

Cairo University

Faculty of Computers and Information



CS112

Programming - I

Assignment 1

2018

Ahmed Nasr Eldardery Ibrahim – 20170034

Adham Mamdouh Mohamed – 20170039

Belal Hamdy Ezzat – 20170077

Dr. Mohammad El-Ramly

m.elramly@fci-cu.edu.eg

Team Members:

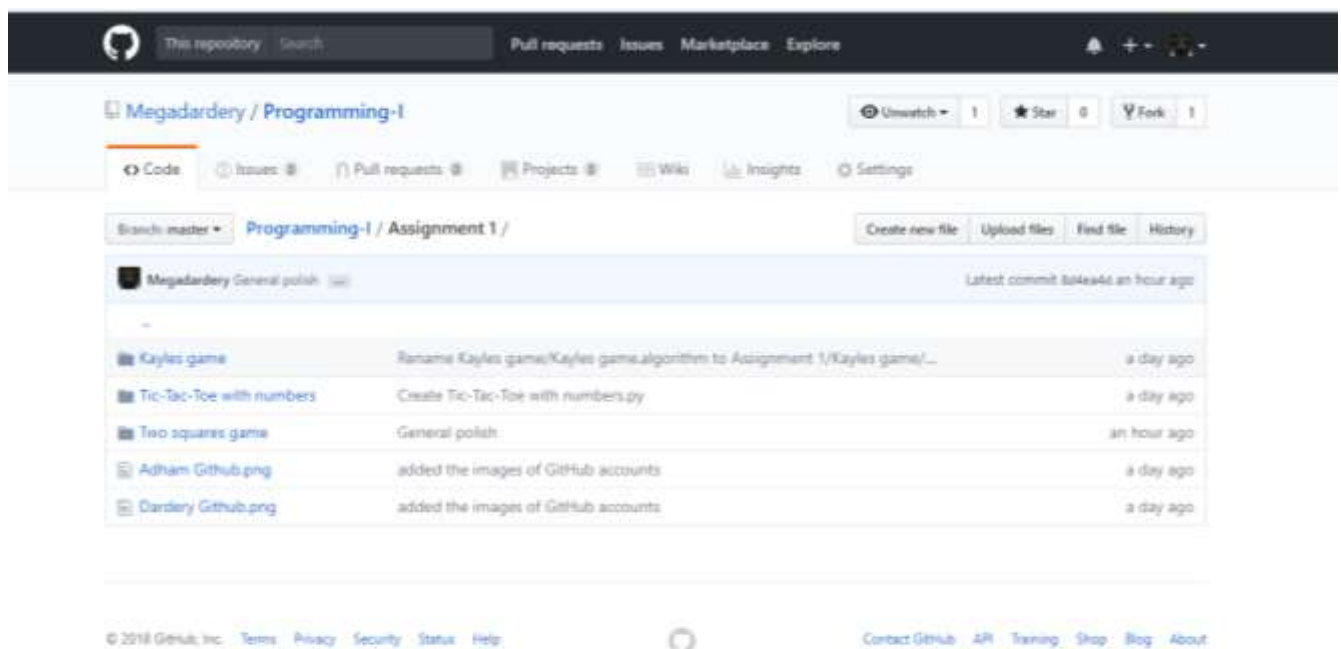
- Ahmed Nasr Eldardery Ibrahim – 20170034
- Adham Mamdouh Mohamed – 20170039
- Belal Hamdy Ezzat – 20170077

List of games made:

- Two squares game
 - With AI.
 - With Graphics
- Kayles game
 - With AI.
 - With Graphics
- 100 game
 - With AI.
- Tic-Tac-Toe with numbers (extra)

Github Repository:

<https://github.com/Megadardery/Programming-I/tree/master/Assignment%201>



Github Accounts:

