

Project: Pac-man

Team2: 薛承昊, 傅旻昊, 李旻镐, 黄鸿贤, 丁润宇

Abstract:

This report will provide a brief overview of our team project resolution process, including:

1. "Pacman" game background history and the rules of the game
2. The objectives of the project and the issues that need to be addressed
3. The way we Tried to solve the problem and the final solution
4. Some ideas about optimizing our "Pacman" game

Background

As far as game is concerned, we can never forget a classical game called pac-man.

Pac-Man, stylized as PAC-MAN, is an arcade game developed by Namco and first released in Japan in May 1980. It was created by Japanese video game designer Toru Iwatani. It was licensed for distribution in the United States by Midway and released in October 1980. Immensely popular from its original release to the present day, Pac-Man is considered one of the classics of the medium, and an icon of 1980s popular culture. Upon its release, the game—and, subsequently, Pac-Man derivatives—became a social phenomenon that yielded high sales of merchandise and inspired a legacy in other media, such as the Pac-Man animated television series and the top-ten hit single "Pac-Man Fever". Pac-Man was popular in the 1980s and 1990s and is still played in the 2010s.

When Pac-Man was released, the most popular arcade video games were space shooters, in particular, Space Invaders and Asteroids. The most visible minority were sports games that were mostly derivatives of Pong. Pac-Man succeeded by creating a new genre. Pac-Man is often credited with being a landmark in video game history and is among the most famous arcade games of all time. It is also one of the highest-grossing video games of all time, having generated more than \$2.5 billion in quarters by the 1990s.

The character has appeared in more than 30 officially licensed game spin-offs, as well as in numerous unauthorized clones and bootlegs. According to the Davie-Brown Index, Pac-Man has the highest brand awareness of any video game character among American consumers, recognized by 94 percent of them. Pac-Man is one of the longest running video game franchises from the golden age of video arcade games. It is part of the collection of the Smithsonian Institution in Washington, D.C. and New York's Museum of Modern Art.

Impact

The game is regarded as one of the most influential video games of all time, for a number of reasons: its titular character was the first original gaming mascot, the game established the maze chase game genre, it demonstrated the potential of characters in video games, it opened gaming to female audiences, and it was gaming's first licensing success. In addition, it was the first video game to feature power-ups, and the individual ghosts had deterministic artificial intelligence which react to player actions. It is also frequently credited as the first game to feature cut scenes, in the form of brief comical interludes about Pac-Man and Blinky chasing each other around during those interludes, though Space Invaders Part II employed a similar technique that same year. Pac-Man is also credited for laying the foundations for the stealth game genre, as it emphasized avoiding enemies rather than fighting them, and had an influence on the early stealth game Metal Gear, where guards chase Solid Snake in a similar manner to Pac-Man when he is spotted.

Pac-Man has also influenced many other games, ranging from the sandbox game Grand Theft Auto (where the player runs over pedestrians and gets chased by police in a similar manner) to early first-person shooters such as MIDI Maze (which had similar maze-based gameplay and character designs). Game designer John Romero credited Pac-Man as the game that had the biggest influence on his career; Wolfenstein 3D was similar in level design[81] and featured a Pac-Man level from a first-person perspective, while Doom had a similar emphasis on mazes, power-ups, killing monsters, and reaching the next level. Pac-Man also influenced the use of power-ups in later games such as Arkanoid, and the game's artificial intelligence inspired programmers who later worked for companies such as Bethesda.

Gameplay

The player controls Pac-Man through a maze, eating pac-dots (also called biscuits or just dots). When all pac-dots are eaten, Pac-Man is taken to the next stage. Between some stages, one of three intermission animations plays.[23] Four enemies (Blinky, Pinky, Inky and Clyde) roam the maze, trying to catch Pac-Man. If an enemy touches Pac-Man, he loses a life. Whenever Pac-Man occupies the same tile as an enemy, he is considered to have collided with that ghost. When all lives have been lost, the game ends. Pac-Man is awarded a single bonus life at 10,000 points by default—DIP switches inside the machine can change the required points to 15,000 or 20,000 ,or disable the bonus life altogether. The amount of lives can be set to 1 life only or up to five lives maximum.

