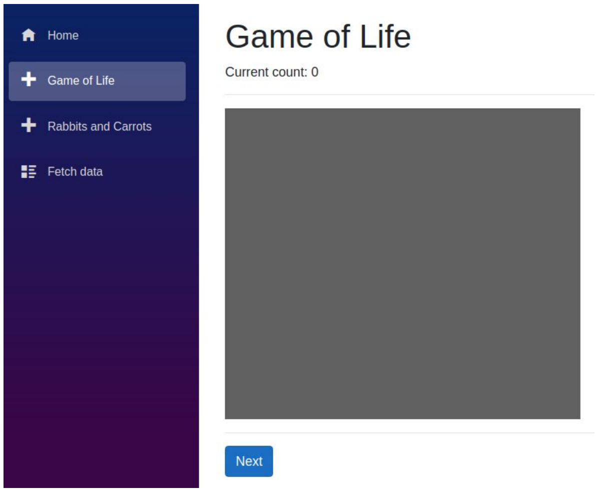
TUTORIAL 2

HCI Elements in a Web-Razor .NET Project

1 (10%). Previously in Advanced Programming …

You must have an advanced version of the previous Problem-based Case. In this problem is required two features, the menu must have the options, Game of Life and Rabbits and Carrots. Moreover, it is required that in the Game of Life appear an empty environment of 30 by 30 cells. The output must look like this:



Game of Life

Current count: 0

Thus, your steps should be:

W1-Step1. Be sure that you have a blazorserver project with a version of the Game of LIfe and Rabbit and Carrots. The minimum is that you can show the menu having the four options.

W1-Step2. Change the dimension of the environment to 30 by 30.

W1-Step3. Run the project, remember to use https://localhost:5001 and choose advanced options to skip the security mechanism.

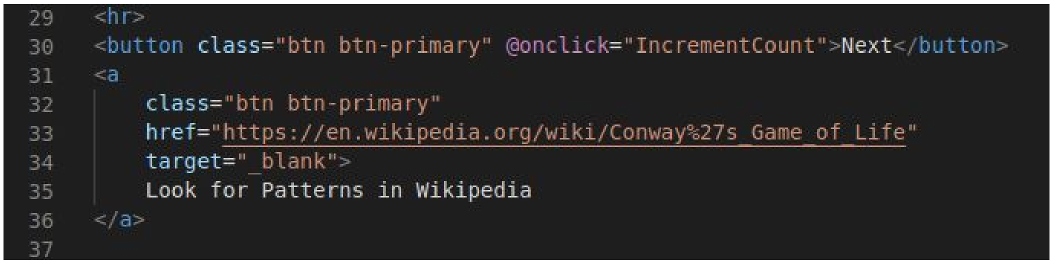
W1-Step4. Take your screenshot. Include part of your desktop’s background, date, and time, then paste it in the personal chat.

2 (20%). Anchors

The <a> HTML tag is for generating links in web pages. This is called an anchor. We can use them in Razor projects too because it has an HTML element. Additionally, we can change its appearance using a known css style (also as a way of applying Nielsen’s consistency heuristic). The <a> elements allow specifying the web site of reference, in the href attribute, and where it can be opened. We can set that the referenced web site uses a new tab by specifying “_blank” as the value of the target attribute.

In this exercise we modify the previous project. Specifically ONLY the file Counter.razor by adding an anchor to the wikipedia definition of Game of Life using the same style as the “next” button.

W2-Step 1. Add the anchor in the Counter.razor file. The following code shows how:

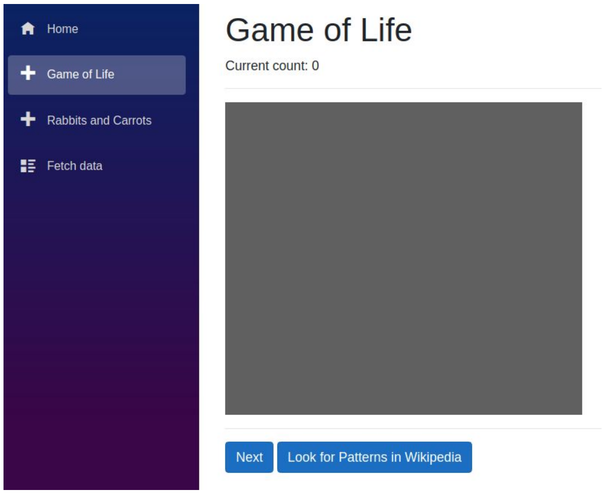


```
<hr>
<button class="btn btn-primary" @onclick="IncrementCount">Next</button>
<a
    class="btn btn-primary"
    href="https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life"
    target="_blank">
    Look for Patterns in Wikipedia
</a>