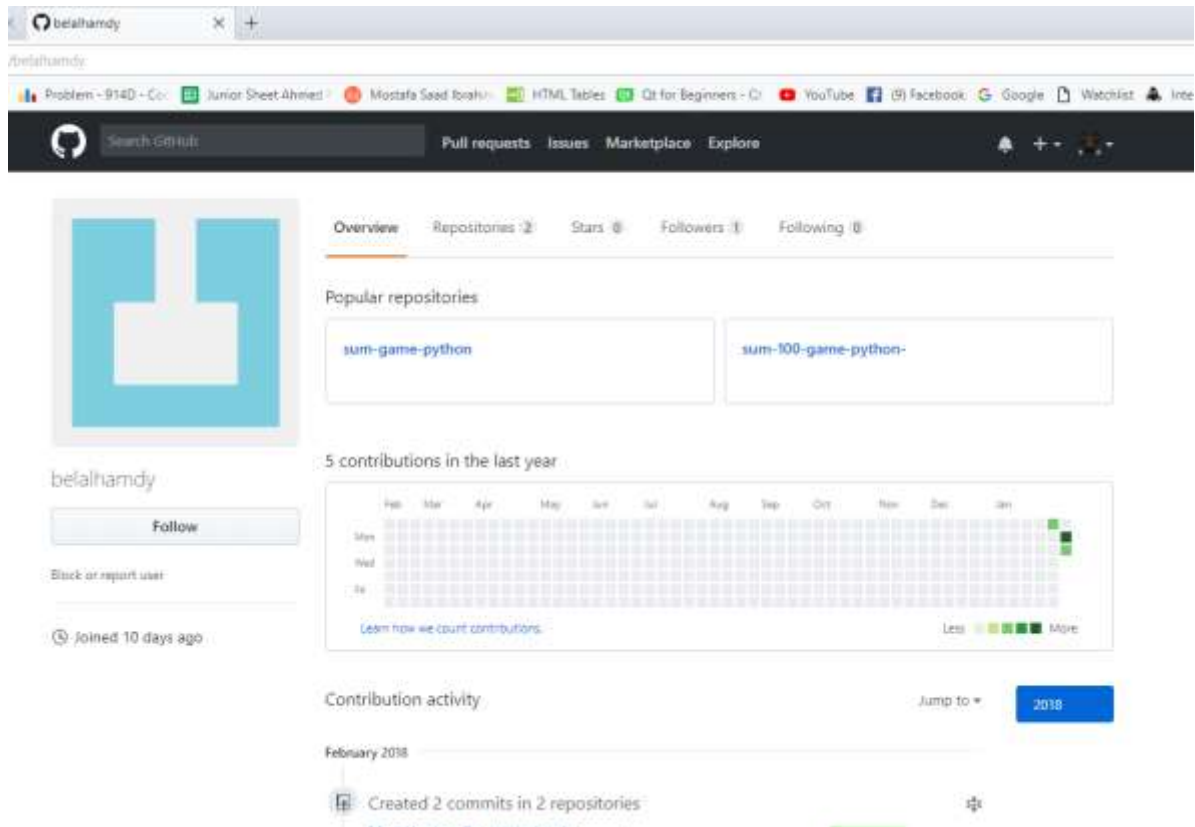
Ahmed Nasr Eldardery Ibrahim (megadardery):

The screenshot shows the GitHub profile of Megadardery. The profile includes a profile picture of a man with glasses and a dark shirt. The name 'Megadardery' is displayed below the picture, along with the username 'Megadardery' and a link to 'Add a bio'. The navigation bar shows 'Overview' as the active tab, with links for 'Repositories' (8), 'Stars' (0), 'Followers' (0), and 'Following' (0). The 'Popular repositories' section lists several repositories: 'dolphin' (Forked from degasus/dolphin, C++), 'LOVE-Example-Browser' (Forked from love2d-community/LOVE-Example-Browser, Lua), 'machine-language' (C#), 'management-mcqs' (C#), 'Programming-I' (Python), and 'GLideN64' (Forked from gonetz/GLideN64, C). The 'Programming-I' repository is highlighted with a Python logo and a version number '1'.

