

PacMan Time 3D

Ficarola Filippo 1815827

September 2021

Contents

1	Introduction	3
1.1	Rules and Commands	3
2	Menu Scene	4
3	Models	5
3.1	Pac-Man model	5
3.2	Ghost model	5
4	Environment	5
4.1	Skybox, Ground & Walls	5
4.2	Lights	6
5	Animations, Movements & Collisions	7
5.1	Loop Animations and Movements	7
5.2	Super Pac-Man Mode	8

1 Introduction

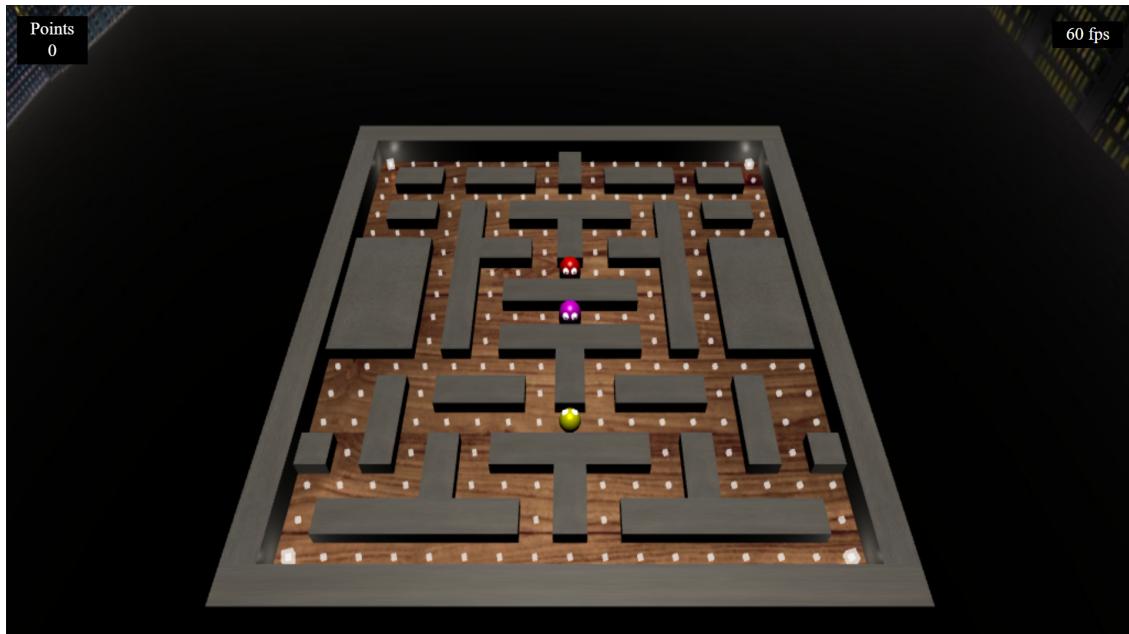
The project is a simple game and 3D rendering engine written in JavaScript and it is entirely realized by using the Babylon.js library. The Babylon documentation was really helpful for the implementation of this project.

1.1 Rules and Commands

Everybody knows the legendary Pac-Man, a maze action game developed and released by Namco for arcades in 1980 and played for years. So this project tries to follow the original Pac-man, but with some differences.

So as soon as we press the play button we can see the maze where the game takes place. This maze is full of small cubes that should be eaten by our Pac-Man to reach a total of 200 points, necessary to win the match, but among them we can even see two ghosts, the antagonists of our Pac-Man. These ghosts move randomly in the maze and if they touch our Pac-Man the match is lost.

Even if the ghosts are lethal for Pac-Man is possible to fight them thanks to the big bright cubes that we can see at the four corners of the maze. These big cubes gives to our Pac-Man 8 seconds of immunity, and allows him to knock out the enemies for a while if touched.



The knocked out ghost will disappear for 10 seconds and then come back to its last position to start again moving randomly among the maze.

In addition, we can see on the left-top side of the screen a box that keeps count of the points we are accumulating during the game (small cubes gives 1 point while the big ones gives us 8 points) and on the right-top side the current fps.

To move Pac-Man we can use the standard keys W-A-S-D (It is suggested to press one button at time since diagonal movements will probably cause collision with walls and stop the character).

