Pisco Med Publishing

Python-Based Little Dinosaur Game Design

Laihao Huang

Zhejiang Gongshang University, Hangzhou 310000, China.

Abstract: Python is a cross-platform and interpreted high-level programming language which is open source and free. It has a rich and powerful library that can easily connect various modules made in other languages, so Python is often called the "glue" language. [1]The Python language has developed rapidly in recent years and has a wide range of applications. A large number of Python applications can be found in such fields as Web programming, graphics processing, hacker programming, big data processing, web crawlers, scientific computing, and game programming and so on. The invention and research of the Python language are mainly for facilitating learning and application. Therefore, the Python language is open source, with characteristics of simple syntax and developer-friendly writing and understanding. This article aims to master the advantages of Python language in the field of game development through the design and analysis of the little dinosaur game.

Keywords: Python; Little Dinosaur; Game Design

1. Overall planning design of game development

As we all know, Google Chrome will provide a dinosaur game when there is no network connection to help ease our anxiety. At this time, just press the space bar, the little dinosaur will move and begin to run wild in the desert, and then there will be a steady stream of obstacles such as cacti, birds and so on. Players need to control the little dinosaur to jump into the air to avoid these obstacles. Enter chrome://dino/ in the address bar of the browser, and you can play this game even when connected to the Internet.[2]

With the rapid development of the Internet industry, various powerful programming languages and real-time electronic game development modules have also been born, such as the Pygame module in Python, which provides a technical basis for the development of game programs. For this reason, this article proposes the development of a little dinosaur game based on Python language. According to the running process of the game and the embedding of game elements, the development is divided into 4 classes: dinosaurs, obstacles, layouts, and integrals. Finally, it will encapsulate each class together.

2. Detailed design of game program

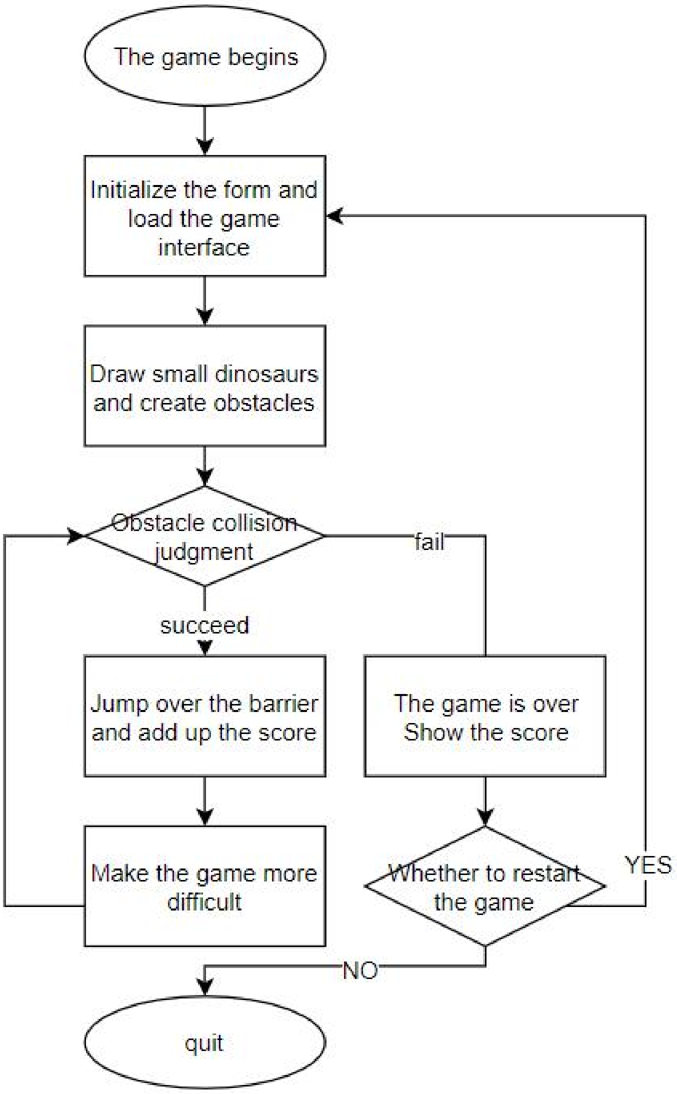
The game begins
Initialize the form and load the game interface
Draw small dinosaurs and create obstacles
Obstacle collision judgment
fail
succeed
Jump over the barrier and add up the score
The game is over Show the score
Make the game more difficult
Whether to restart the game
YES
NO
quit

