

The Design of Airplane War Game Based on Python

Xiangyu Chen, Kun Liu*

College of Applied Science and Technology, Beijing Union University, Beijing, China

* Corresponding author: Kun Liu

Abstract: With the development of the times and social progress, electronic games have become an important form of people's leisure and entertainment. Python language is generally directly embedded in the game, not only can the performance of the game engine be significantly improved, but also can fully enjoy the outstanding advantages of scripted development. This article focuses on a comprehensive discussion on the design of the airplane war game based on Python, and uses the Python library to develop the game content in depth.

Keywords: python; python library; game design; airplane war game

1. Introduction

In the first half of 2021, the actual sales revenue of the Chinese game market reached 150.49 billion yuan, an increase of 7.8% year-on-year, and the number of Chinese game users was nearly 667 million, an increase of 1.38% year-on-year. Among them, the actual sales revenue of China's self-developed games in the domestic market was 130.112 billion yuan, an increase of 8.3% year-on-year, accounting for more than 80% of the actual sales revenue of the Chinese game market, and the actual sales revenue in overseas markets was US\$8.468 billion, an increase of 11.58% year-on-year [1], China's self-developed games actually accounted for an upward trend in revenue. Game is a product that can help people relieve stress, entertain, and make people feel a sense of accomplishment and pleasure, and is loved by more and more people. In recent years, under the promotion of the vigorous development of Internet technology, the scale of development of the game industry in China has continued to grow, and the output value has reached about 250 billion yuan. The corresponding game service industry has also achieved rapid development. In this era, the game industry has faced tremendous challenges. For a long period of time in the future, with the improvement of China's free game research and development capabilities, China's game industry will continue to maintain a high-speed development trend, and the scale of development will also become larger and larger, so that games will be accepted and integrated into a wider audience. In their lives, it has become one of the important components of people's leisure and entertainment activities. The

advantages of Pygame can better meet the diversified needs in the game development process, and can provide relatively rich resources for game development and design work, improve work efficiency, shorten development cycles, and reduce development costs.

2. System Function Analysis

The game designed in this article is "Airplane War". During the operation of the game, players use the up, down, left, and right keys of the keyboard to control our planes that automatically fire bullets at a constant speed to attack randomly generated enemy planes. Accompanied by background music that fits the battle state, it is easier to bring players into it. During the game, some props will be randomly generated [2]. The props include life packs, protective covers and nuclear bombs. The life packs provide the player with the function of adding one life. The protective cover makes our airplane temporarily invincible until the protective cover disappears. The nuclear bomb is used to kill the enemy airplane randomly generated on the full screen in seconds to help oneself through the dangerous situation [2]. This brings a variety of game functions to players and improves player experience. In the game, when our plane's life is all used up, the game ends, and then an interface for the player to choose to continue the game or exit will pop up, and after the end, the record module will update the game record according to the player's score.

Based on the above analysis, the following factors must be considered in the process of development and design, as shown in Figure 1:

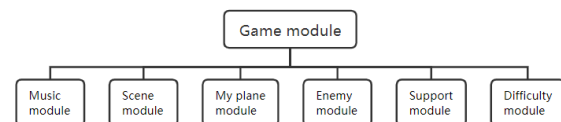


Figure 1. System function module diagram

3. Overall Game Design

The programming language Python used in the design of this airplane war game is version Python3.7. The development tool is IDLE to call the Pygame package and various functions to develop, run and test the game. The entire process of editing, compiling, and debugging of the game program is based on the PC platform.

Pygame is a Python wrapper module of the SDL (Simple DirectMedia Layer) multimedia library. These

classes and classes enable SDL to support cdrom, audio and video output, keyboard, mouse, and joystick input. Pygame is a game library created by using the SDL library, which is used to develop Python program modules for game software. Pygame is a library used by Python. Pygame is a cross-platform Python module that focuses on the design of video games, including images, sounds, etc. based on SDL, allowing real-time video game development. Based on this assumption, all required game functions and concepts can be completely simplified into the logic of the game itself, and all resource structures can be provided by high-level languages [3].

IDLE (Integrated Development and Learning Environment) has basic IDE (Integrated Development Environment) functions. It is a very good choice for non-commercial development. After the Python installation is completed, IDLE will install it automatically. The basic functions of IDLE: grammar emphasis, paragraph indentation, basic text editing, table key control, debugging program. IDLE is generally a standard Python release, and IDLE can be executed in an environment where Python and TK can be executed. After opening IDLE, the interactive interactive prompt of enhanced interactive command line will be displayed, and the menu is the "stripping" style of TK. In other words, click the dotted line of any drop-down menu at the top to upgrade this menu to your own permanent window. IDLE's tester provides breakpoint, step and variable monitoring functions [3].