```

W2-Step 2. Run the project

W2-Step 3. Take your screenshot. Include part of your desktop’s background, date, and time, then paste it in the personal chat.Take the output screenshot following the rules of including part of your desktop’s background, date and time. It should looks like this:

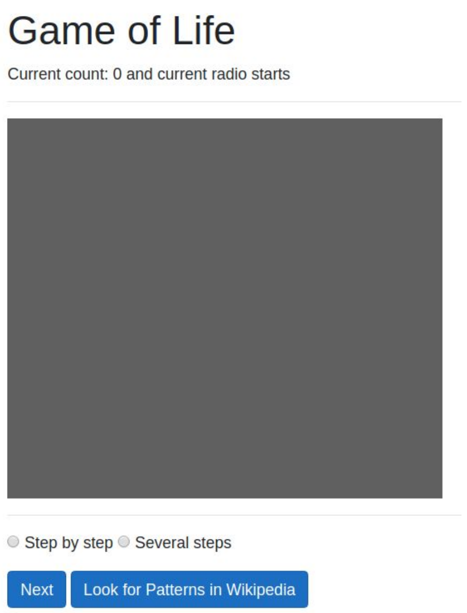


Game of Life

Current count: 0

3 (20%). Radio buttons

Radio buttons implement mutual exclusivity, they act as a group having the same name, but at the same time, each component has a different value. In this exercise you should get the following interface:

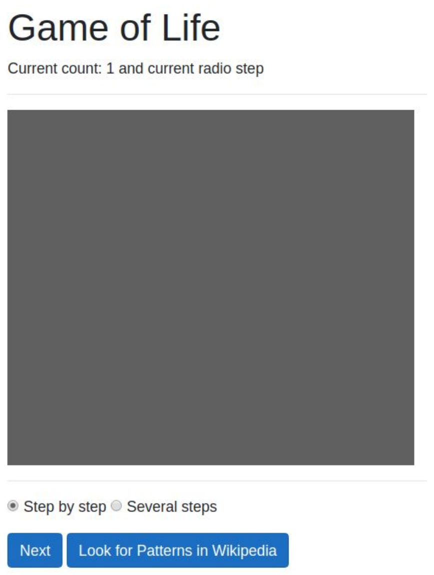


Game of Life

Current count: 0 and current radio starts

○ Step by step ○ Several steps

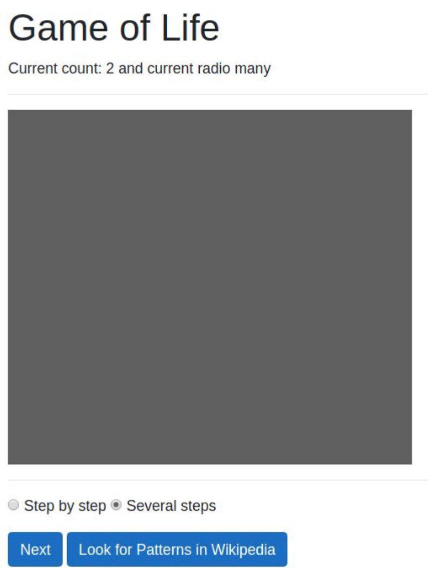
Note that in the superior part the message has been extended. In the foot now we have two radio buttons. But also we have the following behavior: by clicking some of the radio buttons different messages on the top are shown. Watch the following two figuresNote that in the superior part the message has been extended. In the foot now we have two radio buttons. But also we have the following behavior: by clicking some of the radio buttons different messages on the top are shown. Watch the following two figures:



Game of Life

Current count: 1 and current radio step

● Step by step ○ Several steps

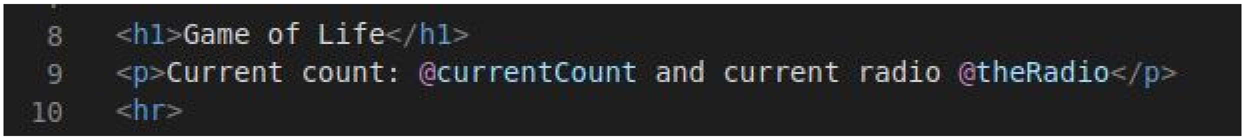


Game of Life

Current count: 2 and current radio many

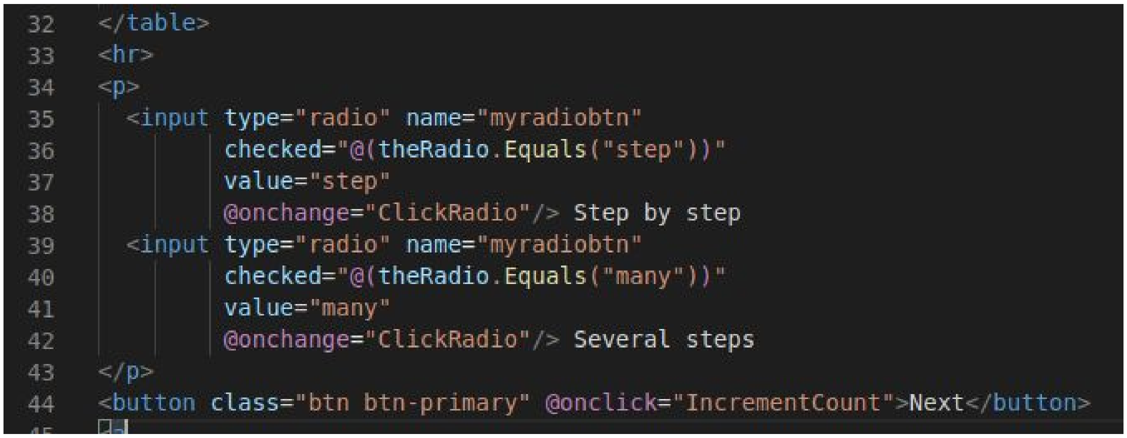
○ Step by step ● Several steps

W3-Step 1. Change the message on the top. We will use the variable theRadio



```
<h1>Game of Life</h1>
<p>Current count: @currentCount and current radio @theRadio</p>
<hr>
```

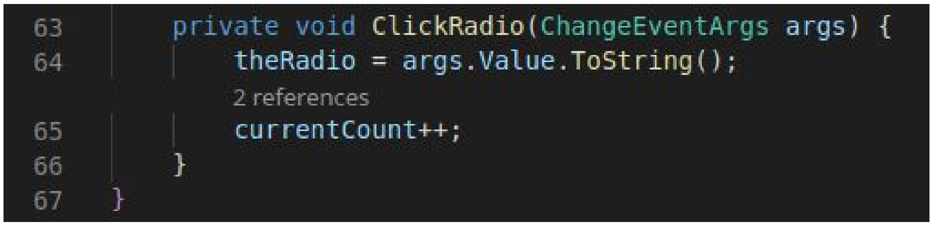
W3-Step 2. Add the radio buttons after the table (the grid) but before the buttons. Note that radio buttons have the same name but different values. Also the razor event @onchange has been used. The checked attribute is calculated (true or false) depending on the current value of the string theRadio. Due to this the first time none of both buttons are checked.



```
</table>
<hr>
<p>
  <input type="radio" name="myradiobtn"
         checked="@(theRadio.Equals("step"))"
         value="step"
         @onchange="ClickRadio"/> Step by step
  <input type="radio" name="myradiobtn"
         checked="@(theRadio.Equals("many"))"
         value="many"
         @onchange="ClickRadio"/> Several steps
</p>
<button class="btn btn-primary" @onclick="IncrementCount">Next</button>
```

As you can see, the @onchange event triggers the ClickRadio method. Therefore it lacks.

W3-Step 3. Add the method ClickRadio in the @code section as follows:



```
    private void ClickRadio(ChangeEventArgs args) {
        theRadio = args.Value.ToString();
        2 references
        currentCount++;
    }
}
```

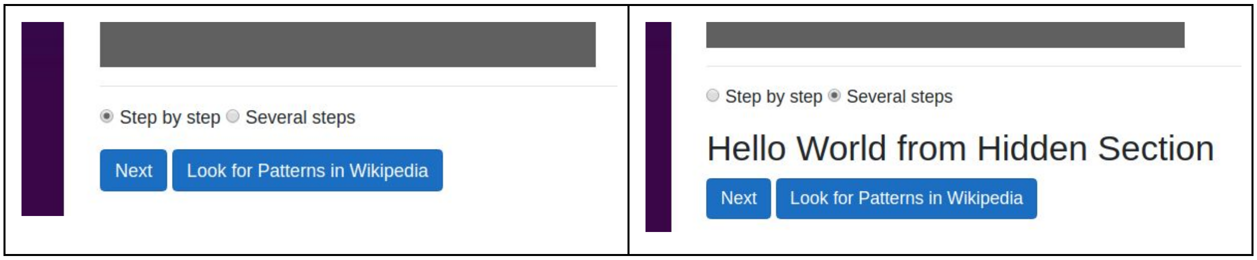
Note that the function is called without arguments but arrives with one argument. This is because the Razor framework adds this argument which represents the clicked object. Note also that, in this method, the counter is increased.