Figure 1

2.1 Background building

In Python game development, the Pygame module is mainly used. After importing the library using import, it will start laying the background. The entire scene of the game needs a window as a carrier to display the picture in the game. For the simplicity of the page, the background size can be set to 1000*500. At the same time, in order to unify the character and background style of the entire game, a background picture is prepared. During development, you can directly call the method in the module to import it.

2.2 Little dinosaurs and their movement patterns

When the little dinosaur jumps, you first need to set the default fixed position of the little dinosaur on the map, and then judge whether the space bar in the keyboard is pressed. Same as the background picture, you need to import three pictures corresponding to running and jumping, and switch the pictures through the state transition to complete the corresponding action. In order to adjust the difficulty and track the status of the little dinosaur, you need to create a little dinosaur and set three parameters at the same time, position, velocity, and acceleration. You need to record the position, speed and acceleration of the little dinosaur respectively. At the same time, in order to return to the initial state when restarting, the initial values of velocity and acceleration must be set to 0. Finally, call the function in the main function to draw the little dinosaur.

Then realize the movement of the little dinosaur. According to its game function, we divide the movement mode of the little dinosaur into the most basic two categories, namely, running and jumping. In order to avoid obstacles, jumping is the key step.

The specific implementation method is as follows: [3]First, the implementation of running is to set the position of the little dinosaur unchanged, and scroll the background to achieve visual advancement. Next, in order to realize the jump, two parameters need to be set: jump-velocity and gravity, which are the displacement and gravity when jumping, to ensure that gravity will be added to the little dinosaur to return to the initial position after the little dinosaur jumps up. Set up a function jump to encapsulate related operations, and realize the up and down displacement of the little dinosaur through the addition and subtraction of parameters and the acceleration of gravity. Then by detecting the pressing of the space bar, call the jump function to complete the jump.

2.3 Obstacles and their movement patterns

Before realizing the appearance of obstacles, it is necessary to consider the size, moving speed and interval of the obstacles. If the obstacles that appear each time are the same, then the game will lose the fun. Therefore, it is necessary to load multiple obstacle pictures of different sizes, and then randomly select and display them. In addition, you need to calculate how often an obstacle appears and display it in the main window.

The specific implementation method is as follows: [4]first import random numbers, create an obstacle class named Obstacle, define a parameter in the class, and then load the obstacle picture in the initialization method. Create a random number from 0 to n, and select obstacles based on this number, and finally create the size of the obstacle rectangle based on the width and height of the picture, and set the drawing coordinates of the obstacle. Then in the Obstacle class, first create the Moving method to realize the movement of the obstacle, and then create the Obstacle method to realize the drawing of the obstacle. Define the time to add obstacles and the list of obstacle objects in the main method, place them in the code of drawing dinosaurs, and set the interval of the time between obstacles. If the interval is too fast or too slow, it will affect the normal progress of the game.

2.4 Collision and integration functions

When realizing collision and integration, you first need to judge whether the two rectangular pictures of the little dinosaur and the obstacle collide. If there is a collision, the function is called to end the game, otherwise, the little dinosaur has crossed the obstacle and added points. And the score is displayed in the form.

The specific implementation method is as follows: In the Obstacle class, create the showScore method below the Obstacle method to display the score in the top middle of the form. Complete the game ending method above the main method, and finally load the game ending picture and display the picture in the middle of the window, and display the score at the same time. In the main method, under the code for drawing the obstacle, it is necessary to judge whether the little dinosaur collides with the obstacle. If there is a collision, the game end function is called, otherwise, the score is increased and the current score is displayed.

