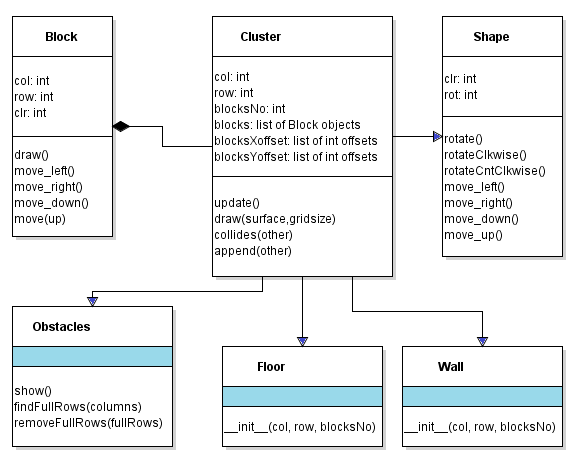
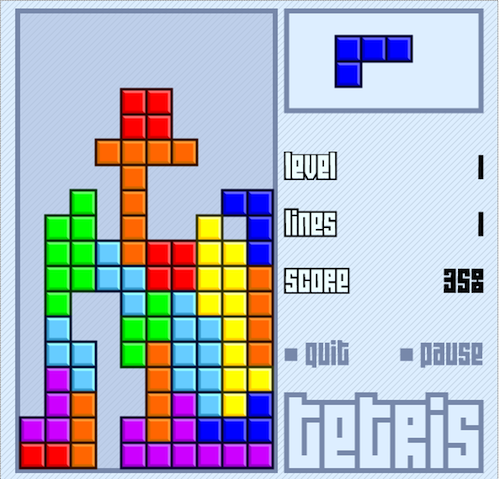
Tetris – Assignment DUE May 22TH Dead line: may 26th

A **Class Diagram** is a visual description of the objects and classes used in the program, and the relationships among them. The diagram doesn’t show how classes are built, but only shows: class attributes, class methods, interactions among objects. We have built the following classes:



The arrows in the class diagram show all inheritance relationships we’ve created. All members from the parent class are also available in the child class.

It is your job now to finish the game, applying your knowledge about Object Oriented Programming.



**Remember: Avoid low-level data processing in your program.  
 Tell your objects what to do, and teach them how**

**to do it in the class.**

Read the hints on the next page and think!

Have fun with the Tetris Game!

**Download from Moodle and read carefully the rubric!**

**You will have to submit the rubric back, with the self-assessment column filled.**

TETRIS Game RUBRIC DUE May 22TH Dead line may 26th

**PROJECTS WILL NOT BE ACCEPTED FOR MARKING AFTER MAY 30TH 23:55**

|  |  |  |  |
| --- | --- | --- | --- |
| **Description** | **Self** | **marks** | **Teach** |
| **KNOWLEDGE \_\_\_/ 15** | | | |
| game field is drawn on the game window (grid can be shown, walls can be shown) |  | **2** |  |
| left and right walls are correctly created and limit horizontal movement of shapes |  | **2** |  |
| floor is correctly created and stops the shapes from moving down |  | **2** |  |
| conflicts between rotating shape and the adjacent border are detected and processed |  | **3** |  |
| new shape appears at the top/middle of the game field |  | **1** |  |
| game keeps a score and displays it every new line is 100 points |  | **2** |  |
| game is over when the current shape collides with the top (create an object – roof or top) |  | **2** |  |
| game displays a background picture and not just black screen |  | **1** |  |
| **APPLICATION \_\_\_/ 20** | | | |
| current shape moves down by itself and stops when it hits an obstacle |  | **4** |  |
| conflicts between rotating shape and obstacles are detected and handled properly |  | **3** |  |
| scoring system is as follows:   * 100 for clearing a single line; * 800 for clearing four lines simultaneously (Tetris); * 1200 for back-to-back Tetris ( four lines |  | **4** |  |
| speed of falling shapes determines the current level, which increases with the score   * After score 500 the game moves to level 2 and medium speed, * After score 1000 the game changes to level 3 and fast speed |  | **3** |  |
| current level is displayed on the screen |  | **1** |  |
| running time is shown on the screen |  | **2** |  |
| next shape is shown on the screen (beside the game field) |  | **3** |  |
| **THINKING \_\_\_\_/ 20** | | | |
| animation is smooth and at a good speed – changing smoothly |  | **2** |  |
| sounds effects are added when dropping a shape and when clearing a line |  | **2** |  |
| pictures/tiles are used to draw the blocks (not just coloured squares) |  | **2** |  |
| game is completed and the player can't crash it (no glitching or freezing) |  | **1** |  |
| Extra Features are added choose from the following: NO EXTRA BONUS MARKS!   * **Intro & Game over screens with images (in separate functions)**➋**,  background music ( original – not my example)** ➊**, pause button** ➊ * **Shadow is added – only unique solutions will be considered! Don’t stare your code!** ➌ |  | **7**  **max** |  |
| zip files and self-assessment rubric is submitted before due date |  | **3** |  |
| independent work in class is observed - no help in troubleshooting and problem-solving |  | **3** |  |
| **COMMUNICATION ­­\_\_\_/ 10** | | | |
| variables (in the main program) are organized in sections and commented (no globals) |  | **2** |  |
| private variables and methods are protected/hidden – game template has no access |  | **2** |  |
| clear meaningful comments that elaborate each section of code – align them on the right |  | **3** |  |
| main program is well formatted – main program is cleared and code is set into functions |  | **3** |  |