



(Company No. 101067-P)

الجامعة الإسلامية العالمية ماليزيا
INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA
يُونِيسَيْتِي إِسْلَامُ أَنْتَارَا بَغْسَا مِلْدِسِيَا

Garden of Knowledge and Virtue

**KULLIYAH OF INFORMATION AND COMMUNICATION
TECHNOLOGY**

SEM 1 2021/2022

CSCI 2304 SECTION 2

INTELLIGENT SYSTEMS

GUESS THE MOVIE

ASSIGNMENT 1

INSTRUCTOR:

ASSOC. PROF. TS. DR. AMELIA RITAHANI BINTI ISMAIL

GROUP MEMBERS:

NAME	MATRIC NO.
AHMED FAISAL	1921967
MUHAMMAD ISMAIL	1922235

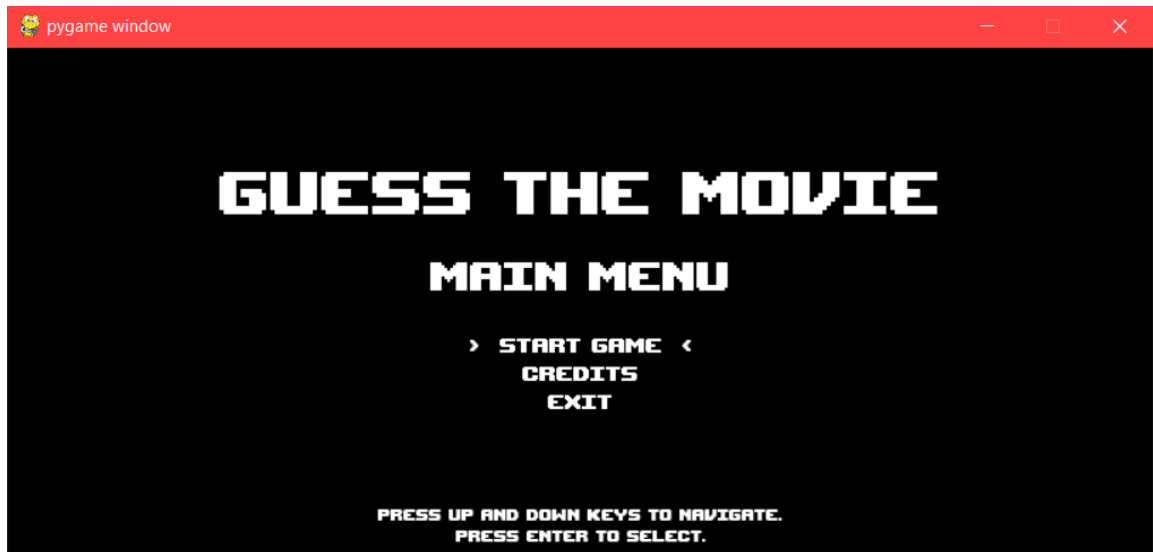
1. Background

Movies are a cultural phenomenon. Majority of people on this planet watch movies. From people who wait patiently for their favourite celebrities to drop a movie to casual watchers that succumb to the trends surrounding them. Everyone has been subjected to movies in one form or another.

Our game would like to test their knowledge of said movies and provide them a challenge to guess 5 different movie names on 5 separate levels within a limited number of chances using their keyboards.

2. Flow of the Game

The game begins with a main menu where users can start the game, check the credits or exit the game.



- a. Upon pressing the start game option, the user can then choose a level out of 5 levels, each level being harder than the last.



- b. After entering a level, the user will find blank spaces on the screen and a life counter in the corner. The user will have to guess which letter may be present in the movie title and fill the blanks.



- c. If the user presses a wrong letter, the game will take away one of its life. So the user will have to be careful to keep in mind the number of lives he has left.
- d. If the user runs out of lives he will be asked to press enter and try again.



- e. If he finishes a level, he will be advanced to the next level until he reaches level 5 and finishes the game.
- f. The users can also go back to the main menu screen using the escape button at any time they please.