Adham Mamdouh Mohamed (adhammamdouh):

The screenshot shows the GitHub profile of adhammamdouh. The profile includes a profile picture of a man with short dark hair and a red jacket. The name 'adhammamdouh' is displayed below the picture, along with the username 'adhammamdouh' and a 'Follow' button. The navigation bar shows 'Overview' as the active tab, with links for 'Repositories' (1), 'Stars' (0), 'Followers' (0), and 'Following' (0). The 'Popular repositories' section lists one repository: 'Programming-I' (Forked from Megadardery/Programming-I, Python). The '8 contributions in the last year' section shows a grid of contribution activity for the months of Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec, and Jan. The grid shows contributions in Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec, and Jan. The 'Contribution activity' section shows a bar chart for February 2018, with a 'Jump to' button and a '2018' button.

Belal Hamdy Ezzat (belalhamdy):



Algorithms:

100 Game:

1. the user chooses the number of players (1-2).
2. if the number of players == 2 .
 - 2.1. The first player enters number from 1 to 10 and from 1 to (100-sum) (errors are handled).
 - 2.2. Then we add to the sum the number of the first user .
 - 2.3. Then checking if the sum reaches 100.
 - 2.4. Repeat these steps with player 2 until one from them wins.
3. if the number of players == 1 .
 - 3.1. The player enters number from 1 to 10 and from 1 to (100-sum) in the same time.
 - 3.2. Then add this number to the sum.
 - 3.3. In strategy - based pc it makes somehow calculations to play the probably number. This strategy has a bug to make the user win.

Two squares game:

//Main game algorithm:

1. Take valid input from player 1:
2. Mark his/her inputs as filled
3. If no more moves are possible:
 - 3.1. player 1 won
 - 3.2. exit
4. Take valid input from player 2:
5. Mark his/her inputs as filled
6. If no more moves are possible:
 - 6.1. player 2 won
 - 6.2. exit
7. go to step #1

//Perfect AI. algorithm

//(Mirror player 1's input)

1. Get player 1's move as x and y
2. $a = 16 - x + 1$
3. $b = 16 - y + 1$
4. Set A.I's move as a and b

//Know if there are possible moves:

1. FOR i FROM 1 TO 16 (inclusive)
 - 1.1. IF (the ith square is free) AND (the square to the right of the ith is free OR the square under the ith is free)
 - 1.1.1. return ValidMoveExists
2. return NoMoreMovesArePossible

Kayles game:

```
1.  list =
['1','2','3','4','5','6','7','8','9','0','1','2','3','4','5','6','7','8','9','0']
2.  spaces = 20
3.  If player1's turn
3.1.    Print Player's turn
3.2.    Get Input from Player 1
3.3.    If Input is one number:
3.3.1.      Set x = input
3.3.2.      Set y = x
3.3.3.      Decrement spaces by 1
3.4.    Else:
3.4.1.      Set x,y to the two numbers from input
3.4.2.      Decrement spaces by 1
3.5.    If x,y are not possible moves: goto 3.2.
3.6.    set turn to player2's

4.  Else If player2's turn
4.1.    Print "Player 2's turn"
4.2.    Get Input from Player 2
4.3.    If Input is one number:
4.3.1.      Set x = input
4.3.2.      Set y = x
4.3.3.      Decrement spaces by 1
4.4.    Else:
4.4.1.      Set x,y to the two numbers from input
4.4.2.      Decrement spaces by 1
4.5.    If x,y are not possible moves: goto 4.2.
4.6.    set turn to player1's

5.  Update list according to x, y

6.  If spaces == 0
6.1.    If player1's turn: print "player 1 is the winner"
6.2.    Else: print "player 2 is the winner"
6.3.    Exit loop
7.  Go to step 3
```