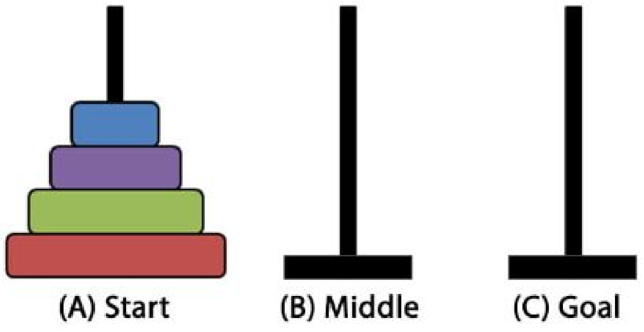
Problem-based Case 3

SVG in a Web-Razor .NET Project

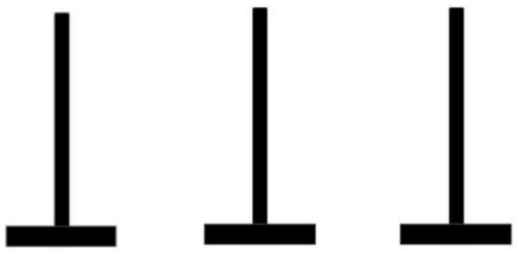
The way of answering is by reporting the results in the personnel chat using the usual way of pasting the screenshots and corresponding code which must include part of your desktop background, and, at the same time, it should include the time of the screenshot. Each expected answer will have an identification alphanumeric string. Please paste it before each screenshot.

In this problem-based case you will implement a manual version of the classic game of Towers of Hanoi,. It is a classic puzzle about three towers and three disks. The problem is to transport all disks from the first tower to the third tower, but never a bigger disk can be on a smaller disk.



1 (20%). Solving a basic SVG injection.

The first part you must put the three towers in a SVG space. But it has some conditions. You need to create a new blazorserver project. Then you need to change the menu to show a different option saying “Towers of Hanoi”. And in this option the content must be the three empty towers, like this:

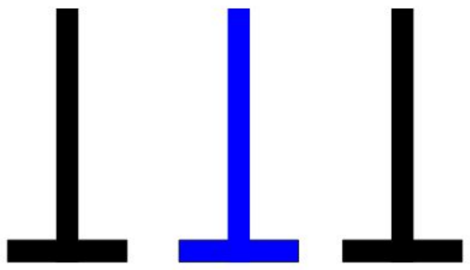


The expected answers and their identifications and contents are:

- PbC03-1-A. An screenshot showing the menu and the three empty towers
- PbC03-1-B. The code in the blazor file

2 (20%). Solving a basic SVG interaction

In this part you must introduce a basic interaction, each tower (in the base or in the column) should accept a click, and, the action must imply a color change, from black to blue and from blue to black. When a black tower is clicked and there is another blue tower, then only the clicked tower must change to blue, and the other blue tower must change to black. Therefore the click works like a tower selector.

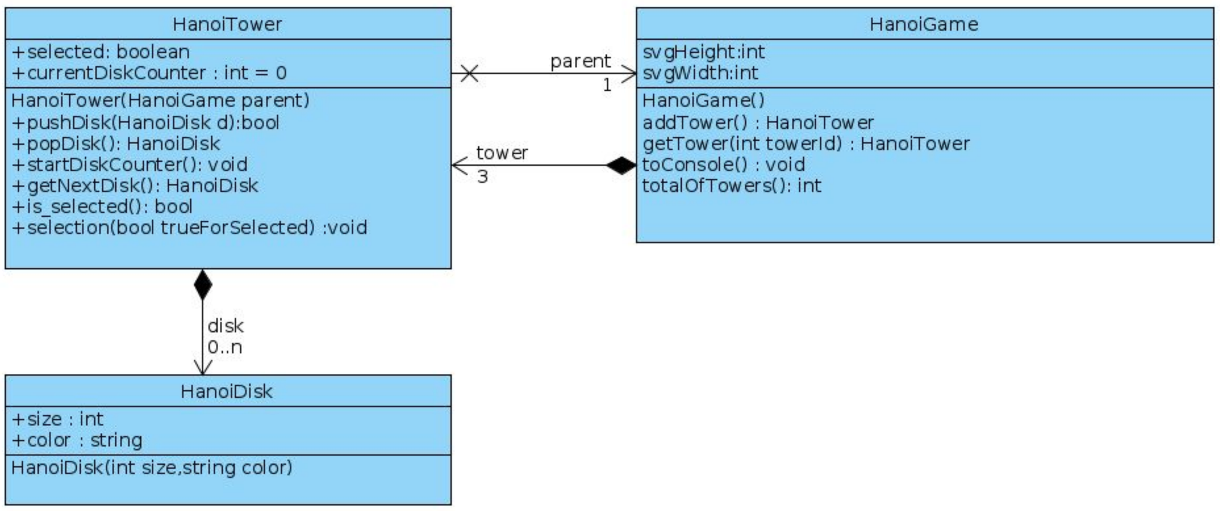


The expected answers and their identifications and contents are:

- PbC03-2-A. Three screenshots showing the sequence of clicking different towers, therefore in each screenshot different towers must hold the blue color.
- PbC03-2-B. The code in the blazor file of the view and the function controlling the onclick event.

