**Learning Scenario 11 – Guessing the number of cats in a shelter**

Task: Cat shelter keeper Martha wants you to guess the exact number of cats that she has in her shelter. The number can be anywhere between 1 and 100. When the player types the number she answers if current input number is less, more or equal to the right number of cats.

1. Open the template file:

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

There is an image for the background and the main character - cat shelter keeper.

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

1. First we have to randomly choose a number of cats in a shelter. The number must be selected from the interval from 1 to 100. We will need that value later in a game for comparing it to the player’s answers, so it has to be stored in a variable.

Create a new variable *number\_of\_cats* and assign it a random value from 1 to 100. This number will represent a number of cats in a shelter.

Use the blocks:  and .

1. Think about: 1) which actions will be repeated in a game and 2) how many times we will have to repeat those actions. Can we predict in advance how many guesses will the player have to make in order to figure out the correct number?

1. The following actions will be repeated in a game:
   1. Player will enter a number. We can get an input from a player with the use of  block.
   2. We have to check if the number is:
      1. Greater,

Combine the blocks below to find out if the entered number is greater than the value stored in a variable *numbe\_of\_cats:*

, ,  and .

* + 1. Smaller,

Combine the blocks below to find out if the entered number is greater than the value stored in a variable *numbe\_of\_cats:*

, ,  and .

* + 1. Same?

To detect if the player answered correctly we will use a small “trick” that will be explained in the next chapter.

1. How many times will the code from 4) repeat? Well..until the player won’t guess the right number. Think about how we cannot predict how many tries a player will need to guess the right number. In such situations where we have to repeatedly execute the same actions until a sentinel event occurs (in our case, the sentinel event occurs when the correct number is entered), we use a repeat until loop.



“Repeat until loop” will repeat all the blocks in its body until the condition in its head is true. If the condition is evaluated as false it will go into another iteration. When the “repeat until loop” exits, blocks that are placed below will be executed.

In our case we will have to ask a player to input her guess and compare that value to the value stored in the number\_of\_catsvariable. We will have to stop when the player’s input will be equal to the number of cats, if not, we will have to do it again.

Define the condition for checking if the input is equal to the number of cats using these blocks: ,  and .

1. **Let’s summarize:** What will happen if the entered number will be different than the number of cats? Loop will go into its next iteration.
2. **Let’s summarize:** What will happen if the entered number will be equal to the number of cats? The loop will stop and the blocks placed below the loop will be executed.
3. This is the “trick” we were talking about earlier and that enables us to detect the right answer without checking the condition explicitly inside the loop.

Let’s see what will happen when the player enters the right number. The two conditions inside the loop that check if the value if greater or smaller will not be satisfied and the loop will go check the condition in its head. This condition will be true, so the loop wont go into the next iteration and will stop executing. The program will start to execute the blocks placed below the loop.

1. If we know that “repeat until loop” is behaving like this, we can use that knowledge to our advantage. We can conclude that if the blocks placed below the loop are executing, the player must have guessed the right number. Then we can congratulate the player.
2. In Snap! we can show or hide the value of the variable to the payer. This can be done by clicking on the checkbox next to the variable name: .

Consider if it is good to have the value of the *number\_of\_cats* variable visible to the player?

1. You can upgrade the game using these suggestions:
   1. Count the guesses.
   2. In the beginning ask the player to enter her name and greet him. Use her name when giving feedback or asking to input the next try.
   3. Give feedback that will depend on the player's success. If the player guesses the correct number in five or less tries, give her a cat as a reward. If she guess the correct number on the first try, provide a special feedback..
   4. Add your own ideas and upgrade the game to your likings.