

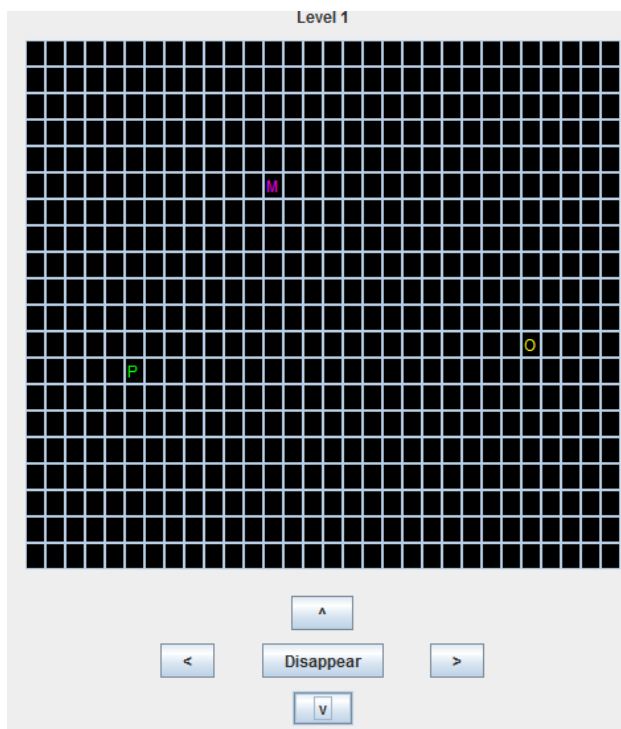
Writing a simple class

Documentation and style will be assessed as part of your grade.

Download two files for this project from the Schoology's course site. They are:

- `Player.java`, which you will complete for a game.
- `game.jar` contains all of the necessary components for the game, including a GUI.

The game provides a series of levels. In each level, the user controls a character (representing the game player) which must collect treasures and avoid monsters. The character is controlled by five buttons: to move up, down, left, and right, and to disappear or reappear. The monsters will move toward the player (the character on the grid), and if one “gets” the player, the game is over. The player collects treasures by moving into a space that currently holds a treasure. When all treasures in a level are collected, the level is cleared and the player moves on to the next level. When all levels are completed, the game is won. The initial screen of the game looks like as follows:



The green 'P' stands for the player.
The yellow 'O' stands for the treasure.
The magenta 'M' stands for the monster.

The “Disappear (or Reappear)” button makes the player appear or disappear. The advantage of disappearing is that the monsters cannot see the player (but they can still get the player if they happen to move into the player, or if the player happens to move into them while the player is invisible). The disadvantage is that the player character (“P”) is invisible and cannot collect any treasure.

In the lower levels, the monsters will stop moving if they don't see the player, but in the higher levels, they will remember where the player were and keep moving toward that spot.

To play the game, compile `Player.java` and run either

```
java -cp .:game.jar MonsterGame (if  
your system is Linux or Mac)
```

or

```
java -cp .;game.jar MonsterGame (if  
your system is Windows)
```

The buttons currently don't work. You will write the methods of the `Player` class to make those buttons work.

The Player class's contract with the rest of the game program is as follows:

- It is given an initial location (`int x` and `y` coordinates in the grid) and the size of the grid in its constructor. The current constructor does nothing with this information.
- It needs to report its current location when the methods `getXPos()` and `getYPos()` are called. Currently these methods always say the player is in position (0, 0). You should update those statements.
- When the user clicks on the appropriate button, the method `up()`, `down()`, `left()`, `right()`, and `disOrReappear()` are invoked on the object.
- When the method `getIcon()` is called, it must return a character. If the player is visible, the character returned will represent the player on the screen; if the player is invisible, it should return a space character. Currently, it always returns the letter 'P' (for the player).

Your task is to finish this class, **according to the above contract**. You should think about:

- What information this object (a Player object) needs to store (this will help you determine what the instance variables should be).
 - Figure out necessary instance variables first. Hint: recall that the constructor initializes instance variables by using the values delivered by the formal parameters.
- What the methods `getXPos()`, `getYPos()`, and `getIcon()` should return, based on the values of the instance variables.
- How the methods `up()`, `down()`, `left()`, `right()`, and `disOrReappear()` affect the instance variables.

Grid positions are numbered starting at zero, increasing toward the right and toward the top. You are responsible to make sure that if the user clicks (for example) the left button when the player is already on the left edge of the board, then the player does not actually move any farther left. Otherwise the program will crash.

Your code should NOT include any particular value (i.e., 5 or 30) for the size of grid or the position of the Player object.

What to submit:

- A hard copy of `Player.java` to the instructor AND
- The `Player.java` file to the Schoology course site.
 - You don't need to create a script file for this project.

Before you turn it in

When you have a program that works, think of it as a rough draft. Now it is time to polish it. Read through it to see if there are things that should be fixed. Are the variable names helpful? Is there anything confusing in what you have written? Anywhere that a comment is called for? Also, make sure that you have followed **the coding guidelines for this class**. **Documentation and style will be assessed as part of your grade.**

Printing files at the CS Lab

```
$ a2ps -2 --file-align=virtual filename(s)
```

For example, the following command prints `PigLatin.java` and `typescript` files.

```
$ a2ps -2 --file-align=virtual PigLatin.java typescript
```