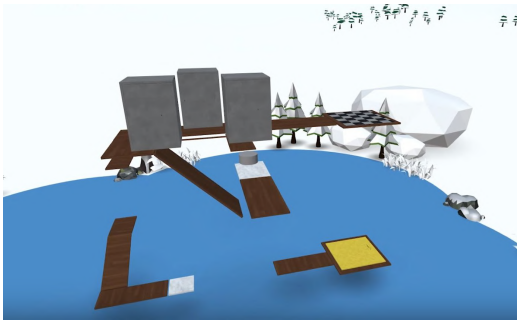
Parkour game creation Lesson plan

Created by the CoSpaces team



Education level: From middle school

Subject: STEM, computer science

Format: Individual or in groups

Duration: Approx. 3 hours

Introduction and lesson objectives:

Through this lesson, students will learn the basics of designing and programming a parkour game from scratch, using CoSpaces Edu and its visual block-based coding language CoBlocks. This activity can be done in groups of students or individually.

In this basic parkour game, the player will have 2 obstacles to pass, without touching the ground. Whenever the player falls, they'll be taken back to the parkour's start position and lose a life. If the maximum amount of lives has been reached, it's a game over. But if the player manages to reach the goal before that, the game is won!

Students will first build the parkour and then learn to program all of this!

Learning goals and student benefits:

- Develop 3D creation skills
- Develop spatial thinking
- Foster creativity
- Learn game design logic
- Practice computational thinking
- Learn coding skills

Activity example:

1. As an introduction, show your students the Games tutorial giving them an overview of what they'll create.
2. Ask students to think about how the player will go through their parkour.
3. Let your students follow the step-by-step creation guide attached to build and program their very own parkour game!

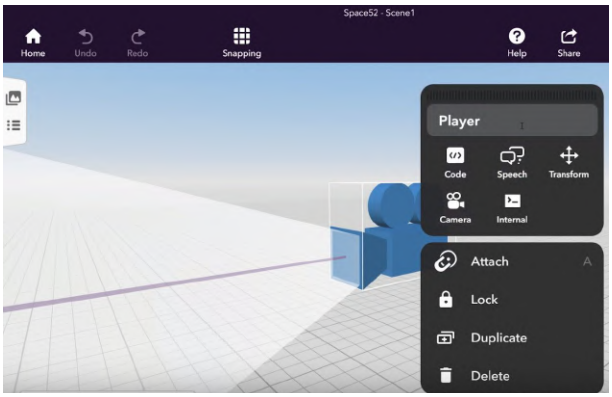
Extension idea:

Let your students play their classmates' parkour games and exchange feedback.

Assessment and evaluation suggestions:

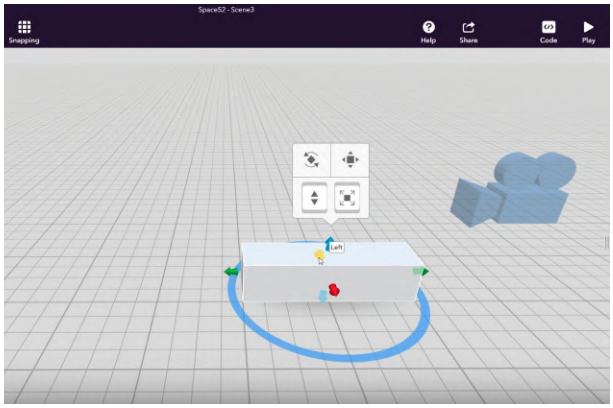
- Have your students managed to create a parkour game in CoSpaces Edu?
- Were your students able to program their game with CoBlocks?
- Is the player able to go through the game?
- Does the game have a clear outcome (game over, winning, score reach, etc.)?
- Does your students' work reflect a good understanding of the different creation tools available in CoSpaces Edu?
- Does your students' work reflect a good understanding of basic coding?

