Gebze Technical University
Computer Engineering Department

CSE443 - Object Oriented Analysis and Design
Fall 2017-2018
Homework 1 - v2.

You have just been hired by “Oyun Bizim İşimiz A.Ş.” as developer for their new game JobHunt. It’s a standard snake game with a new twist.

The main character can be either a python (made up of n squares and initial score multiplier of x1) or an anaconda (made up of 2n squares and initial score multiplier of x2) (chosen at start).

The snake is a new graduate looking for work. Random job offers appear on the game board as items and the snake tries to catch them by eating them. The snake is constantly on the move, you can only control her/his direction. The job offer items stay on the board for a limited amount of time. For every job offer the snake catches, she/he earns points. However, the snake also needs food. That’s why she/he also needs to eat the food items from the board in order to maintain her/his stamina that is constantly reduced over time. If stamina reaches zero, the game ends. Food items also appear randomly on the board for a limited amount of time. The more job offers it eats, the longer the snake becomes. Of course if the snake crashes on the borders or on itself the game ends.

After eating 5 job offers a round ends (the snake moves faster with every passing round). At the end of every round, give the player an option to decorate her/his snake with one object from: foreign languages (x2), training certificates (x3), internships (x1.5); each of which will add a small multiplier to the base score-multiplier of the snake. The graphical interface must show the total score-multiplier (decorations included).

e.g. initially the score multiplier is x1. After eating one foreign language item it becomes x2. After eating another one it becomes x4. After eating an internship item it becomes x6 and so on.

The game is at its very initial stage, so your boss wants you to design the classes so that it’ll be very very easy to add new snakes and decorations in the future. Use the decorator design pattern to solve this issue.

Draw the class diagrams (20p) and implement the game (80p).