W3-Step 4. Run the project

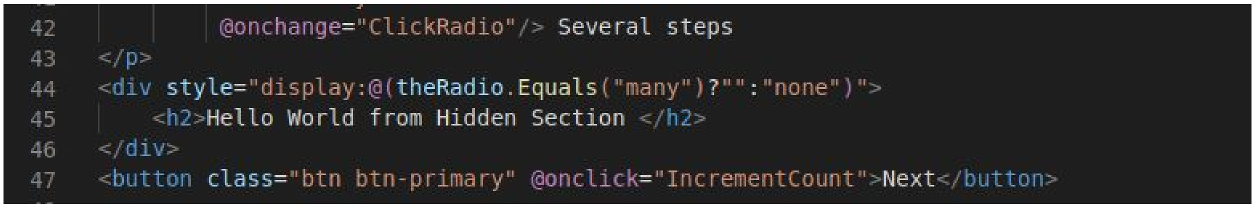
W3-Step 5. Take your screenshot. Include part of your desktop's background, date, and time, then paste it in the personal chat.

4 (20%). Hidden Interface Sections

Hidden interface sections accomplish consistency of interfaces showing only what has sense for the user in this moment of the interaction. In this exercise we will produce a view of a hidden section only when the radio button is on the “Several steps” choice. Following this exercise you should get the next interface:


● Step by step ○ Several steps
Next
Look for Patterns in Wikipedia
○ Step by step ● Several steps
Hello World from Hidden Section
Next
Look for Patterns in Wikipedia

W4 Step 1. In HTML the <div> tag adds a division into the web page. Div elements support the style attribute which allows set the “display” sub-attribute inside it. When is set to “” (empty string) the div is visible and when it is set to “none” the div is invisible, and all its content. Add the following code which includes a div section. Note that it is between the radio buttons and the final buttons.



```
            @onchange="ClickRadio"/> Several steps
</p>
<div style="display:@(theRadio.Equals("many")?"":"none")">
    <h2>Hello World from Hidden Section </h2>
</div>
<button class="btn btn-primary" @onclick="IncrementCount">Next</button>
```

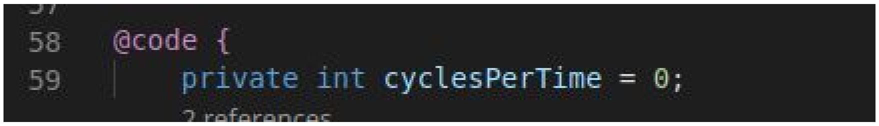
W4-Step 2. Run the project

W4-Step 3. Take **two screenshots**, one for the code and one for the output. Include part of your desktop’s background, date, and time, then paste them in the personal chat.

5 (20%). Input fields

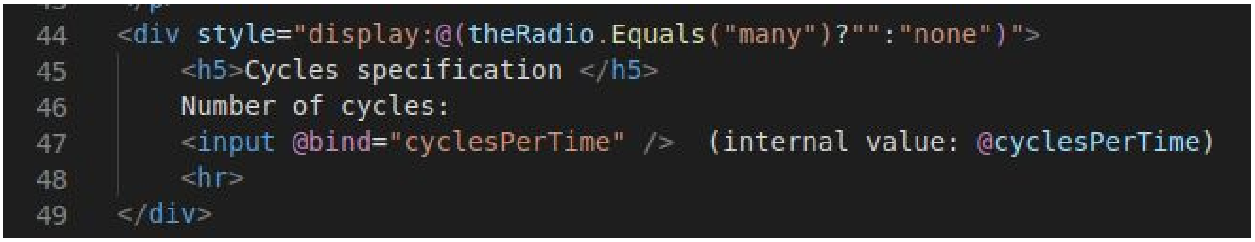
Input fields are the common interface element for entering data to web systems. Razor framework can handle fields by using the @bind directive for connecting the code with the HTML output.

W5 - Step 1. Add the following variable in the @code section.



```
@code {
    private int cyclesPerTime = 0;
    2 references
```

W5 - Step 2. In order to change the internal content of the hidden section, we will bind the previous int variable to the input field. The correct data type is controlled by the Razor framework and it depends on the data type assigned in the code. Note that a “bad” content is changed to the previous value, you can check it by changing the focus using the tab key. The code of the hidden sections has been modified to :



```
<div style="display:@(theRadio.Equals("many")?"":"none")">
    <h5>Cycles specification </h5>
    Number of cycles:
    <input @bind="cyclesPerTime" />  (internal value: @cyclesPerTime)
    <hr>
</div>
```

W5-Step 3. Run the project

W5-Step 4. Take **two screenshots**, one for the new div section and one for the output. Include part of your desktop’s background, date, and time, then paste them in the personal chat

6 (10%). Other input elements

Other input elements exist in HTML, for example <select> for combo boxes, checkboxes, and textareas. For getting the last 10% you should try **ONLY one of them** and put an example inside the hidden section. As the previous exercises you should use a visible output to show the changed values in this field. Also in this case take **two screenshots**, one for the new div section and one for the output. Include part of your desktop’s background, date, and time, then paste them in the personal chat.