Creation guide



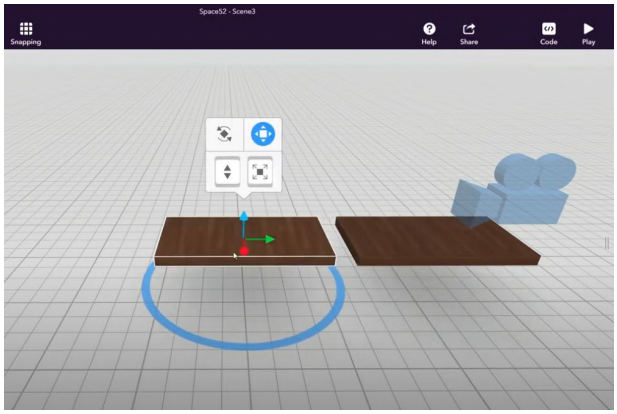
This parkour game is played in the first person perspective, meaning that the camera's view is what the player sees.

To help you conceive this when you create your game, rename your camera to **"Player"**.



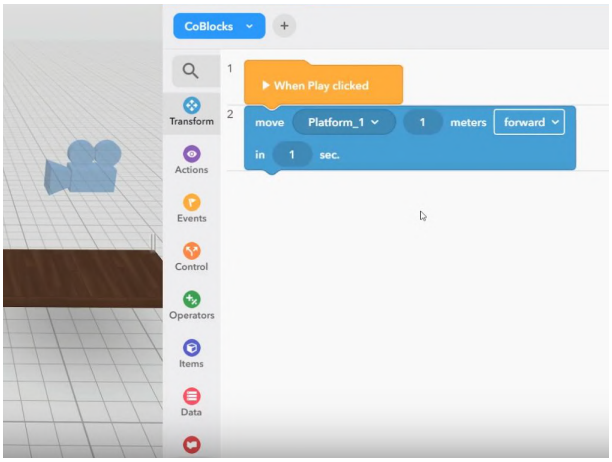
To build your parkour, use the Cuboid available under Building in the Library.

This is a great **building block** to create a track for the player to walk on.



Place a small platform at the end of this track that will move back and forth.

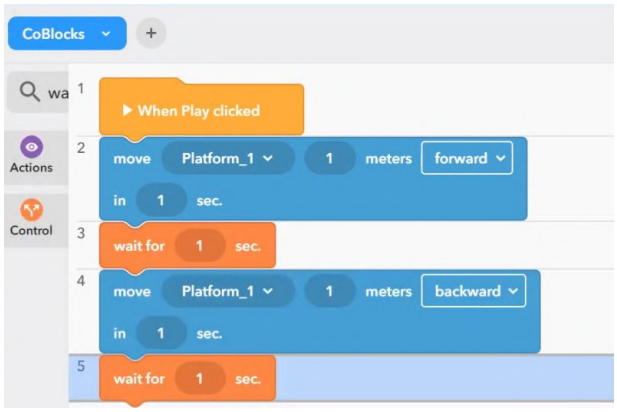
Name it and enable its Use in CoBlocks.



Create a **CoBlocks** script and use the **move** CoBlock to define the platform's movement.

To quickly review your code, disable the player-camera while building. Double or right-click the camera and disable **Use as main camera**.

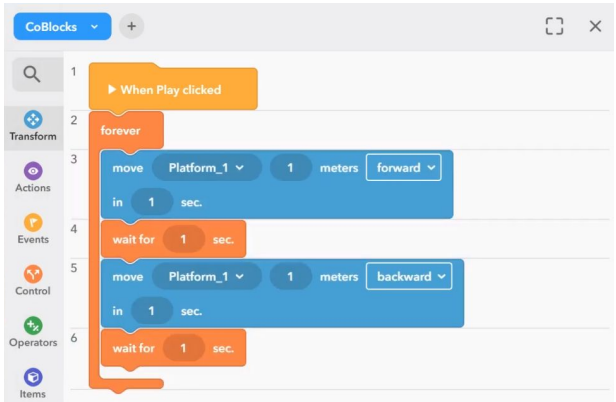
Hit **Play** to test whether your platform is moving correctly!



