251

IV SEM B.Tech (IT)

in partial fulfillment for the award of the degree of

Bachelor of Technology
In

Information Technology
At

Department of Information Technology

National Institute of Technology Karnataka, Surathkal.

February 2016

Abstract

Scrabble is a word game in which players score points by placing tiles, each bearing a single letter, onto a gameboard which is divided into a grid of squares. The tiles must form words which, in crossword fashion, flow left to right in rows or downwards in columns. The words must be defined in a standard dictionary.

Traditionally,the game is multiplayer where two or more players compete to extend the crossword such that they earn the maximum amount of points. This project instead presents uniplayer version of scrabble crossword game wherein just one physical player plays and the opponent is the computer itself.

The human player extends to the crossword by placing letters from the available tiles onto the board. For the computer player to perform, an algorithm for artificial intelligence is implemented because of which the it predicts the vacant grids on the crossword board,finds the highest scoring word that can be formed from the letters available and places them on the crossword grid.The words placed by each of them are validated from the dictionary file for their correctness.

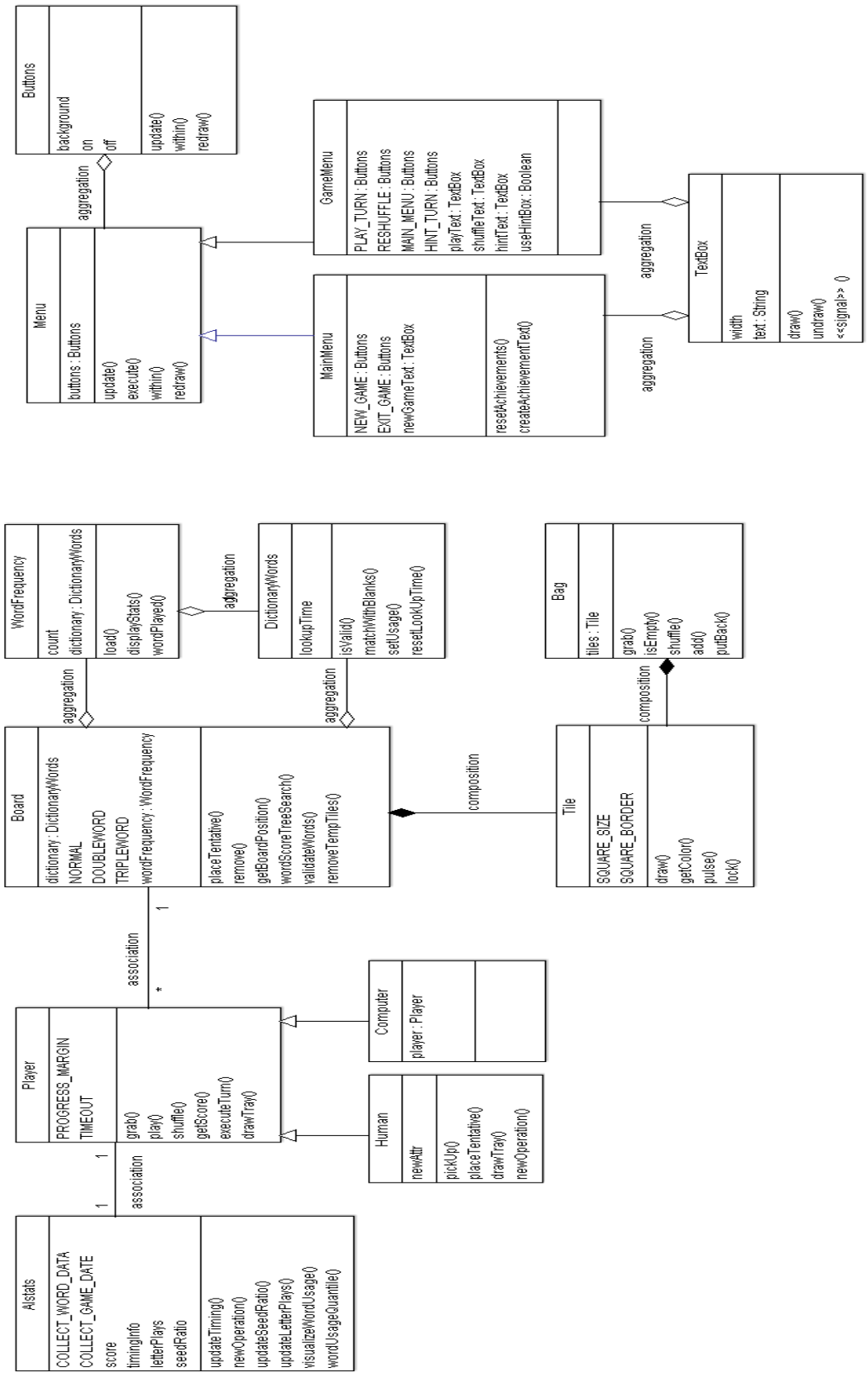
Table of Contents

1. Introduction

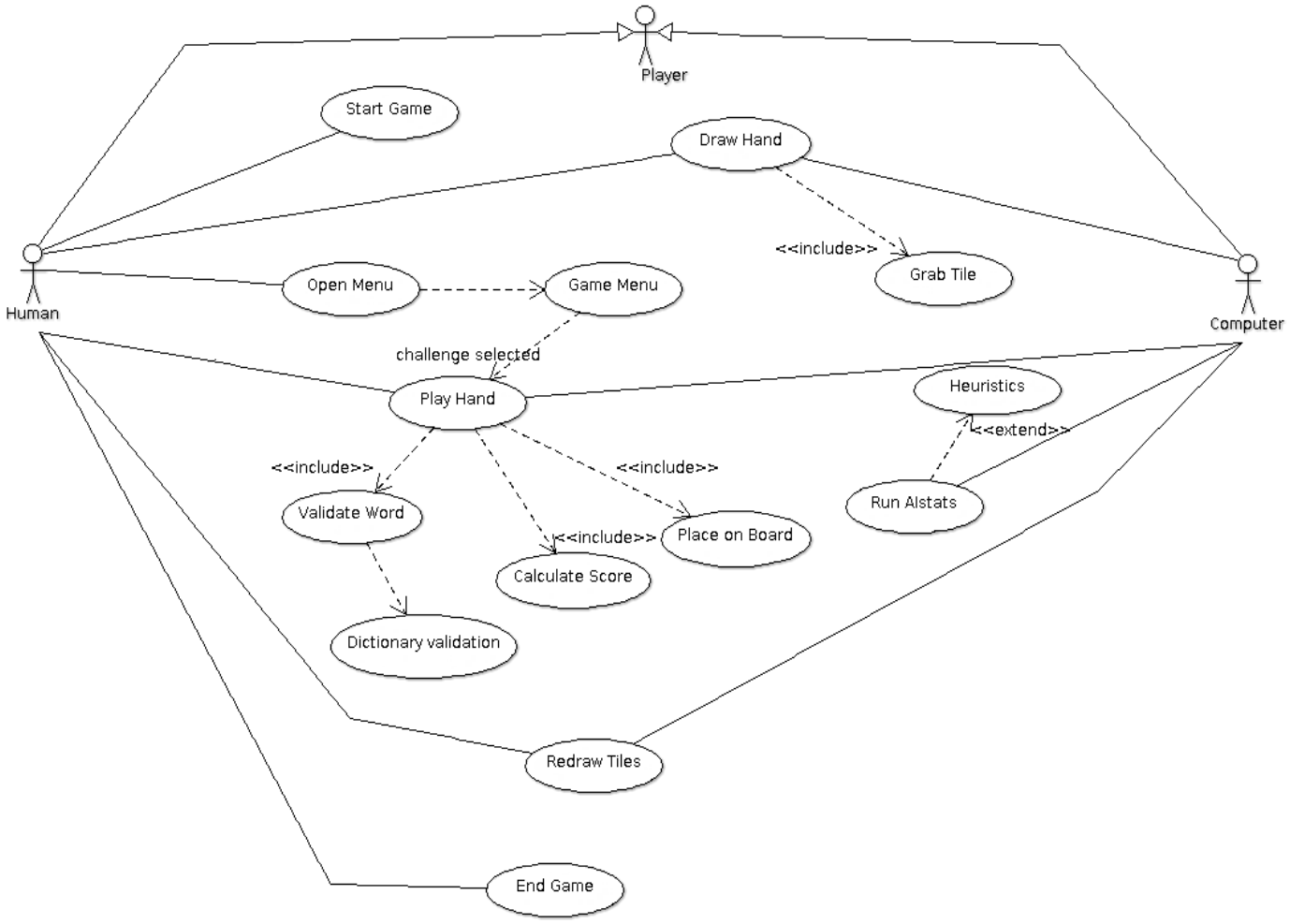
The version of Scrabble presented in the project has the following scoring strategy: Each tile has a preset number of points and the board itself has some bonus points. The aim of the game is to get as many points as possible.