2 Menu Scene

The Menu scene is a simple scene where on the top we can see the title of the game, a brief explanation of the game on the bottom and a purple button on the right that gives access to the game. This is the scene where the game will return if we win or lose. In the middle we can see the animated models of Pac-Man and a ghost in a black background and a light that allow us to see the models. There is also an `ArcRotateCamera` that allows us to see the models from different points of view.



3 Models

The models are not imported but both Pac-Man and Ghosts models were built using hierarchical relationships that link some basic Babylon meshes (like spheres or cylinders).

3.1 Pac-Man model

The Pac-Man model that we can see in the Menu scene was built using two hemispheres (spheres with the component slice=0.5) with the same center and with the upper one initially rotated of π with respect to the other one. The eye model (that is the same for the ghost, but with different positions) is formed by two spheres, a big white one and a second one smaller of a black color for the pupil.

In this model the hierarchy is the following: The Bottom hemisphere is the parent of the Upper hemisphere. The Upper hemisphere is the parent of the two Eyes and finally each Pupil is the children of the corresponding Eye. This hierarchy was useful to create the Mouth animation.

3.2 Ghost model

The Ghost model is formed by a Cylinder (body) that is the parent of an hemisphere (head) positioned on the top of the body. Like the Pac-Man model the hemisphere is the parent of the two Eyes that are respectively parents of each Pupil.

4 Environment

4.1 Skybox, Ground & Walls

To create the illusion of a realistic ambient the whole scene is surrounded by a skybox and i applied to it a skybox city image. The ground is positioned in the center of the skybox and it is realized using a texture that gives the illusion of a wooden floor. To allow the user to explore the entire map with the mouse i used an `ArcRotateCamera` in the game, even if for a better user experience during the match it should be leaved to its initial position.

The floor is surrounded by a perimeter of four great walls which delimit the maze. Then we have smaller walls which create the maze

inside the map. All these walls are rectangular of different sizes for a total of 34 walls. To this wall i attached a texture that gives the illusion of cement.

4.2 Lights

There are 5 lights in game scene:

- One Directional Light positioned on the top of the maze necessary to illuminate the ground.
- Four additional Point Lights positioned in the same place where the big points stand. They have a range of 200 so can illuminate objects only if they are close enough and they disappear whenever our Pac-Man eats the associated big point.

Since there are five lights and the Babylon standard for every material allows them to manage only four of them the property "maxSimultaneousLights" was incremented to 6, so each material (Pac-Man material, Walls material etc.) can manage all of them.



5 Animations, Movements & Collisions

5.1 Loop Animations and Movements

Since there are no imported models, even the animations were created by me, and we have different types of animations, some of them constant during the game and others that start only when certain events happen.

- **Pac-Man Mouth Animation :** It is present in both Menu scene and Game scene, and it is a looped (which repeats after a certain frame rate) rotation along the X axe (from π to $3/4\pi$ of the Upper hemisphere of our Pac-Man. Because of the hierarchical relationships, Eyes and Pupils move with upper hemisphere.
- **Fluctuate Animation :** It is common for Ghosts, Small points and Big points and it simulate a fluctuating behavior necessary especially for Ghosts. It is a loop animation that affects the Position along the Y axe and brings the objects form a certain height to another one and repeats.
- **Rotating point Animation :** We can see even that the points in the maze constantly rotate along the Y axe at the same frame rate of their fluctuation.
- **Ghost Eyes Animation :** It is a simple rotation along the Y axe which affects the eyes (and so the Pupils) of the Ghosts moving them in loop from $1/4\pi$ to $-1/4\pi$.
- **Pac-Man Movement:** This is created using an inputMap that captures the actions of the user on the keys W-A-S-D to move, at a certain velocity, the Bottom Hemisphere of our Pac-Man (so the entire Pac-Man because of the hierarchy) respectively Up, Left, Down and Right. In addiction, each time we press one of these buttons Pac-Man rotates along the Y axe to the corresponding direction of the movement.

Obviously Pac-Man should not pass trough the walls, so to manage the collisions in this case, using the function `IntersectMesh()` (in a for cycle which scans an array where all the walls are stored) each time our character touches a wall if key is pressed an equal

velocity to the Pac-Man velocity, with same direction but opposite verse, is applied to Pac-Man. So even if the user is pressing a movement key it does not move in this direction if there is a wall.