Right now, it's just moving in one direction and not coming back.

Duplicate the **move** CoBlock and choose the **opposite direction**.

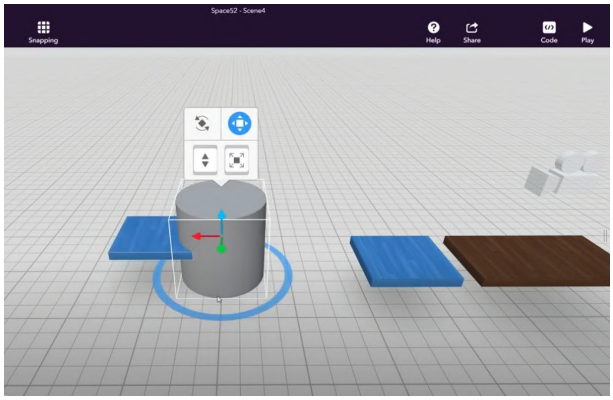
Add a **wait** CoBlock in-between and afterwards so that the platform stays still for a bit, giving the player a chance to get on and off.



You'll want this code to repeat for the platform to keep moving.

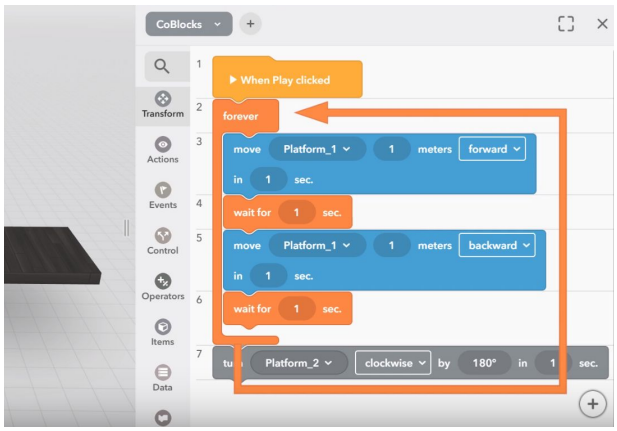
Placing it inside a **forever** CoBlock will do the trick!

The platform will now move back and forth endlessly.



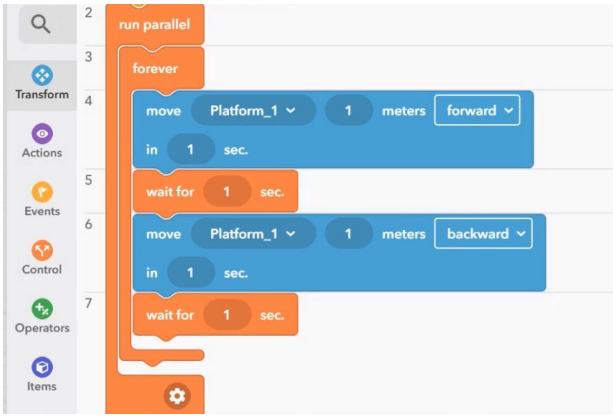
To build the second obstacle, let's use a **Cylinder** from the **Library** and **Attach** a **Cuboid** to it.

Get a **turn** CoBlock this time to make the cylinder turn.

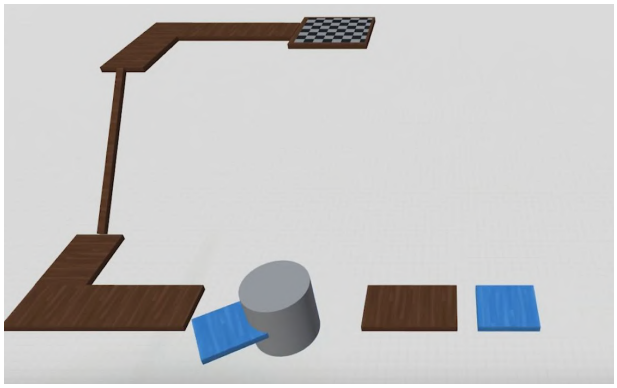


See how the platform is moving but the cylinder isn't?

