



Code a Pac-Man in JavaScript

Beginner Level • 13-16 y/o • 1h30

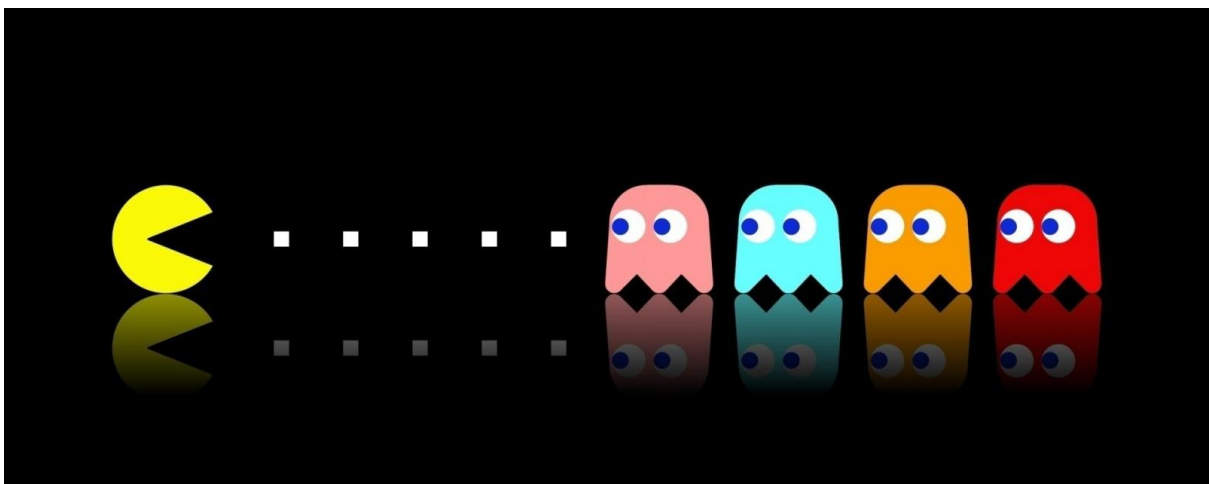
1. Presentation of the game



Pac-Man is a game released in Japan in 1980 that involves moving Pac-Man inside a Labyrinth in order to make him eat all the “**pacgums**” without being touched by ghosts.

Some elements of **Pac-Man**'s code have been lost . . . It's up to you to complete the game code to get it working again!

In this exercise, you will learn how to program loops, conditions in **JavaScript**, and a web game's overall functioning.



2. Tools & Resources

2.1 Resources

To get started, go to <https://repl.it/@EmmaEpitech/PacMan>.

The screen is split into several parts: the file part, on the left: this is where you will code the game. And the rendered part, on the right: this is where you will see your progress and can test your game.

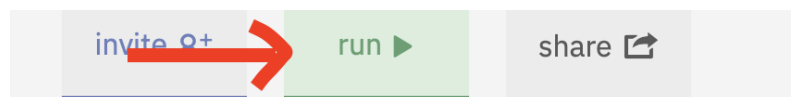
2.2 Getting started with the platform

On the left side of the screen, you can see a list of files.

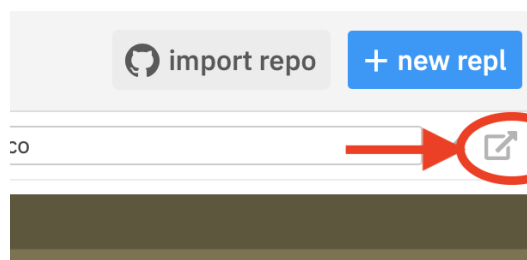
HTML files, images, and **JavaScript** files.

Today, you will only need to edit the "**game.js**" file. However, you can go and see what happens in the other files if you wish.

To run your code, press the green arrow "**RUN**" at the top of the screen.




If your screen is too small to display the entire game, you can open it in a new tab by clicking on the following button:




2.3 Game.js

Several functions are present in the **game.js** file. Your role will be to complete them.


```
function initialize(map)
```

-  The **initialize** function is one of the very first ones called in the game. Anything you put in it will therefore be executed at the very start of the game.

```
function gameLoop(game, blockType, blockPosition)
```


-  The **gameLoop** function will allow advancing in the game as Pac-Man advances.

```
function onGhostCollision(ghost)
```

-  The **onGhostCollision** function is called for each collision between Pac-Man and a Ghost.


3. Discovery of the game

3.1 Eat Pac-Gums

-  If you run the game, you can see that **PacMan** is moving around but not interacting with other game elements!
The first step will therefore be to allow him to eat the Pac-Gums.

For that, we are going to modify the type of the block when Pac-Man moves. Adds the following line to the start of the **gameLoop** function.

```
game.map.setBlock(blockPosition, Pacman.EMPTY);
```

-  Launch the game to see what happens!