Near the corners of the maze are four larger, flashing dots known as Power Pellets that provide Pac-Man with the temporary ability to eat the enemies. The enemies turn deep blue, reverse direction and usually move more slowly. When an enemy is eaten, its eyes remain and return to the center box where it is regenerated in its normal color. Blue enemies flash white to signal that they are about to become dangerous again and the length of time for which the enemies remain vulnerable varies from one stage to the next, generally becoming shorter as the game progresses. In later stages, the enemies go straight to flashing, bypassing blue, which means that they can only be

eaten for a short amount of time, although they still reverse direction when a power pellet is eaten; starting at stage 20, the ghosts do not become edible (i.e., they do not change color and still make Pac-Man lose a life on contact), but they still reverse direction.

Aims:

Our team's project is to prepare a classic game to eat beans, the preparation of the game mainly need to solve the following questions:

1. Solve the problem of describe "beans". That is, after "Pacman" passing a certain coordinates, the "bean" in the coordinate disappeared and converted into the corresponding score.
2. Solve problem of "wall". After "Pacman" traveling to the wall, it can 't move forward, only along the provision of the channel from the wall into the channel.
3. "Pacman" control problems, the player by pressing the fixed button can be achieved to control the "Pacman".
4. Solve the "enemy" line of the problem, that is, what route "the enemy" will be along?

The process of programming and the result

First of all, before we start designing the game, we should first make sure what elements are present in our game. As a classic "Pac Man" game, we must include the following elements in the game: pac-man, bean, wall and enemy.

1.Pacman:

the first thing we consider is how to move it as a player controlled by the player. We define below four variables, do not do any operation in the game player, the four values are false, and when the game player input specified command, the corresponding direction will be changed to True, and pac-man can achieve the function of moving. The instantiation of the game object is implemented through the global function in the package.

2.Beans

As far as beans are concerned, we first decide to make sure the beans are in the game as a necessary context for the game as a whole. The beans to tag and its matching with the corresponding coordinates, the coordinates and the label with beans in and placed in the dictionary, in solving the problem of coordinate beans, we'll consider giving beans an example, using package circle function to establish the model of beans.

In combination with beans and PAC people, we have to solve the problem of how PAC people eat beans. After all, when the beans are all eaten, the player wins. We define a "eat" function and a function to monitor whether or not to "eat", and when the bean and bean positions coincide, we will first decide whether or not to coincide by

the latter function. And then through the previous function, the accumulation of points and delete the coordinates of beans. In the end, when the coordinates of the beans are completely deleted, the player wins.

3.Wall:

In the game of eating beans, walls are used to limit the movement of beans, the distribution of walls forms channels, Pacman move in the channel, avoid the "enemy", eat "beans" to complete the game.

In our design of the Pacman game, if "Pacman" meet a wall, it can not move to the direction of the wall, which displayed on the screen is "Pacman" is blocked by the wall, unable to move forward.

We calculate the distance between the coordinates of the bean and the coordinates of the wall, change the operation of the corresponding command, to achieve the effect of blocking in our program. The specific approach is to use the four previously defined functions, which is "upmove", "downmove", "leftmove", "rightmove", when the wall appears in a side of "Pacman", the program firstly determines whether the distance between the two is less than the value (15) we set, and then by judging the direction of the well, for example, the left, then the function leftactive will be changed to False, then the result showed on the game is the player can't use the order "leftmove" to make "Pacman" go through the wall. Through this method, we achieve to limit the "Pacman" moving.

4.Enemy:

In "Pacman", the enemy is an important source of challenge, more intelligent enemies will give gamers the feeling of much more high degree of difficulty. With the development of Pacman game, there have been many interesting provisions. In our game, we use the most simple rules: the game ends when "Pacman" meet enemy.

The question we are facing is how to develop the way the enemy moves In order to solve this problem, we thought of three methods

Method 1: the enemy in accordance with the provisions of the route, repeated in the route to move. This is one of the easiest ways to move, and according to the route we have already written inside the program, the enemy will move on this route repeatedly. The advantage of this approach is that it is easy to write the code, but the trick is to use this method the game will be too simple, lack of fun.

Method 2: the enemy move randomly. Using this method, the enemy will move in a random direction. This method uses the code is relatively simple, only need to randomly generate four numbers to represent the four directions. Compared with the previous one, the enemy is more intelligent, the difficulty of the game has increased, and the preparation of the code is also very simple. The downside is that the enemy in this method may have repeatedly hit the wall or go back and forth.