That's because the forever-loop of the first obstacle never ends.

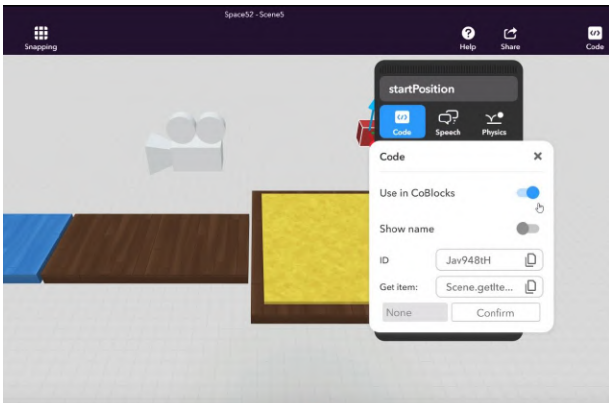


You'll need the **run parallel** CoBlock to let different sections of code get executed simultaneously.



Place the obstacles however you like and add tracks to connect them.

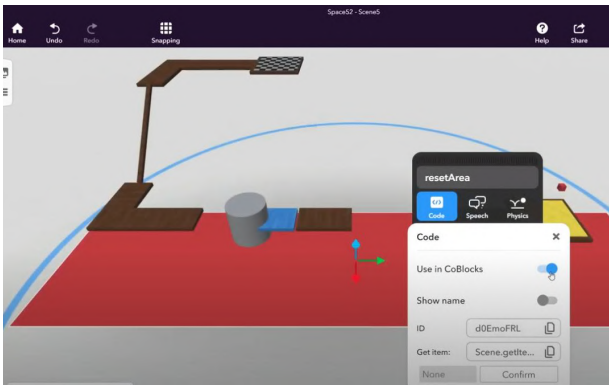
Get creative with your parkour and build more obstacles if you want to!



Time to program this further to turn it into a real game!

Whenever the player falls down and touches the ground, they should be sent back to the start.

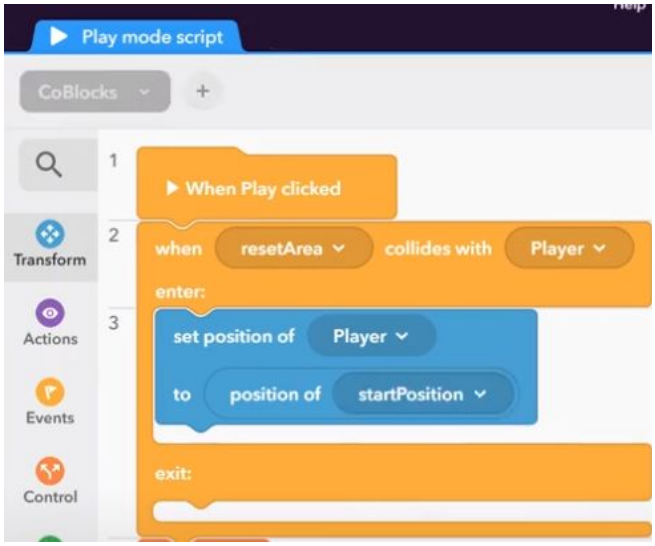
Place a small object like a cube to define the **start position** and name it **“startPosition”**.



Use a **building block** to define a reset-zone and name it **“resetArea”**.

This will send the player back to the start position when they touch it.