3.2 Add special Pac-Gums



In Pac-Man, ghosts can have two states. A “dangerous” state, where they can eat Pac-Man, and a “vulnerable” state where Pac-Man can eat them.

The special Pac-Gums allow the ghosts to pass into this second state.

Adds the following lines to the **initialize** function. This will allow replacing a classic Pac-Gum by a special Pac-Gum.

```
addBonusGum(map, 1, 1);
addBonusGum(map, 17, 1);
addBonusGum(map, 1, 16);
addBonusGum(map, 17, 16);
```



Try changing the values to see what happens!



Launch the game to see the special Pac-Gums!

3.3 Change the state of the ghosts



Going back to the **gameLoop** function. Using a loop, we'll update the state of each ghost when a special Pac-Gum is eaten.

First, add the following condition after the gameLoop function:

```
if (blockType == Pacman.BONUS_GUM) {
}
```

Then add the following loop inside the condition:

```
for (i = 0; i < game.ghosts.length; i += 1) {
}
```

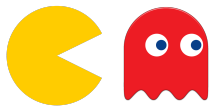
Inside the loop, add the following line to put the ghosts in the state they can be eaten:

```
game.ghosts[i].makeEatable();
```



Launch the game and try to eat a special Pac-Gum!


3.4 Collision with ghosts



As we saw earlier, there are two different states for the ghosts. We will therefore manage the cases of collisions according to the state in which the ghosts are found.


In the **onGhostCollision** function, add the following condition, to eat vulnerable ghosts:


```
if (ghost.isVulnerable()) {  
    ghost.eat();  
    return false;  
}
```

 *Returning **false** tells the game engine that a collision has occurred with a **vulnerable** ghost.*

Then add the condition below:

```
else {  
    return true;  
}
```

 *Returning **true** tells the game engine that a collision has occurred with a **dangerous** ghost.*

 Launch the game and try to eat a ghost!

3.5 Detect when Pac-Man won



The last step, detect the end of the game. A game of Pac-Man is won when all Pac-Gums have been eaten.

At the end of the **gameLoop** function, add the following condition:

```
if (game.user.eaten == NB_GUM) {  
    game.completedLevel();  
}
```

 *Press **RUN** and try to win a game!*

Good luck to you!

4. Bonus

If you have completed all the previous steps, first of all, **Well Done!**



We have prepared a bonus step that you can do to improve the game or make it more difficult: counting points.

As Pac-Man walks forward and eats Pac-Gums, add points to the point counter.

To do this, create a new function called **updateScore** which will take **game** and **blockType** parameters:

```
function updateScore(game, blockType) {  
  
}
```

Add two conditions to test the type of Pac-Gum eaten (normal or special)

```
if (blockType == Pacman.GUM) {  
  
}  
  
if (blockType == Pacman.BONUS_GUM) {  
  
}
```

Then, under each of the above conditions, calls the **addScore** function, with the number of points to add as a parameter: 10 for normal Pac-Gums, and 50 for special Pac-Gums.

```
game.user.addScore(10);
```

Last step: call your new function in the **gameLoop** function

```
updateScore(game, blockType);
```

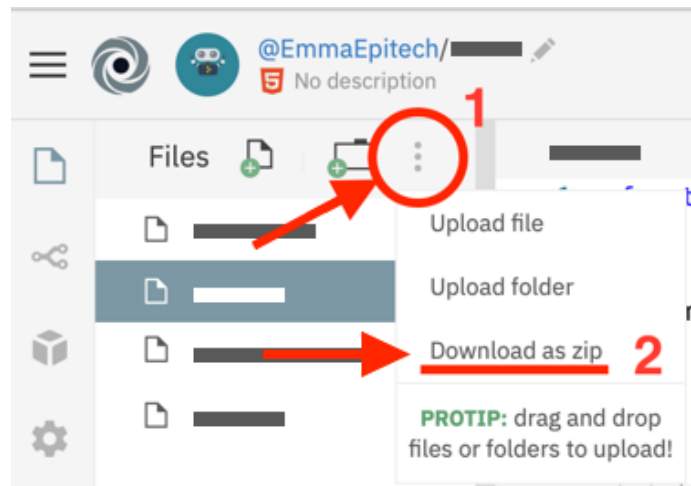


Relaunch the game and try to earn as many points as possible!

5. Save your program



You can save your game on your computer to edit or play it later.
To do this, click on **“Download as zip.”**



6. Few useful links

To learn JavaScript:

→ <https://www.w3schools.com/js>

To redo the exercise:

→ <https://repl.it/@EmmaEpitech/PacMan>

To see our other exercises:

→ <https://repl.it/@EmmaEpitech>

For more information on our activities:

→ <https://www.e-mma.org>