This article uses many functions in the Python library to realize the rapid development and design of game content. The main function of the program mainly implements the definition of member functions and member attributes, inherits related functions and adds related game features, such as the realization of the incremental function of scores, modules, loop detection of various functions and monitoring game related events, such as monitoring games the setting of superbullet events and delivery time, etc. and finally run the overall program. Figure 2 below shows the program design process in detail:

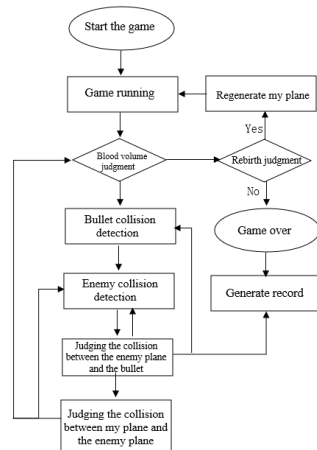


Figure 2. System flow chart

3.1 Background Music Design and Implementation

The main purpose of background music is to render the atmosphere and strengthen the overall appeal of the game. In this "airplane war" game, in terms of background music design, I learned and referenced many games, quoted different music in different scenes, and applied different sound effects during the battle. The sound effects of a single bullet fired, and the sound effects of double bullets after getting the props and the main purpose of nuclear bomb sound effects is to fully demonstrate the intensity of the entire battle process, thereby promoting the overall progress of the game. Considering multiple factors and combining actual needs, a relatively cheerful background music is selected. Part of the code is as shown in Figure 3:

```

01. def main():
02.     pygame.mixer.music.load("sound\game_music.ogg")
03.     pygame.mixer.music.set_volume(0.2)
04.     pygame.mixer.music.play()
05.
06.     enemy3_flying = pygame.mixer.Sound("sound\enemy3_flying.wav")
07.     enemy3_flying.set_volume(0.1)
  
```

Figure 3. Part of the background music code

3.2 Design and Implementation of Enemy Airplane Module

Set the flight status of the enemy airplane to a single vertical downward moving flight trajectory, adjust the flying speed of the enemy airplane according to the difficulty of the game, and call the random function to make the enemy airplane appear at different positions at the top of the window randomly. When setting the altitude, to prevent incomplete airplane or a sense of abruptness when the enemy airplane is spawned, the width and height of different enemy airplane are passed in during initialization, so that the height of the enemy airplane is generated outside the frame, and the width is generated between the width of the frame. Examples of enemy airplane are three types of airplane, large, medium and small, among which the large and medium types of airplane are given HP attributes [4]. The enemy airplane are divided into three categories: small, mid, and big, and load the width and height data of the enemy airplane model to define the attributes of the enemy airplane's flight speed and number of generations, as shown in Figure 4. Among them, when the enemy plane performs collision detection, the blood volume is reduced by 1. Design enemy airplane generation function to generate enemy airplane into two sprite groups. One group of enemies is responsible for collision detection of all enemy airplane, and the other group is used to determine the type of enemy airplane's reduction in blood volume after collision, and pass the draw-line function Draw bleed bar properties. Among them, drawing the health bar needs to obtain the position parameters of the enemy, and by detecting the blood volume deduction value feedback ratio to achieve the decreasing visual effect of the health bar.

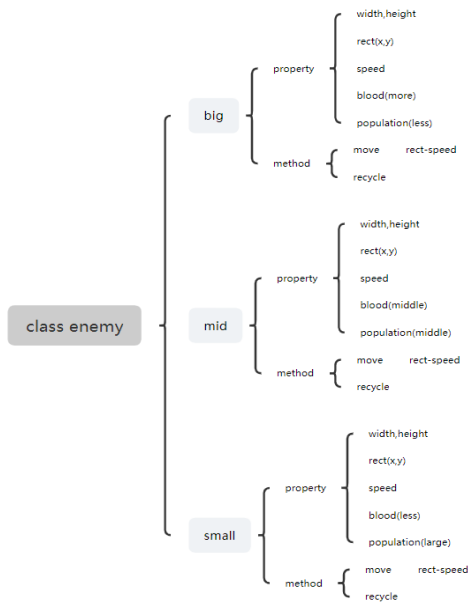


Figure 4. Enemy airplane attributes

3.3 Design and Implementation of the Battle Background

In the process of designing background pictures and background music, we designed six different background pictures based on various needs. After the boss is generated, the background picture will also change. Figure 5 below shows the six different boss-background image:

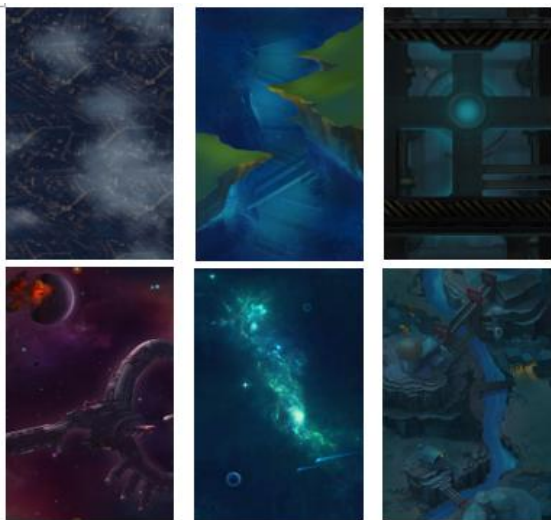


Figure 5. Boss interface background picture

3.4 Support Prop Design and Implementation

On the basis of learning and borrowing from the existing airplane war game mode, this article designed a relatively special bullet form, namely double bullets, clear screen nuclear bombs, and also added a shield module and a missile module, in which the airplane added After shielding, the immune effect can be obtained in a short time. Figure 6 and Figure 7 are the key codes of the nuclear bomb and shield module for clearing the screen:

```

01. import pygame
02. from random import *
03. class Bom(pygame.sprite.Sprite):
04.     def __init__(self,size):
05.         pygame.sprite.Sprite.__init__(self)
06.         self.image = pygame.image.load("images\bomb_supply.png").convert_alpha()
07.         self.mask = pygame.mask.from_surface(self.image)
08.         self.rect = self.image.get_rect()
09.         self.size = size
10.         self.rect.left, self.rect.top = randint(0, self.size[0]-self.rect.width), -4 * self.rect.height
11.         self.speed = 2
12.         self.active = False
    
```

Figure 6. The key code of the nuclear bom

```

01. import pygame
02. class Shield(pygame.sprite.Sprite):
03.     energy = 1500
04.     def __init__(self):
05.         pygame.sprite.Sprite.__init__(self)
06.         self.image1 = pygame.image.load("images\shield01.png").convert_alpha()
07.         self.image2 = pygame.image.load("images\shield02.png").convert_alpha()
08.         self.mask = pygame.mask.from_surface(self.image1)
09.         self.rect = self.image1.get_rect()
10.         self.active = False
11.         self.hit = False
    
```

Figure 7. Key code of shield module

3.5 Game Difficulty Design and Implementation

In terms of difficulty design, this article advocates that in terms of game difficulty control, the enemy's blood volume and score can be directly linked, thereby showing the characteristics of high-risk and high-return games. In the aspect of enemy planes refreshing, you can use the form of destroying enemy planes to control the generation of bosses, ensuring that some game masters can eliminate enemy planes in a short time, and obtain higher scores, and then generate bosses as soon as possible. You can get a higher score by destroying the boss. According to personal game experience, the game difficulty is divided into six different levels. Figure 8 below shows part of the difficulty realization code [5], as follows:

```

01. elif lv == 2 and score > 30000:
02.     if transform:
03.         add_small_enemies(smallenemies,enemies,2)
04.         add_mid_enemies(midenemies,enemies,2)
05.         add_big_enemies(bigenemies,enemies,1)
06.         inc_speed(smallenemies,1)
07.         inc_speed(midenemies,1)
08.         inc_speed(bigenemies,1)
09.         transform = False
10.
11.         is_move = False
12.         boss.reset()
    
```

Figure 8. Part of the difficulty module implementation code

4. Game System Test

The main purpose of system testing is to demonstrate the stability of the system and the realization of various functions. In the process of designing and developing programs, many errors will inevitably occur due to the influence of various factors. Therefore, system testing is very important. A software can only be put on the market after it has been tested [6]. Figure 9 shows the game test interface, and Table 1 below lists the test content in detail, as follows:



Figure 9. Game test interface

Table 1. Game test

Example name	Airplane war game test	
Example numble	001	Designer: Xiangyu Chen
Test target	Whether the game can achieve the expected function	
Step	Operation description	Desired result
1	Open the main-test.exe	The game interface is displayed, the background image is displayed, the background music is sounded, and our airplane and enemy airplane appear on the screen, producing bullets.
2	Click the arrow keys	My plane is moving
3	My elimination score reaches a certain number	Can produce props and boss spaceship.
4	After eating the props	Realize the corresponding function
5	After destroying the boss	Props are generated, and the background image is automatically changed
6	After reaching a certain score, destroy the second boss	Generate a boss spacecraft that can emit lasers
7	Press space	Can eliminate all enemy airplane except boss
8	Encountered enemy bullets or enemy airplane	After our plane is hit, if there is still life, the health value will be automatically reduced by

		1 and will be invincible for a short time; otherwise, if there is no health value, the game will end immediately
9	After closing, open main-test.exe again	Record update

5. Summary

The Python-based airplane war game designed in this paper has completed the basic modules of the game, including initialization module, combat interface module, score module and boss module. The difficulty of the game is upgraded according to the player's game progress, while the equipment supply is expanded and the sub-flexibility is improved. As well as the life reward mechanism, a airplane war game with higher playability and more functions has been realized.

References

- [1] Sina Technology. China Game Industry Report for January-June 2021. Available online: <https://baijiahao.baidu.com/s?id=1706598133947870355&wfr=spider&for=pc> (accessed on 3 December 2021).
- [2] Crawford, C. *The art of computer game design*; Osborne/McGraw-Hill: Berkeley, Calif, UK, 1984.
- [3] Kelly, S. *Basic Introduction to PyGame*; Springer: New York, USA, 2016.
- [4] Adams, E.; Dormans, J. *Game mechanics: advanced game design*; People's Posts and Telecommunications Publishing House: Beijing, China, 2012.
- [5] Sweigart, A. *Making Games with Python & Pygame*. People's Posts and Telecommunications Publishing House: Beijing, China, 2012.
- [6] Kelly, S. *Python, PyGame, and Raspberry Pi Game Development*; Springer: New York, USA, 2019.