**Learning Scenario 6 – Chameleon’s summer vacation**

Task: Create a game where the player will be able to move the main character - chameleon around the screen with arrow keys. Chameleon will change its looks (costumes) based on the color of the background. Background is divided in three parts with uniform color, each representing a different place: blue represents the sea, sandy color for the beach and green for the forest.

Chameleon has to change its looks and also tell where he is in five different situations:

1) when swimming in the sea, he has to change his color to blue and say “I am swimming in the sea”;

2) when he is between the sea and the beach his skin turns half blue-half sandy color and he says “I am between the sea and the beach”;

3) on the beach, he takes on a sandy color and says “I am relaxing at the beach”;

4) between the beach and the forest, he turns half green-half sandy color and says “I am between the beach and the forest”;

5) in the forest, his skin turns green and he says “I am cooling off in the tree shade”;

The process of creating the game:

1. Open the template file:

<https://snap.berkeley.edu/project?user=zapusek&project=chameleon_template>

There is a background divided into three unicolor parts. You will also find five different looks for a chameleon. You can optionally place on a background other items to make a scene more realistic, such as: waves, sea shells, trees… You have to be careful not to choose items that are entirely different color than background and are at the same time bigger than the chameleon. In this case the sensing block won’t be able to correctly detect the color of the background if the player will place chameleon there.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
|  |  |  |  |  |

1. Write a code for moving the chameleon with arrow keys (up, down, left, right). Set his step length to 10 pixels. Remember to rotate him in the appropriate direction before each step.

Chameleon should not move over the edge of the screen so use the block  to make him bounce back.

1. Chameleon changes his looks according to the color of the background at his current position. We can move him freely across the screen, so it is impossible to predict his position beforehand. This is the reason why we have to check his position repeatedly. In this kind of situation where we have to constantly check if something has happened we use a forever loop.

1. Kameleon spreminja barvo glede na ozadje, kjer se trenutno nahaja. Ker lahko kameleona s tipkami poljubno premikamo, ni mogoče vnaprej predvideti njegove lokacije. Zato jo moramo ves čas preverjati. V primerih, ko moramo ves čas preverjati, ali se je nekaj zgodilo, uporabimo zanko, ki se izvaja “za vedno”.



1. What are the possible locations of the chameleon on the screen? We noticed there are five:
   1. he is entirely on the blue color
   2. he is entirely on the sandy color
   3. he is entirely on the green color
   4. he is partly on the blue color and partly on the sandy color
   5. he is partly on the sandy color and partly on the green color
2. In *Snap!* we have a special block that tells us what color the object is currently touching. We can find it in the Sensing group: .
3. This block gives us data about whether it’s: *true* or *false* that object is touching a specified color. We can specify a color by clicking on a little square and then select a color. We can choose it from a color palette or by clicking on a part of the screen with desired color.

Below is the example showing us how to get data: *true* or *false* from a sensing block if object is touching the sandy color:

|  |  |  |
| --- | --- | --- |
|  |  |  |

1. Block  is not a command block. It represents a logical expression that can be put into a control block. Control blocks execute other blocks (that we put in their body) only if the value of the logical expression (in their head) is *true* and don’t do anything if the value is *false.* Examples of control blocks are: if block, repeat until loop and event “When”. In our case we will have to use if control block.
2. Combine blocks below to check if the chameleon is touching the sandy color. In that case switch to a sandy chameleon look:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

1. Now you can use similar blocks and then combine all of them together to also check the other two conditions: if the chameleon is touching blue and green color.
2. **Remember!** If there are multiple if blocks with fulfilled condition in a sequence, only the commands of the last one will have the effect in the game. Use that knowledge when creating the code.
3. Now we have to take into consideration other two cases that can happen in game, that is when chameleon is touching two colors of the background at the same time. If you want to check if multiple conditions are true at the same time you have to combine them using a logical operator “and”. You can find logical operators in the “Operators” group.



1. Combine two blocks for sensing color: 2x , logical operator: and , if block:  and block for switching costumes:  to implement detecting those additional cases.
2. Complete the game with the use of block  to program a chameleon to say on which part of the screen is he currently positioned.

Chameleon summer vacation: <https://snap.berkeley.edu/project?user=zapusek&project=chameleon>