Pac Man Prototype

By Vincent Hubau

For Funcom technical test

Documentation:

You begin on a menu asking you if you want to begin playing or quit the application. When the game begins, you can move with the arrows.

You have to eat every puck on the map, including the power-ups. Ghosts will chase you and try to kill you. As soon as you get hit, you will get teleported to your spawn point and lose a life.

You can teleport yourself from the left “door” on the left of the map as well as the right side. You will get teleported to the other side of the map.

If you don’t have any life left, you will have a Game Over. Press ‘R’ to restart or ‘M’ to go back to the menu. If you eat every puck, you will get the same menu, but saying you’ve won.

Functionalities:

There is a basic menu to begin playing. When you lose or finish the level, you can restart it or go back to the main menu.

This Pac-Man is pretty simple. It has a single level based on the original game. Your goal is to eat every single puck.

When you begin the game, you have 3 lives and you have the ability to eat pucks. Every puck (white) is 100 points and every power-up (Yellow/Orange) is 250 points. Every 20 seconds, a bonus puck (red) appears at your original spawn position and gives you 500 points. You have 10 seconds to get it, if you don’t, it will disappear and you another one will appear 20 seconds later.

When you make 10000 points without dying, you get a life, and you can only have a maximum of 5 lives. When you die, you go back to your original spawn point.

The ghosts are chasing you at every time and you can use the teleport doors at the extreme right and left of the level. You can also eat them if you have gotten a power-up in the last 15 seconds, a sound and change in the appearance of the ghosts tells you when you can eat them.

Design choice explanations:

I chose to do the prototype in “3D” instead of 2D. By 3D I’m talking about using 3D models and a 3D world instead of using sprites and playing on a 2D map. I chose to work in 3D mostly because of the different projects I know about Funcom. The Secret World and Age of Conan were both made as 3D worlds and I thought working in the same environment could show that I can work with your teams.

To direct the different ghosts, I choose to use the navmesh. That way, I could have an efficient pathfinding that I know is working and adapted to the engine. Using the navmesh helped me to save some time, and developing a function that already exists and is pretty efficient is a waste of time. Refactoring is a good way to save time and resources. Because of the way I did the navmesh, the ghosts can chase you down the teleport corridors but as you teleport, they will go back the corridor to get you on the other side of the map.

The player moves its pawn by simple force addition on its rigidbody. That way, it continues to move in one direction until it gets a collision. Unfortunately, I have also had some problems with it that I explain a bit further. I tried to attach the player to some precise coordinates to avoid that problem but didn’t manage to make it works.

In order to know my map and only have to generate the navmesh once, I made the map a prefab. I spend a small time doing it with different cubes but at least it’s done and can be instantiated if needed.

Because of the time I spent around small problems (mostly on the AI and the way they or the player move), all the ghosts have the same AI. It is more a question of priority to get the game to work as much as possible than a time management problem. I preferred to prioritize the gameplay over the AI quality.

Because most of the code goes to the gameplay and AI, I didn’t spend a lot of it on the menus, but they are working as wanted (mostly). You can play the game, quit it, and when you die, restart or return to the menu.

Know bugs and problems:

I have a weird glitch on the “play” selection on the menu. For no visible reason, the collider box for the OnMouseEnter and OnMouseExit is weird and allows the player to click a bit above the “play” button.

Because of the way I made the map, the enemy navmesh is more permissive than the original way the ghosts moved. As a result, they can follow you in the teleport corridor and around their “cage”. It also seems that because of the way I made them get a target to go to, they sometimes “freeze” before continuing to pursue the player.

Player movement is not that good. The pawn gets slower, can lose some power and the player has to press the arrow another time. The problem comes from the fact that the pawn hits the walls and loses power. I tried to constraint the translations but it was not as good as I wanted so get back to the way it is now.

I don’t really know why but the player kind of bounce around the various walls. It could be a problem of physic materials (despite the fact I tried to diminish the bounce by applying some). I must admit I don’t really know why it acts like that.

Enemy collision, they can collide with each other even if they are not supposed to. I tried to ignore their collision but it doesn’t seem to work. On the same kind of problem, even if the spawn position of the enemy I set, it often happens that it spawns around the cage instead of inside.

If killed on the player spawn position, because the enemy is still here, you will probably be killed until game over. This could be resolved by having a wait before the respawn and making the enemies go away as soon as they killed you.

At the moment, some sounds can be cut and the loop that I activate for the siren goes onto the “paku” sound. It can be resolved by adding more Audio Sources to the game controller and assigning the clips to different sources.

There might be other bugs I didn’t see or that I didn’t think were important to be mentioned here.