"Catch me" 🎮

As a group of experts it's now about to program a tag-game. For this purpose, each of you must bring in their previously learned expert knowledge! It is the aim, that a character of your choice hunts and catches another character of your choice. The following points will help you to implement this task. Of course, you can set your creativity free.

The four experts:
In order to be able to better estimate which expert can use his knowledge at any particular point, the respective experts are addressed explicitly during the course of the work sheet.

- Expert MD: Motion & Drawing
- Expert SC: Sound & Communication
- Expert FV: Feelings & Variables
- Expert GA: Graphic & Appearance

Basic Structure of the Game:

1. Create a game character, which is supposed to be the catcher later on (e.g., the 🦊). Adjust the size of the figure to your background and program it so that you can control it with the arrow keys (Events and Motion). Keep in mind that the catcher must be able to appear and submerge.

   Caution: Use 🔄 in order to adjust the direction of the catcher.

2. Create another game character which is supposed to be caught in the game (e.g., 🍂). Adjust the size of the characters to your background.

3. Choose three different stage sets, you want to use in your game (2 for the game process, 1 for the end).

4. Create a variable for the score from Data.

5. Add a stopwatch from Sensing ( Optional ).
"Catch me"

**Start of the Game:**
- The game should be started by clicking on the 🚀 (Events).
- Stopwatch and score must be reset when the game is started. Use the following puzzle pieces from Sensing and Data (also think about the "forever puzzle-piece"):
  ![Reset Timer]
  ![Set Score to 0]
- The stage should show the first stage (Appearance).
- The catcher 🐿️ should always be at the same position (x-/ y-coordinate) when the game starts. For this purpose, you could choose the centre of the stage and set its position x = 0 and y = 0 (Motion).
- Also consider a fixed starting position for the character to be caught 🐻 (Motion).
- When starting the game, a nice sound should resonate (Sound).

**Tag:**
If the catcher 🐿️ has caught (or touched) another game character, the following ought to happen:

- The caught figure 🐱‍🐉 should descend and reappear on the stage at a random location (Motion).
  **Use:**
  ![Pick Random 1 to 10]
  In order to create a random number (Operators).
- The caught character 🐮 should send the message to every character that he was caught (Events).
  **Use:**
  ![When I Receive Caught]
  For the following three issues:
- A sound should be played (Sound).
- The score should be increased by 1 (Data).
- After the figure has been captured, the stage should change as well (Appearance). Just choose two of your three stage sets!
End of the Game:

- The game will end after 30 seconds, as soon as the timer displays more than 30 seconds. **For this purpose use:**

  ![Timer](image)

**Bonus:**

**MD**
- The character that is supposed to get caught should return to its original position (**Motion**). Then he shall ask your name (**Sensing**).

**FV**
- Besides, the character is expected to tell you, how often he has been caught (**Appearance**). You will need five puzzle pieces for this purpose.

**Combine them in the right order.**

![Puzzle Pieces](image)

**SC**
- Again the character to be caught should send a message to announce the end of the game (**Events**).

  **Use the following puzzle piece again:**

  ![Message](image)

**SC**
- The end of the game is also initiated by a nice sound (**Sound**).

**GA**
- At the end of the game, the stage is supposed to be a ceremonial stage (your third one) (**Appearance**).
Congratulations!
You have reached your goal!