- **Ghosts Movement :** The Ghosts Movement was slightly more complicated since they have to move free in the maze. They start from an initial position and, when the game is ready a random function is called to decide the first direction of the Ghost. This function creates a random variable that can assume values from 0 to 1, so, since the possible directions are four there are four different `if` (every `if` catches an interval of 0.25). So for example if the variable assume a value of 0.2 the ghost will go along Z axe with positive velocity, or if it assume a value of 0.55 the Ghost will go along the X axe with negative velocity and so on.

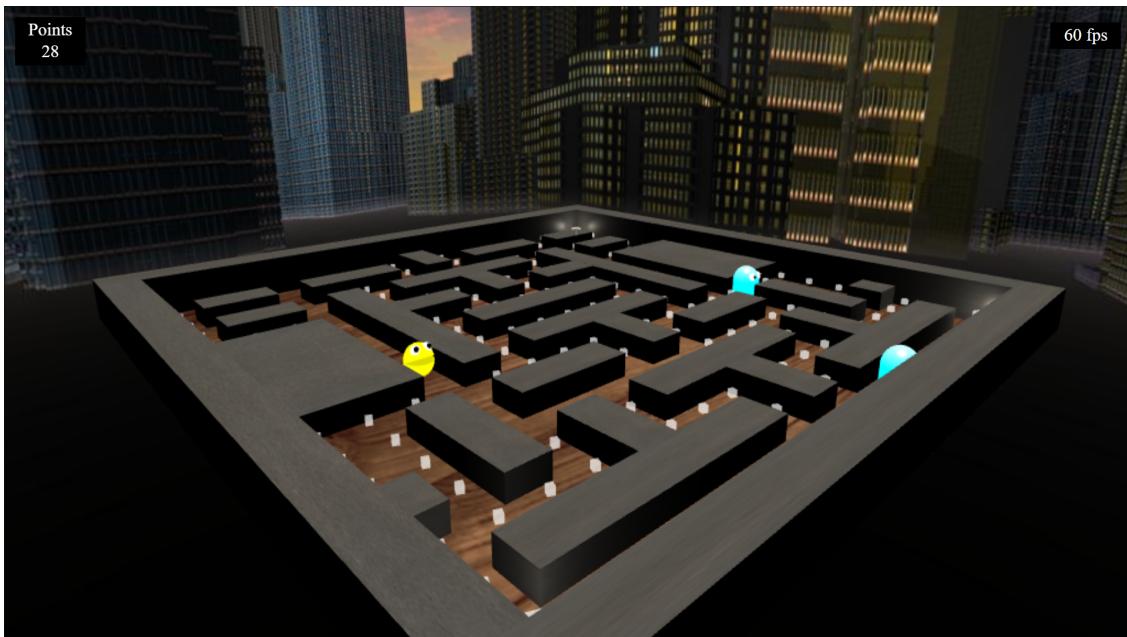
This function is even called each time the Ghost hits a wall. In this case its position is translated backward (with respect to its current velocity, to avoid deeper intersections with the walls) and a new direction is decided from the random function until it moves along a new free direction.

5.2 Super Pac-Man Mode

We have already talked about the four Big Points in the maze and now we will discuss their effect on the game.

In a normal scenario Pac-Man is slower than its enemies and if it hits one of them the `killPacman` animation starts. This animation will stop both Ghosts in the maze and will move our Pac-Man under the ground with a translation along the Y axe. A time out of three seconds will show us the text "YOU LOSE" and then we will return to the Menu Scene.

Instead, if we reach one of the four Big Points things changes. In this case the texture of Pac-Man will become brighter thanks to a the yellow component of the `emissiveColor`, while the Ghosts will become of a bright Teal and the light present inside the eaten Big point will be switched off.



Now, for a period of 8 seconds Pac-Man's velocity is increased while the Ghost's velocity is decreased so it can easily follow their enemies and eat them. In fact when Pac-Man hits a Ghost during "Super Pac-Man" the animation `killGhost` will start and the Ghost is stopped and sent under the ground for a perion of 10 seconds, with a variation of position along the Y axe. After that it will come back on the maze, the random function will be called again and it will start moving again. We can see that the Super Pac-Man effect disappear when textures of the characters will become normal.