When it is the user's turn, a tile bar is displayed at the bottom of the screen. The user then has to form a word from these and using the cursor place the letters one by one onto the board which has been coloured as to indicate the bonus scores. After the first 2 letters are placed the particular row or column gets highlighted. The said word must join on to what is already on the board, it is then locked by pressing play. Next, is the computer's turn. A blue progress bar is displayed as the AI calculates the next move. The algorithm tries every combination of letters it can play and keeps track of which moves are both valid and score enough points. The AI has been built to give a result within 15 seconds.

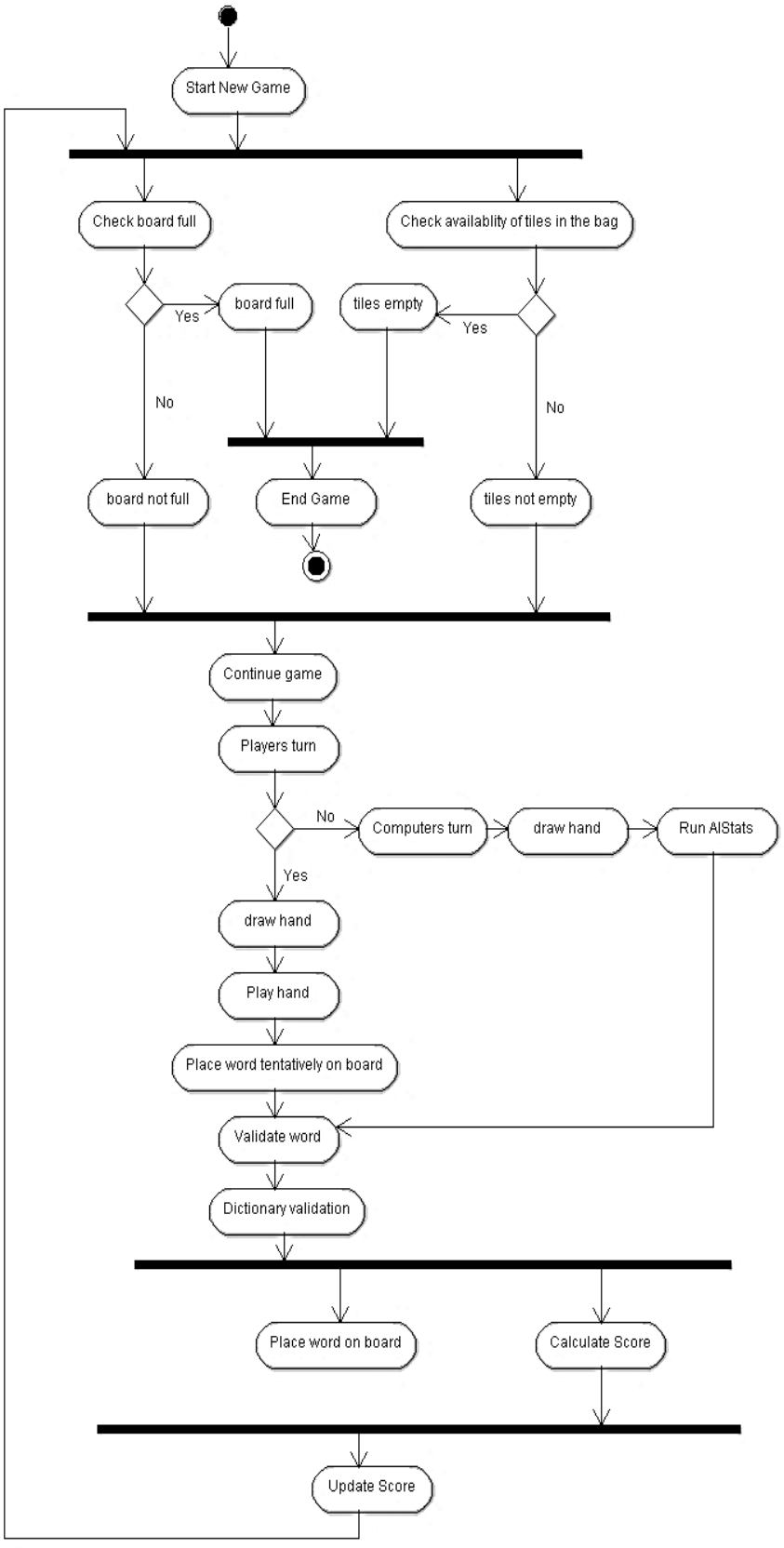
CLASS DIAGRAM



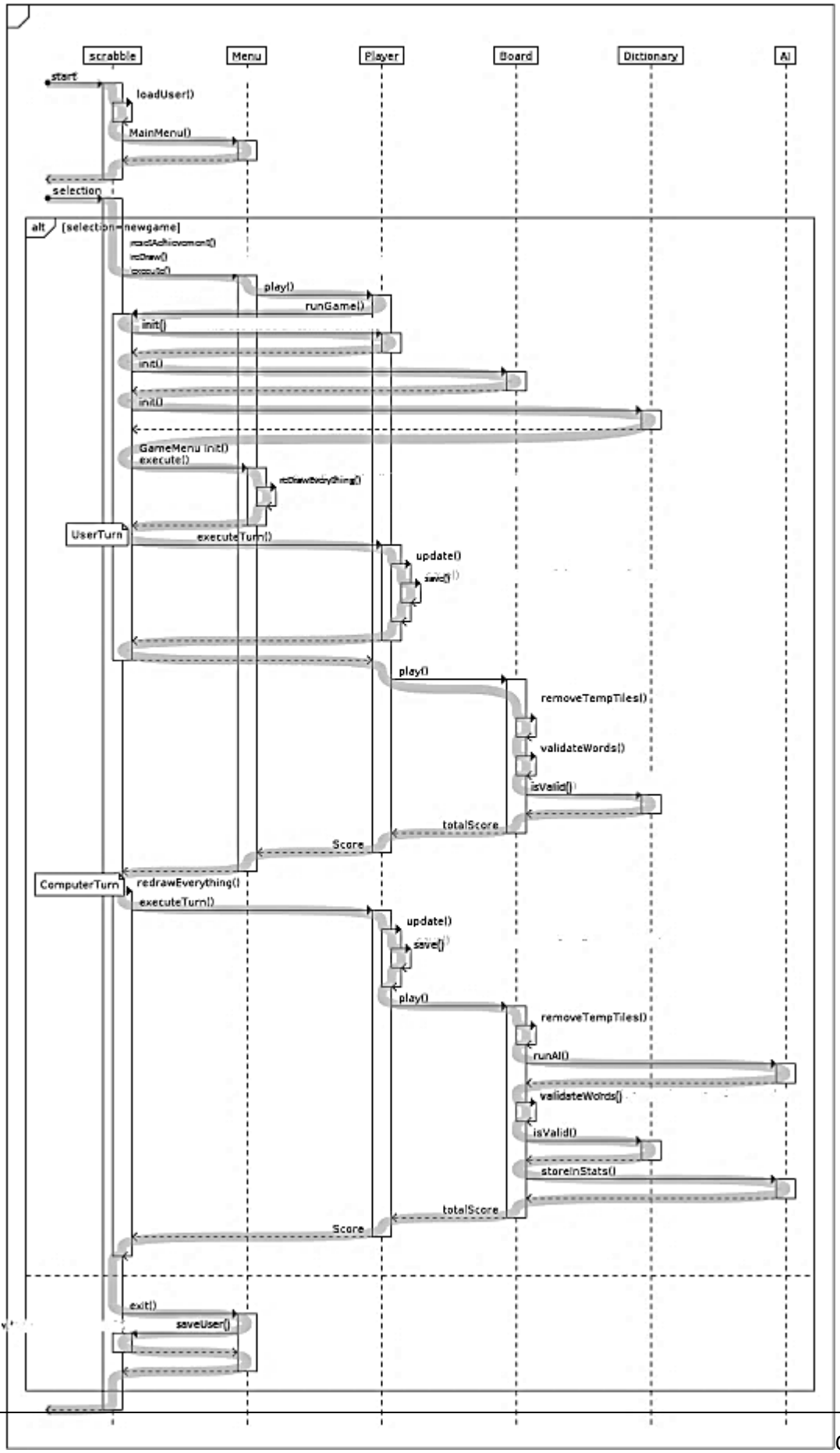
USE CASE DIAGRAM



ACTIVITY DIAGRAM



SEQUENCE DIAGRAM



OAD

3. Implementation

The project presented has been implemented in 3 parts.

1. Graphical Interface.

2. Play Functions for Human Player

3. Algorithms for Computer to play optimally

3.1 Graphical Interface

The interface has been implemented using the pygame package of the Python library. The ‘pygame.display’ modes are used to initialize the game window and display the main menu. The main menu consists of buttons that allow the user to start the game or exit game. Mouse Events are used for implementation of action listeners. Start Game button updates the screen to the Game Menu which consists of the crossword board, score board, tile bar, computer progess bar an two buttons: one to confirm play and the other to exit. This updation of window occurs due the ‘pygame.display.update’ function defined in the pygame module. Functions to colour the crossword board for bonus points, place the tiles on the tile bar are written and mouse motion events are again used for action listening. ’pygame.display.blit’ function is used to display textboxes and user defined buttons onto the window .The tiles get highlighted once the user selects them to place on the board. A function is written in the Board class for this.

3.2 Play functions for Human Player

A Player class is implemented which acts as the base class for two classes : 1. Human and 2. AI. The user player is initialized by calling the human class constructor which in turn calls the player class i.e. its base class constructor. The user selected tiles are placed on the board through the place tentative function in written in Board class. Once the user clicks on the play button on the user interface, the user play flag sets and the play function is called. The play functions receives

tiles from the ‘placetentative’ function as a parameter. It then checks if the word is valid or not and returns a success flag. If it is true, the score is calculated, else function to remove the tentatively placed tiles on the board is called. The turn flag is then changed and the which changes the player instance to that of the AI class. The execution occurs similarly in the next turn of the player.

3.3 Algorithm for computer to play

The AI class constructor is called if the player object is an instance of the AI class. This initializes the computer player. The computer then checks the available tiles and seeds (i.e number of blank tiles on the board to extend the crossword).It recursively searches through the wordspace trying all permutations to see which assignment of the word yields highest points. These checks occur within the play function itself, however the condition for their execution is the object being the instance of AI class. To ensure that the recursive calls do not get stuck in finding the optimal word, a time limit of 20 seconds is set after which the function returns with the most optimal valid word found until then. This time is shown by the update bar on the interface screen. The seed ration is updated after every play. The turn again changes to the human player as the flag the reset.

4. Conclusion and References

The presented project thus allows a single player to satisfactorily play scrabble against the computer. It does not leave the necessity for multiple players to be present to play the game. The game is adequate from the user's perspective since the basic functionality for the user play and the AI has been implemented. However since a brute force approach has been used, there is much scope for improvement. The algorithm for calculating possible slots produces duplicates, so making sure every slot is unique will reduce the computer analysing time. Tile slots may also be sorted so as to implement a shortest job first approach. The further advancements in the algorithm could be implementation of various heuristic approaches to let the computer have a more optimal play.

References :

[1].Python Pygame Documentation, “www.pygame.org/docs**”**

[2].Jacobson and Appel, ”http://www.gtoal.com/wordgames/scrabble_solver/findmoves.c.html”

[3].The Phrontistry - 1996-2014 Stephen Chrisomalis - ”http://phrontistery.info/”