**Introduction to Computer Science**

**Section G**

**PROGRAMMING PROJECT**

**PHASE 1 Due:** Monday, November 24, 2014 (You have to slip it in my office through the door)

**PHASE 2 DUE:** Monday, December 08, 2014. (anytime till the labs are open)

**IMPORTANT**

* Use const type and DO NOT HARD CODE ANY NUMBER
* Your code should be properly commented
* Variable and function names should be meaningful
* You ARE NOT ALLOWED to use break, continue and goto
* You ARE NOT ALLOWED to use global variables
* You ARE NOT ALLOWED to use more than one return statement in a function
* Your program should be user friendly
* Deduction of marks for messy and confusing code
* **Plagiarism will not be tolerated**. It will result in a straight F in the course and forwarded to DC committee, who might award 5 F’s in all courses you are taking.

You have a choice between the following projects:

1. 2048 game
2. Your version of angry birds
3. Simplified form of robozzle
4. Your version of Eliza
5. Simple physics simulator
6. Your version of candy crush
7. A game of your choice but it has to be discussed and approved by Dr. Mehreen. You must discuss it before the 20th, otherwise you have to choose one project from within the list. Games like pacman, space invaders, tic tac toe, etc. are **not acceptable**.

**PHASE 1**

For phase 1 you have to design the system on paper. You have to submit a map of all functions that you will implement along with how data is exchanged between the different functions. Use an ellipse to denote a function and write the function name within that ellipse. Show data exchange between different functions using arrows, each arrow being labeled by the data. This should be hand written and if one A4 sheet is not enough then you can attach two or three sheets together.

**PHASE 2**

You have to submit a complete working system, along with the source code and a readme file that tells us how to use your software.

**DETAILS OF THE GAMES**

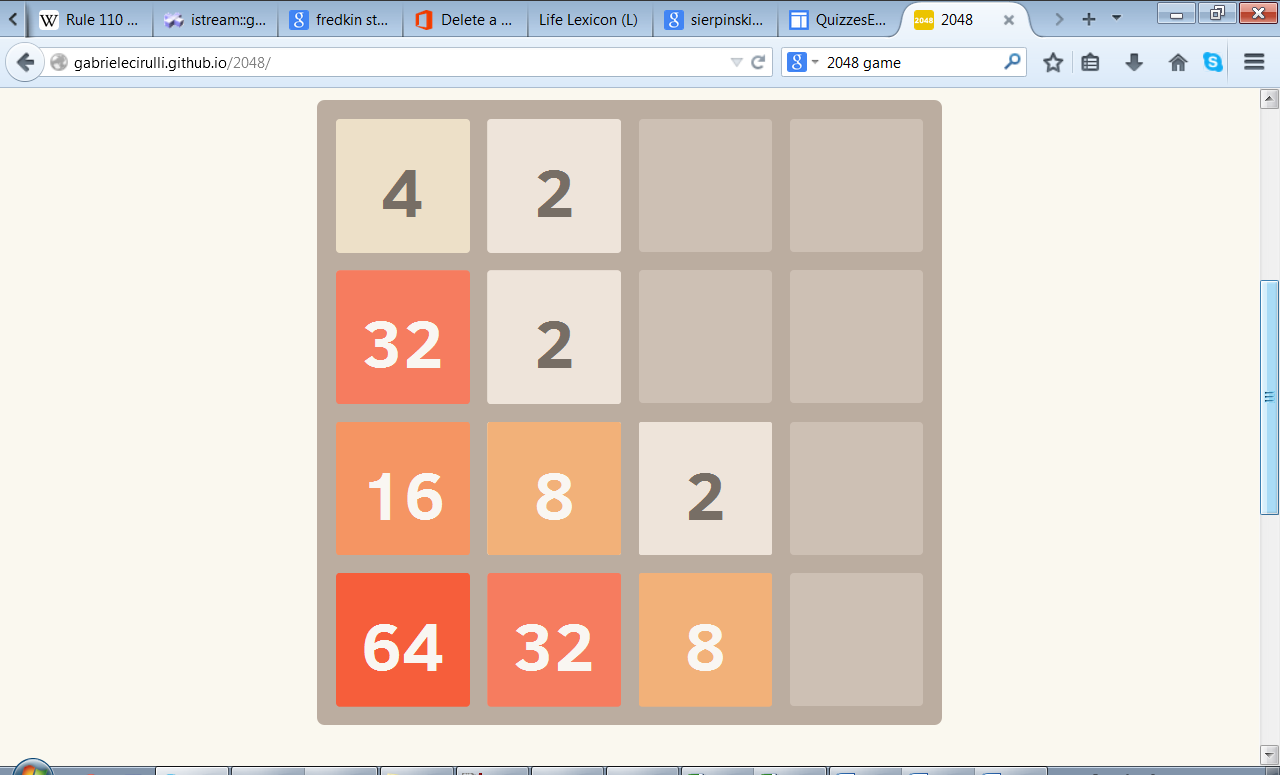
Decide which game you want to implement and read its details as given in the text that follows.