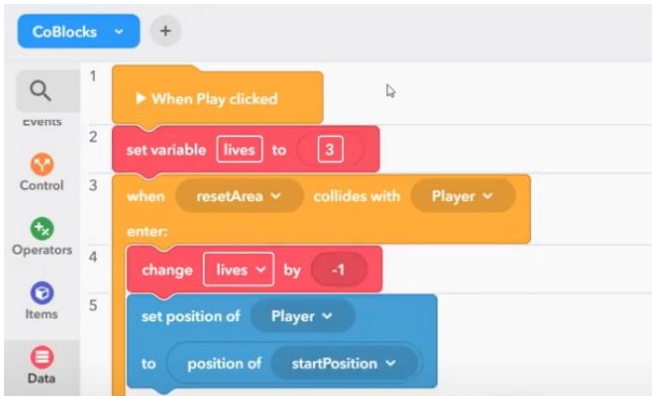
Don’t forget to enable its Use in CoBlocks!



Set a when **resetArea** collides with **Player** CoBlock above the other CoBlocks to test what happens when they collide.

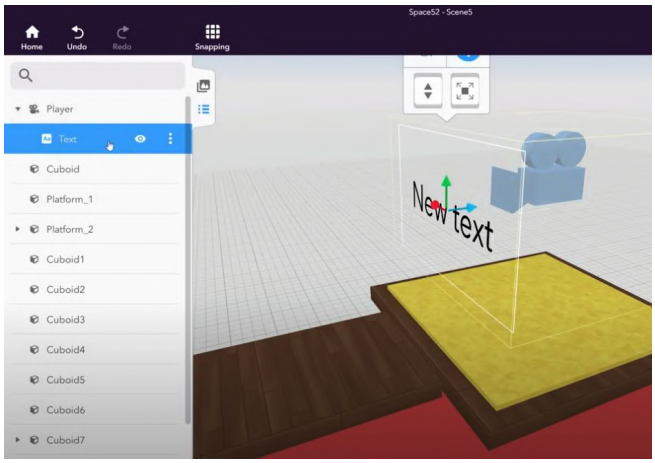
Using the set position and position of CoBlocks, program that whenever they collide, the player's position is set to **"startPosition"**.

Enable the player-camera again and jump onto the **resetArea** to test whether it's working!



To keep count of the player's lives, create a new **variable** named **"lives"** and insert a number for the initial **number of lives.**

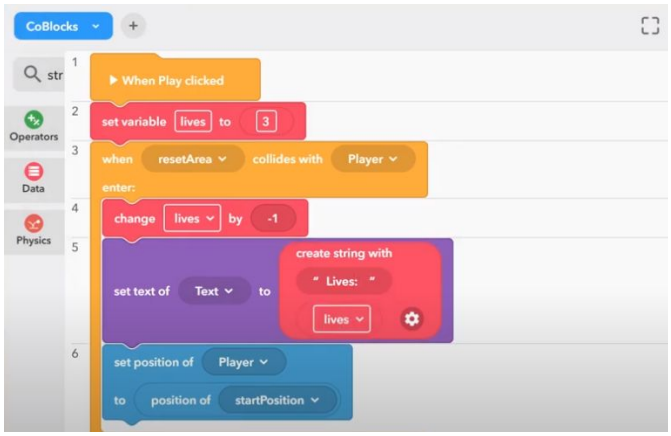
Change the **variable** by **-1** whenever the player collides with the ground, losing a life.



It would be nice to show the player how many lives they have left.

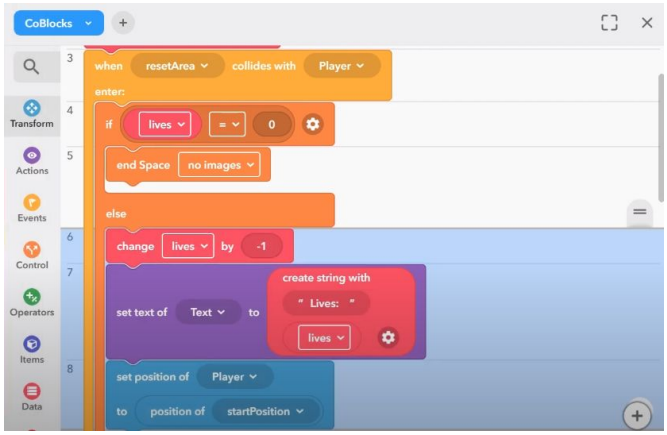
Let's use the Text available under Building to display the player's lives, and attach it to the camera so that it's always shown.