3 (20%). Implementing a basic object structure on C# for Hanoi Towers

In this part you must introduce a basic object structure which must implement the following UML model in C# classes. Review de Carrots & Rabbits example in order to identify how UML elements can be implemented on C#.


HanoiTower
+selected: boolean
+currentDiskCounter : int = 0
HanoiTower(HanoiGame parent)
+pushDisk(HanoiDisk d):bool
+popDisk(): HanoiDisk
+startDiskCounter(): void
+getNextDisk(): HanoiDisk
+is_selected(): bool
+selection(bool trueForSelected) :void
HanoiGame
svgHeight:int
svgWidth:int
HanoiGame()
addTower() : HanoiTower
getTower(int towerId) : HanoiTower
toConsole() : void
totalOfTowers(): int
parent
1
tower
3
disk
0..n
HanoiDisk
+size : int
+color : string
HanoiDisk(int size,string color)

Following this schema the instruction for creating the basic Towers of Hanoi, should be

```
h = new HanoiGame();
t  = new HanoiTower(h)
h.addTower(t);
t.puskDisk(new HanoiDisk(6,”orange”));
t.pushDisk(new HanoiDisk(4,”blue”));
t.puskDisk(new HanoiDisk(2,”yellow”));
h.addTower(new HanoiTower(h));
h.addTower(new HanoiTower(h));
h.toConsole();
```

The `toConsole` method must show, in the terminal de state of Hanoi Towers, in the case of the configuration it must be:

```
Tower 0 - 3 disks
Disk 0 6 orange
Disk 1 4 blue
Disk 2 2 yellow
Tower 1 - 0 disks
Tower 2 - 0 disks
```

However after the actions

```
h.getTower(1).push(h.getTower(0).pop());
h.getTower(2).push(h.getTower(0).pop());
h.toConsole();
```

Must show in the terminal

```
Tower 0 - 1 disks
Disk 0 6 orange
Tower 1 - 1 disks
Disk 0 2 yellow
Tower 3 - 1 disks
Disk 0 4 blue
```

To do this the methods `startDiskCounter` and `getNextDisk` result relevant. The first is for starting to count in the base of the involved tower. This way `getNextDisk` returns the next HanoiDisk in the stack of the involved tower, or null either there are no more disks.

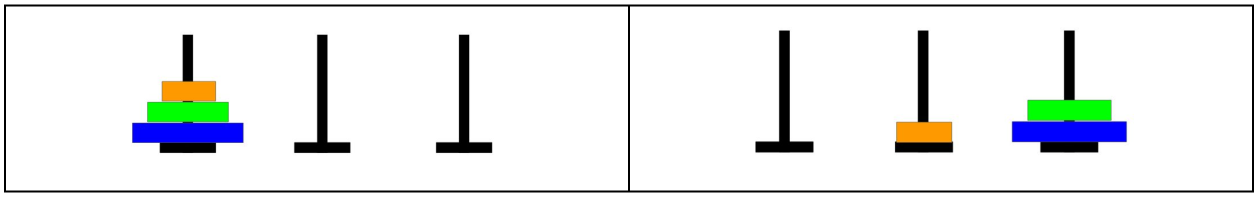
The expected answers and their identifications and contents are:

- PbC03-3-A. The code of the class HanoiGame.
- PbC03-3-B. The code of the class HanoiTower.
- PbC03-3-C. The code of the class HanoiDisk.
- PbC03-3-D. The code for the configuration of the HanoiGame having the two calls to the `toConsole` method in the razor file.
- PbC03-3-E. The **two** screenshots of the terminal, showing both configurations, the initial state and the state after the two simulated pushs.

4 (20%). Displaying the three towers in SVG

In this part you must develop the SVG view of the HanoiGame having the 3 towers. The generation should be generic, it means that must be correct for all the combinations of the three towers having none, one, two or three disks.

To test this, create the following two configurations using the `pushDisk` method from the HanoiTower class.



The expected answers and their identifications and contents are:

PbC03-4-A.	The razor file with the first configuration and the generation of the towers.
PbC03-4-B.	The output of the first configuration in the internet browser.
PbC03-3-C.	The razor file with the second configuration and the generation of the towers (should be the same as the first case).
PbC03-3-D.	The output of the second configuration in the internet browser.

5 (20%). Interacting with the Hanoi disks

In this part you must add an event handler for each disk and each tower in the game. The interaction on the towers must follow the problem 2, i.e., the clicked tower must change from black to blue. Now you must update the display for generating blue towers when they have the `selected` attribute on true. The selected tower is the target tower, thus, when you click a disk, this disk must be added (pop) on the target (blue) tower either it accomplishes the condition of small disks on big disks. If there is no selected towers then a small message must be on the screen.

PbC03-5-A.	The fragment of razor file of the Game of Hanoi with the generic view and initial configuration.
PbC03-4-B.	The fragment of razor file of the Game of Hanoi with both event handlers, i.e. onclick on tower and onclick on disk.
PbC03-3-C.	eeThree outputs of the game in the internet browser
PbC03-3-D.	Compress the folder of the project in **a ZIP file** and attach it as part of your delivery.