ISSN: 2664-9640

Java-based Super Mario game development

Wenqing Zang

School of Information, Beijing Institute of Materials, Beijing, Tongzhou, China

Abstract

Super Mary is a classic and popular platform jump game from Nintendo, which was first released in 1985 and became a childhood memory for a generation. The main character, Mario (Mario), needs to jump, avoid and defeat the enemy through various levels, and finally save the princess imprisoned by the dragon. The background of this project is based on the love of Super Mary games and the homage to classic games, and the pursuit of learning and practice of the Java programming language.

Keywords

Java, game development, eclipse.

1. Introduction

With the development and changes of The Times, computer technology has been integrated into our life through various ways. Every industry of computer has become an indispensable part of us, such as the game industry. Java language is an object-oriented programming language, Java language was born in the early 1990s, since its official introduction, its rapid development has changed the whole Web world. Java is based on the most popular C and C++ languages at that time. Due to the powerful Java language, it was widely accepted by the industry shortly after its advent, and at the same time, many software developers have developed a lot of products that support Java. The popularity and rapid development of the network, as well as the continuous penetration of Web technology, make Java language in the modern society's economic development and scientific research occupy a more and more important position.[1,2,3] In the project, we can use the knowledge and skills of Java programming language to restore and expand the Super Mary game, we will be supported by technology, using the characteristics and functions of Java programming language, to achieve the basic operation and play of Super Mary game, to provide players with an entertainment and enjoy the platform. Through this project, we also hope to inspire more people's interest in programming and game development, as well as nostalgia and love for classic games.

2. Design and Implementation

(1) Game frame construction: design and implement the basic frame of the game, including the game background, drawing Windows, etc.

(2) Character and enemy design: Create and manage the character (Mario) and enemies in the game, including the movement, jumping and death of the character.

(3) Map and level design: Design and implement the maps and levels of the game, including the layout of the map, the placement of obstacles and items.

(4) Collision detection: Realize collision detection between characters and enemies, props and other game elements, and deal with the logic after collision.

(5) Props and abilities design: design and implement various props and abilities in the game, such as mushrooms, man-eating flowers, flags, etc., and provide characters with corresponding actions and jumping capabilities.

(6) Music: Add the background music of the game to enhance the entertainment and atmosphere of the game.

3. Game flow

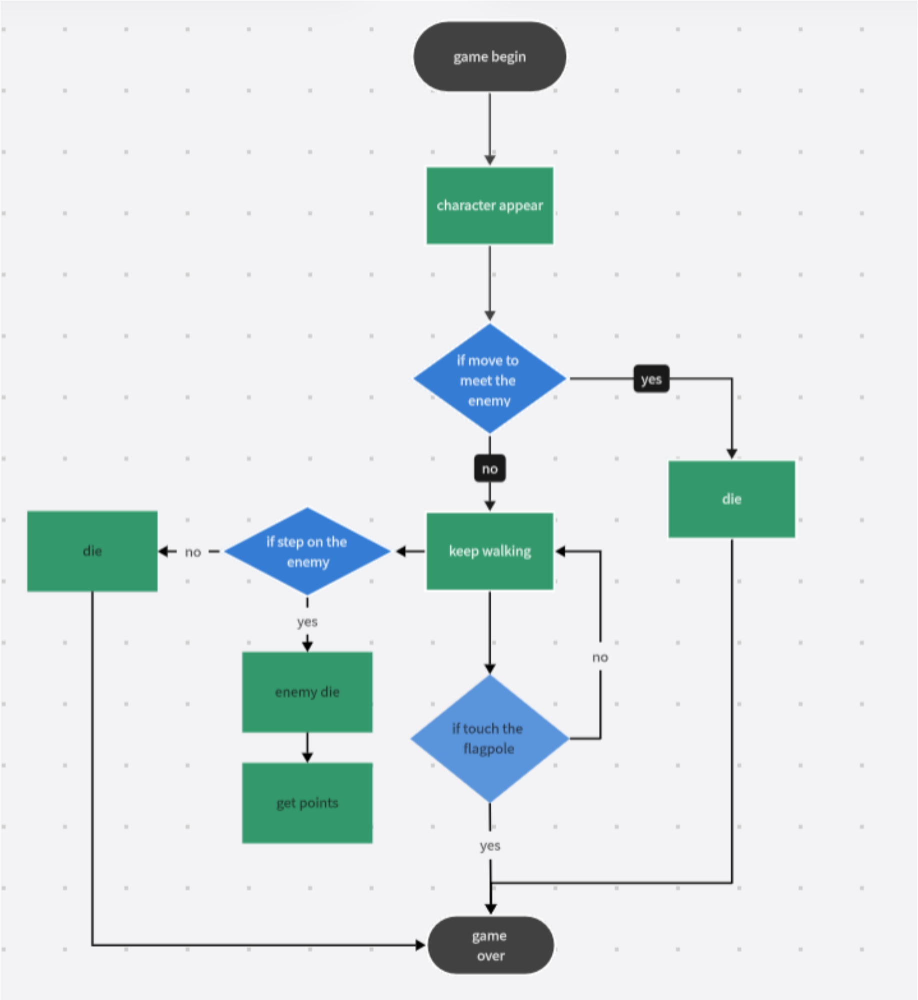

Figure 1 Basic game flow1

Super Mario is a very classic and popular platform-jumping game from Nintendo that was first released in 1985 and became a childhood memory for a generation. The game's main character, Mario, has to navigate through various levels, jumping, dodging and defeating enemies to ultimately rescue a princess who is being held captive by a dragon. The background of the project is based on the love of Super Mario games and homage to classic games, as well as the pursuit of learning and practicing the Java programming language.