To attach it, open the hierarchy and drag and drop the text onto the player-camera.



Place the text in a nice position in front of the camera and add a **setText** CoBlock, just after a life has been lost and the number of lives therefore modified.

Type in **“lives”** and then insert the **lives variable** to show the amount of lives left. Or even better: Place a **create string with** CoBlock!

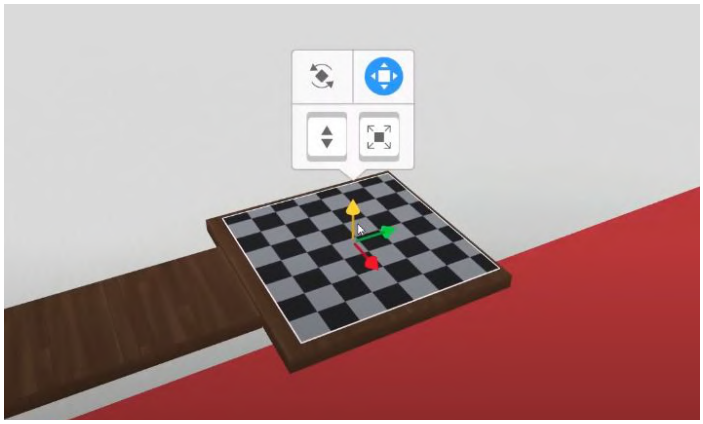


Whenever the player collides with the ground, check whether they have any lives left. The **if** CoBlock lets you test your variables.

If the **number of lives** equals **0**, it means that the player has **lost** and the game is over. But otherwise, the player has some lives left, they lose one life and can keep trying.

You may notice that the text only appears after the player fell for the first time.

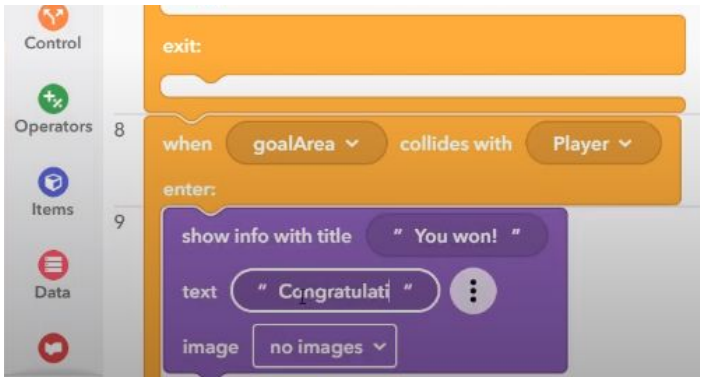
Let’s show the number of lives as soon as the game has started, by copying the **setText** CoBlock and moving it just after the declaration of the variable.



Just one thing left: Let's define the **goal area** that the player has to reach to win the game.

Place a **building block** wherever you want the goal to be and name it **"goalArea"**.

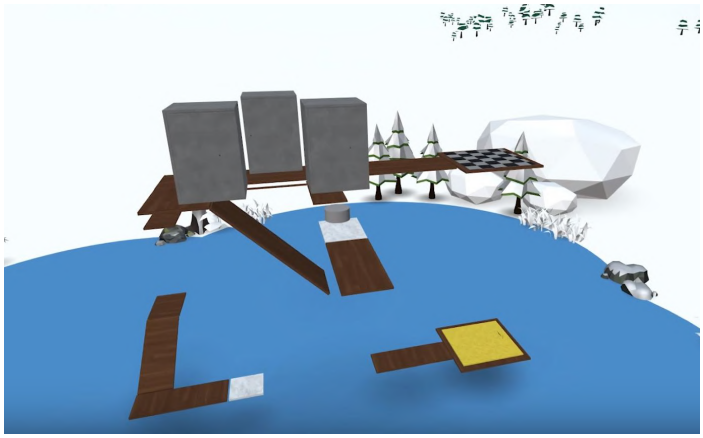
As soon as the player collides with it, they win the game.