3. Explanation of Graph Theory and State Space Search

The search algorithm that we have encoded into our game is the Breadth-First search algorithm. We have used it not just once, but 4 times. Once in the `search_print_letter` method, twice in the `calc_total_lives` method and once in the `victory_state` method. Since they are all using the same algorithm, we decided to elaborate on it just once using our `search_print_letter` method.

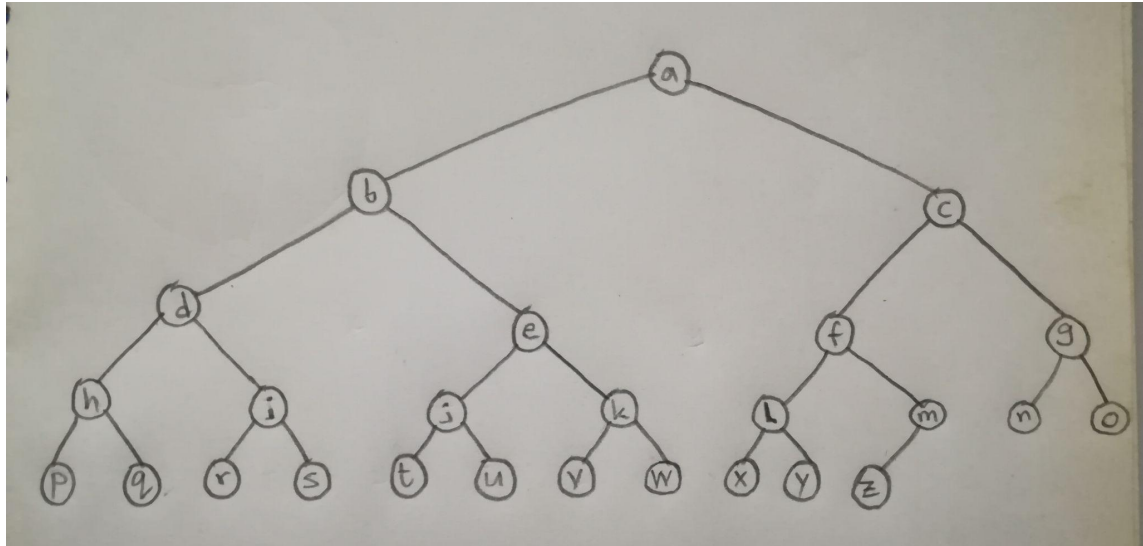
In the `search_print_letter` method, we have implemented the Breadth-First search algorithm as follows:

```
def search_print_letter(self, movie_name, movie_name_length): # Searches and prints letters chosen by the user.
    for i in range(movie_name_length):
        for j in range(self.total_alphabets):
            if j % 2 == 0:
                if movie_name[i] == self.alphabets[j] and self.alphabets[j + 1]:
                    self.level.print_text(movie_name[i], 25,
                                          self.mid_w - 10 - ((movie_name_length - 2) * 10) + (20 * i),
                                          self.mid_h - 5, self.level.WHITE)
```

The way our game works is that, when a letter key is pressed, it is set to True in the default constructor, `Level`, in the `self.alphabets` list. This bit of code checks every frame whether a letter in that list has been set to True or not. This is done by comparing each letter in the movie name to each letter in that list sequentially. If they are the same letter

and that letter has been set to True in the self.alphabets list, the letter is printed onto the screen.

For the sake of visualization, if we were to put the just letters in self.alphabets list into a binary tree, we will get the following:



Our method, `search_print_letter`, basically takes each letter in the movie name and compares it to this tree using the Breadth-First search algorithm. For example: say the 1st letter in the movie name is 'c', our method will first compare it to 'a'. Since they are not the same character, it will move on to the next letter, 'b'. Again they are not the same, so it will move to the next letter, 'c'. This time they are the same. Then, this method will check if 'c' has been set to True or not in the self.alphabets list, or rather, whether the user has pressed 'c' or not. If the user has, then it will print 'c' onto the screen. If not, then it will not print it and will move on to check for the next letter in the movie name.

This whole process is repeated every frame until all the letters in the movie name have been chosen by the user and subsequently displayed on to the screen or until the user has no more lives left.