3. Important game details and function optimization:

3.1 The overall operation of the game

There are several problems in game running: how to judge when the game is in progress or over, and how to restart the game.

Solve the first problem first. You need to set a variable running to distinguish the state of the game, and set a function to end the game, and the running and continuation of the game only need to determine the value of running. Initially set the value to 1, and then all parts that should stop the game will call the game end function to set running to 0 for unified management.

Then solve the second problem. The game ends when the little dinosaur hits an obstacle. At this point, you need to ask the player whether to restart and adjust the data accordingly.

The specific implementation method is as follows: first display the prompt sentence in the center of the screen, judge that the player enters N and Y from the keyboard, and automatically close the window when the input is N and exit the game. When the input is Y, you need to clear the obstacles on the screen and redraw Little dinosaur and reset other key parameters in the game.

3.2 Detailed optimization on the small dinosaur movement mode

Recalling the previous implementation of the small dinosaur movement mode, the small dinosaur jumps by adding and subtracting parameters, but in actual play, the player often presses the space bar many times, and the small dinosaur will continue to jump in the air at this time, which is obviously not consistent with common sense. Therefore, it is necessary to add a parameter jumping before the original jump function, which is initialized to 0 at first, and is updated to 1 when the jump function is called, and the judgment is performed until the little dinosaur returns to the initial position and it is updated back to 0. During the game, as long as the value of jumping is 1, it means that you are in the air and cannot jump again.

3.3 Memory optimization for game running

In the implementation of the previous obstacle module, this function was realized by randomly calling obstacles and setting their moving speed and appearance interval. But at the same time, a problem arises. When calling obstacles, each obstacle is an independent individual. When the little dinosaur jumps over the obstacle, the obstacle does not disappear but just moves outside the main window, and the player cannot see it. That's it. As the number of obstacles in the later game increases, new obstacles are constantly called, and the memory consumed by the game will continue to increase.

To solve this problem, you can set a function remove-out-bound-ememies to loop through each obstacle. When the coordinate of the obstacle is detected as a negative number, that is, after it has exceeded the screen, it will be destroyed and the memory will be recycled.

3.4 In-game difficulty enhancement function

In actual games, a single game content will make players feel boring and lose interest. For this reason, the difficulty of the game needs to be adjusted to increase the playability.

The specific implementation method is as follows: Set the function increase-enemy-velocity, and increase the obstacle moving speed after a period of time, in which attention should be paid to the adjustment of the obstacle appearance interval set before, and both too fast speed and too short interval will cause the game to appear A situation that must fail.

Conclusion

With the development of artificial intelligence in recent years, the application of the Python language has become more and more common. Using the Pygame module in the Python language to design a little dinosaur game, you can find that the Python language is also very easy to develop games, and it can complete the same function with less code than other languages. [1]Python has a powerful standard library. [5]The core of the Python language only contains common types and functions such as numbers, strings, lists, dictionaries, and files, while the Python standard library provides additional power such as system management, network communication, text processing, database interfaces, graphics systems, and XML processing features. The Python standard library has a clear naming interface and well-documented, so it is easy to learn and use.

References

[1] Yuxian dong. little dinosaur game design and analysis based on python [j]. modern information technology, june 25,2019, volume 3, issue 12:81-85.
[2] zuoru xie, lin miaoyan.change ideas to challenge the “little dinosaur” game [j].china information technology education, volume 13, june 2020, issue 14:120.
[3] changlong liu.python efficient development in practical practice [m]. beijing: electronic industry publishing house, 2019.
[4] dark horse programmer. introduction to python fast programming [m]. beijing: people’s post and telecommunications press, 2018.
[5] baidu encyclopedia.python［db/ol］.https://baike.baidu.com/item/python，2021-4.