**2048 GAME**

**Difficulty level: easy**

Play the game on:

<http://gabrielecirulli.github.io/2048/>



Implement all parts of the game, along with current score and best score. The ‘start new game’ option should also be there.

**OPTIONAL** for extra bonus: To add different flavor to the game you can make a bigger board with the numbers in powers of 3 instead of powers of 2. So this time three 3s will make a 9 and three 9s will make a 27 and so on.

Make the user interface as nice as possible

*Good luck*

**YOUR VERSION OF ANGRY BIRDS**

Missile fired by the shooter

**Difficulty level: intermediate**

Make a simpler version of angry birds as below:

target that will bounce up and down on the bar below

shooter, that the player can rotate via the arrow keys

**NOTE:**

You have to implement the laws of motion to develop this game so you need to study the trajectory of a projectile. Any extra parameters should be input by the user in a creative way

When the user fires, the missile, it follows the trajectory of a projectile to hit the target. If the target is hit, increase user’s score otherwise give him/her negative points

Be creative and think of all the fun things you can implement here

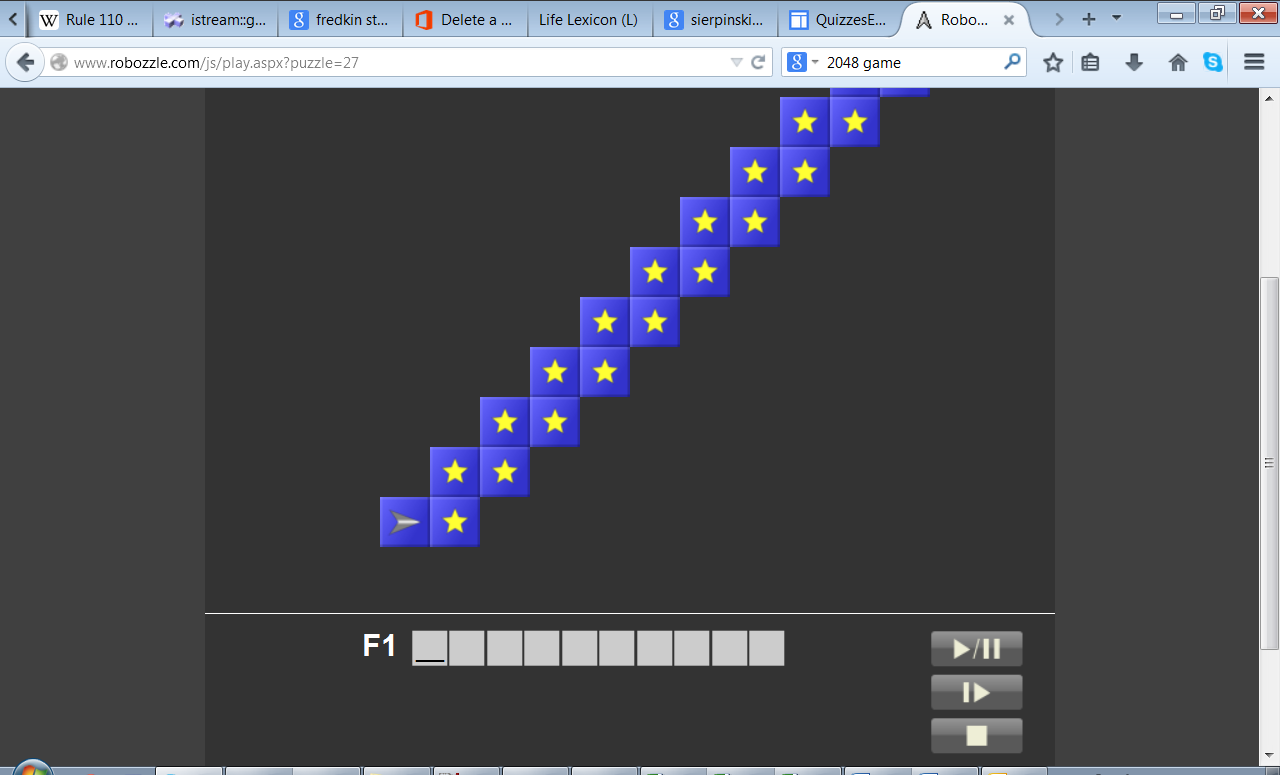
*Good luck and enjoy*

**SIMPLIFIED VERSION OF ROBOZZLE**

**Difficulty level: challenging**

Play the game at:

<http://www.robozzle.com/>



Implement a simpler rectangular grid of stars. You can read it from a file and that way it can be changed easily changed before starting the game. You can put a ‘-‘ in cells where the robot is not allowed to go

Implement only F1 and get a bonus for implementing F2.

Only implement the play button

Implementing robozzle is like implementing a small interpreter which looks at a series of commands and executes them. You can define special symbols/characters for commands. Like ‘f’ for forward, ‘r’ for rotate right, ‘l’ for rotate left, etc. Show the symbols and meaning at the bottom of the screen. Play the game online to understand how it is working.