You can show an **info panel** to inform the player of their victory.

Pay attention to the order of the when collides with CoBlocks!

You can view the whole code for building a parkour game like this on the next pages.

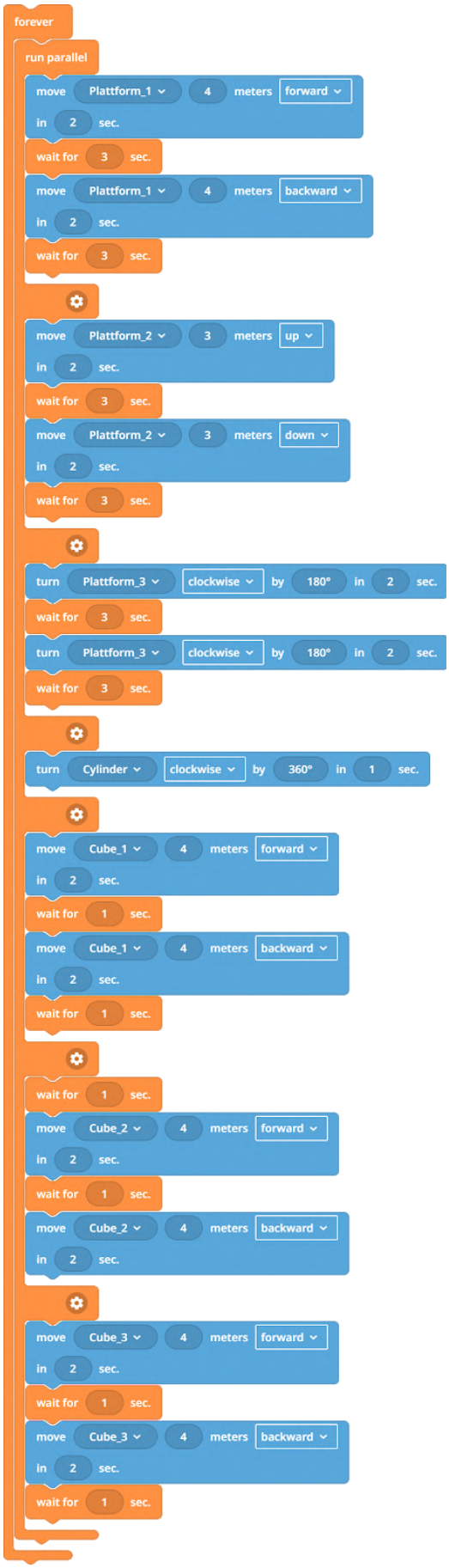


Finally, add any decorations you like with objects from the Library and give your parkour an interesting theme!