In the game, the user manipulates the game characters to move, kill enemies, dodge enemies, as well as gain points, and finally finish the game and exit the game. The basic flow chart of the game is shown in Figure 1.

4. Feasibility analysis

First, JAVA is a general-purpose programming language that is perfect for developing games. However, developing a complex game like Super Mario requires a certain amount of development experience and skill. Secondly, choosing the right game engine and framework can greatly simplify the development process. JAVA has some popular game engines and frameworks, such as libGDX and jMonkeyEngine, that offer a wealth of tools and features that help speed up game development. Then, Super Mario games need to achieve beautiful graphical effects and realistic physics simulations. JAVA provides various libraries and tools, such as JavaFX and OpenGL, that can be used to implement graphics rendering and physics simulations. Finally, Super Mario games contain a large number of resources such as graphics, animation, and audio, and effective resource management and optimization is essential for game performance and user experience. JAVA provides several tools and techniques, such as multi-threading and caching mechanisms, that can help deal with the problem of resource loading and management. Therefore, choosing the right game engine and framework as well as graphics and physics engine can significantly simplify the development process, pay attention to resource management and optimization to ensure the game performance and user experience, and also need sufficient time and energy to complete the development and testing of the game. In general, this project development is feasible, but it needs appropriate skills and tool support. As well as adequate preparation of time and resources.

5. Debug run

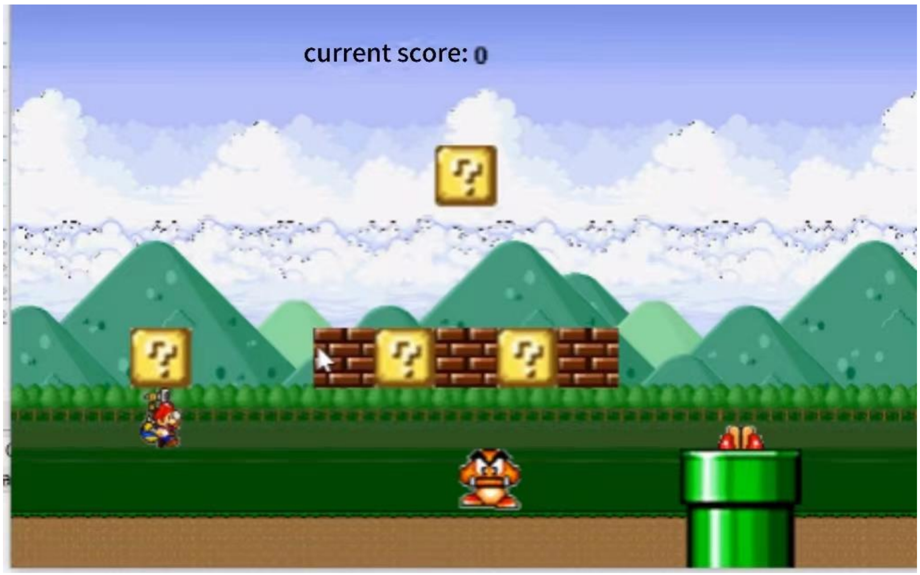

Figure 2 Run effect2

This program is written through the Window system +Eclipse development platform, Figure 2 for the successful operation of the program effect.

In the debugging process encountered a lot of problems, including the following problems:

1. Physical simulation: realize the movement and jump of the character and the movement of the enemy, simulate the movement and collision of the object. During the project, due to some code problems, the character could not move, moved too fast, the enemy did not appear, and the obstacles were disorderly, resulting in the map could not be carried out.

2. Collision detection: Design an effective collision detection algorithm to determine whether the characters collide with enemies, obstacles and other game elements. In the course of the project, the characters did not die after hitting the enemy, the enemies did not die after the characters killed the enemies, and the characters and enemies crossed the obstacles after hitting the obstacles.

3. Map design and level difficulty: Design the map layout and level difficulty to ensure that the game is playable and challenging.

4. Sound effects and music integration: Select appropriate sound effects and background music, and integrate them into the game to enhance the entertainment and atmosphere of the game. In the process of the project, due to the lack of knowledge of Java functions, the music could not

be played. Later, I solved the problem by asking the instructor and inquiring relevant knowledge.

5. The appearance of the game character: the character appears in the correct position on the map, so that it can carry out various functions. In the process of the project, due to the wrong setting of some parameters, the position of the character appeared too far forward, back, up and then fell directly out of the map, and appeared in the land under the ground. Later, through several experiments, the role appeared in the correct position on the map.

6. The Conclusion

Through this project, we not only improve the understanding and application ability of Java programming language, but also learn the process and method of game development. We deepened our understanding of game physics simulation, collision detection algorithm and user interface design, and exercised our ability of problem solving and teamwork.

References

[1] G. Li. Crazy Java Handout (2nd edition) : Electronic Industry Publishing House, 2014:2.
[2] J.H. Zhao. Java Programming: Beijing Machine Press, 2005:1-2.
[3]Tomorrowtech. Java From Introduction to Mastery (3rd Edition) : Tsinghua University Press, 2014.