Implementing a challenging problem is always a source of joy and satisfaction

*Good luck*

**YOUR VERSION OF ELIZA**

**Difficulty level : intermediate**

Try out Eliza at:

<http://nlp-addiction.com/eliza/>

Eliza was one of the first computer programs to demonstrate the concept of natural language processing and building a ‘chatbot’ or a ‘talking’ agent specialized for a certain task. You have to build your version of Eliza that specializes in performing mathematical calculations. For example:

User: Calculate for me 2+3

Your program: The answer is 5

User: What is 10 multiply 7

Your program: The required result is 70

This program is an example of text parsing and text processing. So given a string of characters, you will have to separate out the numbers and numeric operators and perform the calculation. The numeric operators can also occur in the text as strings. You have to do a minimum of ‘+’, ‘-‘, ‘\*’,’/’, ‘%’ . Your program’s responses should also be more ‘human like’ and having minimum repetitions.

This would be your first fabulous artificial intelligence program

*Good luck*

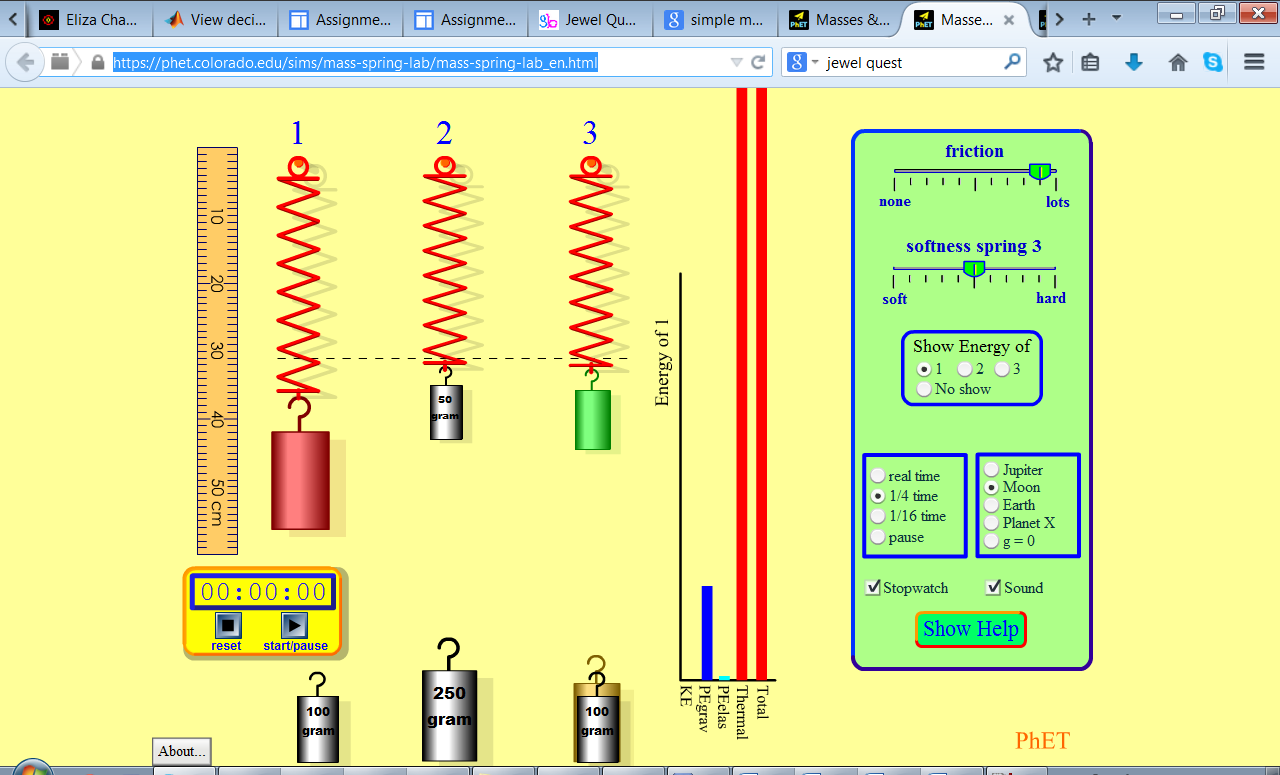
**SIMPLE PHYSICS SIMULATOR**

**Difficulty level: intermediate**

Try out this masses and springs simulator at:

<https://phet.colorado.edu/sims/mass-spring-lab/mass-spring-lab_en.html>

Show the scale



User should be able to select different weights using arrow keys

Find a creative way to input different parameters required for the mass and spring equation for each spring

Implement the animation for the three springs

You need to implement the equations governing the motion of springs and masses.

Writing simulations are not only fun but also a great learning experience

*Good luck*

**YOUR VERSION OF CANDY CRUSH**

Difficulty level: easy



Implement the special candy as bonus

The user can select the two candies to swap using arrow keys and enter key

For extra bonus you can make special patterns like L-shaped patterns disappear

Implementing games and animations enhance your programming concepts

*Good luck*