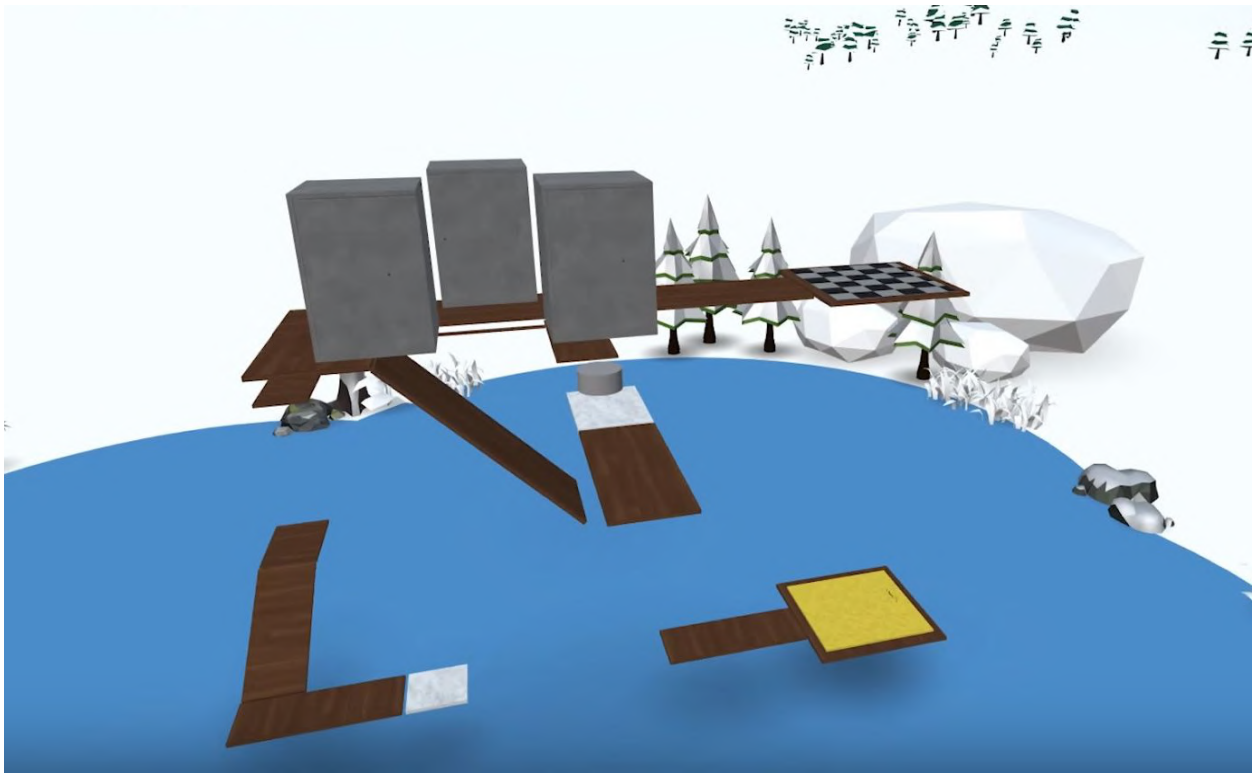

PRO


When Play clicked
set variable lifes to 3
set text of LifesText to
create string with
" Lives: "
lifes
when Player collides with Ground
enter:
change lifes by -1
set text of LifesText to
create string with
" Lives: "
lifes
set position of Player
to position of startPosition
exit:
when Goal collides with Player
enter:
show info panel with title " You won! "
text " Text "
image no images
wait for 3 sec.
end CoSpace no images
exit:


PRO


forever
run parallel
move Plattform_1 4 meters forward in 2 sec.
wait for 3 sec.
move Plattform_1 4 meters backward in 2 sec.
wait for 3 sec.
move Plattform_2 3 meters up in 2 sec.
wait for 3 sec.
move Plattform_2 3 meters down in 2 sec.
wait for 3 sec.
turn Plattform_3 clockwise by 180° in 2 sec.
wait for 3 sec.
turn Plattform_3 clockwise by 180° in 2 sec.
wait for 3 sec.
turn Cylinder clockwise by 360° in 1 sec.
move Cube_1 4 meters forward in 2 sec.
wait for 1 sec.
move Cube_1 4 meters backward in 2 sec.
wait for 1 sec.
wait for 1 sec.
move Cube_2 4 meters forward in 2 sec.
wait for 1 sec.
move Cube_2 4 meters backward in 2 sec.
move Cube_3 4 meters forward in 2 sec.
wait for 1 sec.
move Cube_3 4 meters backward in 2 sec.
wait for 1 sec.

Example CoSpace



Parkour game

edu.cospaces.io/YYQ-DFN