Method 3: the enemy calculate the optimal path according to the location where the "Pacman" is, and then move to "Pacman". This method requires a more complex algorithm, but it is clear that the enemy will become more intelligent. The advantage of this method is that the game becomes challenging, the enemy begins to have simple

intelligence, but the shortcomings of the method are obvious, that is, we need to write complex code, while a simple test on the computer shows that this method is out of computer computing Ability which makes the game experience worse

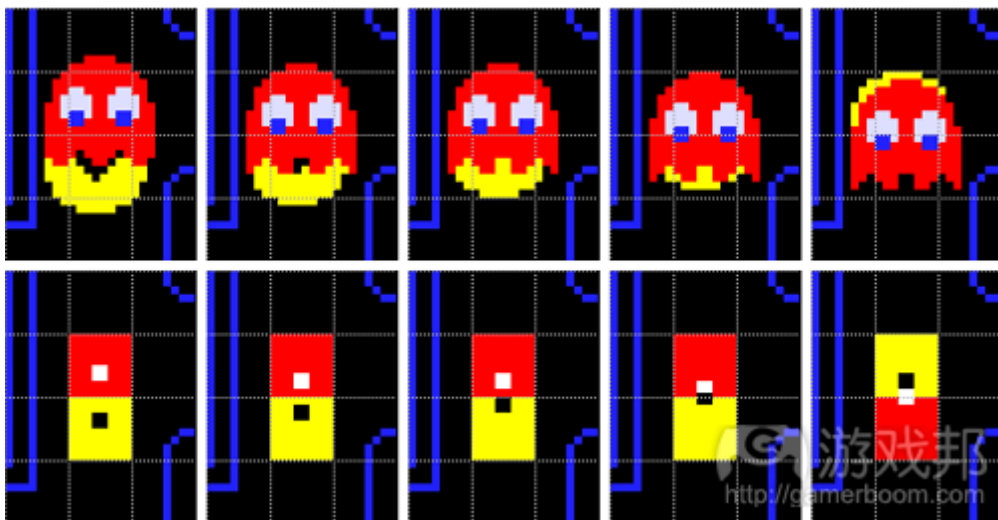
Based on the above analysis and taking into account the actual computer condition, we finally used the second method in our program

Moving forward and wrapping up

The problem of crossing the ghost (remain to be settled)

Occasionally, "Pac Man" walks straight through the ghost without being hurt. If you play this game long enough, you will eventually see the pac-man hit the ghost but unharmed appeared on the other side, this phenomenon does not occur frequently, so when they enjoy the moment! Some game player even in their model considered in this mysterious "through" probability.

The root cause of this phenomenon lies in the way the game detects the collision between pac-man and four ghosts. Whenever the pac-man and a ghost take the same map, the game is always judged to be "Pac Man" hit the ghost, and then the player loses his life. It's not important whether the ghost moves onto the map that the PAC takes, or if the pac-man moves to the map that the ghost occupies, anyway, the results of the two cases are the same. This logic has been sufficient for dealing with collisions of 99% of the game, but it does not take into account a very special situation:



Pass thru Bug(from home.comcast.net)

The image above shows the necessary conditions for this particular case. The upper five adjacent frames show the ghost and the pac-man crossing each other. The lower frame with their pixel locations currently occupied by maps and their center point to show the same scene. Pac-man and ghosts are in exactly the same position and speed as they cause the supply and marketing society to exchange maps. In other words, the center point of pac-man to move up to the ghost' position, and the center point of the ghost moved down to the Pac Man map, two at the same speed, causing them to pass each other. The game did not detect collision. Note that from the upper frame, the original position of pac-man is in the fourth frame map edge; from the lower frame, it is still considered in the map inside, but the upward moving 1 pixels will make it occupied under a texture edge. When the pac-man and the ghost swap the map in fifth frames, the pac-man can continue its happy journey because it has not been hit by a ghost!

Summary

As game enthusiasts, we made the project not only to try to write Pac Man in the python environment, but also to pay tribute to such a world-famous game. Throughout the process, we found a lot of problems, and finally solved them. We have a deeper understanding of how to write Python programs, and most importantly, we have lots of fun during the process.