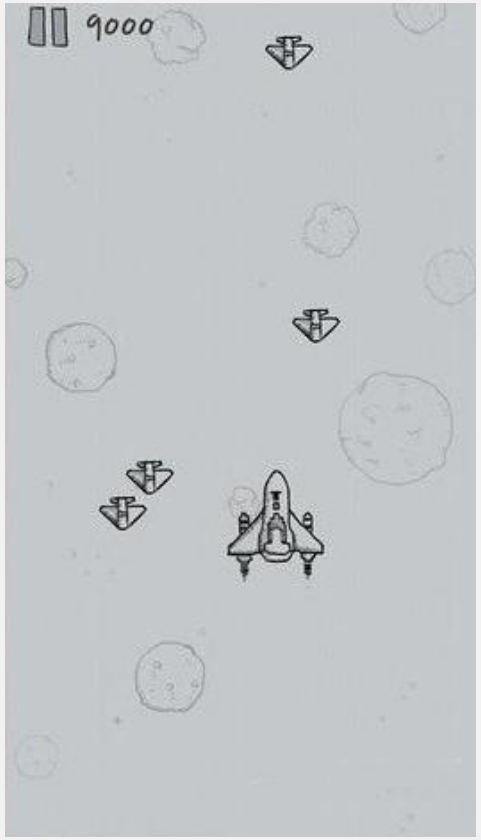
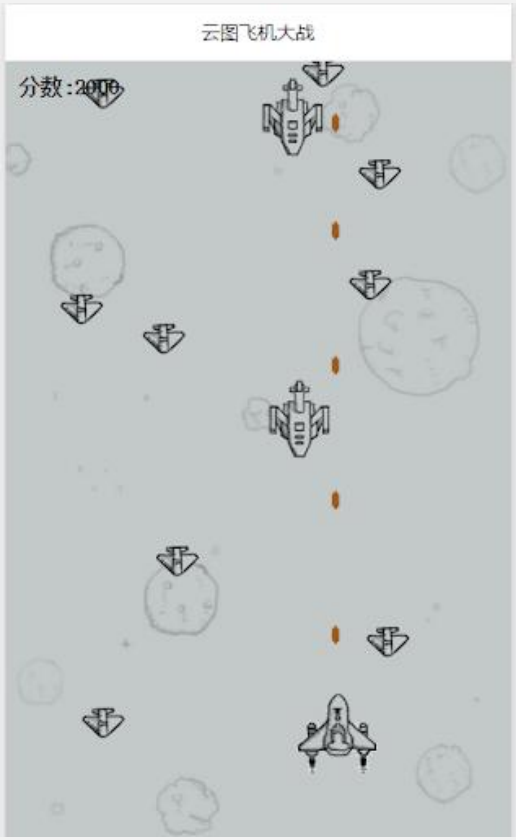
Python Project
Aircraft combat game

LanxinLi 20004530

Source of inspiration

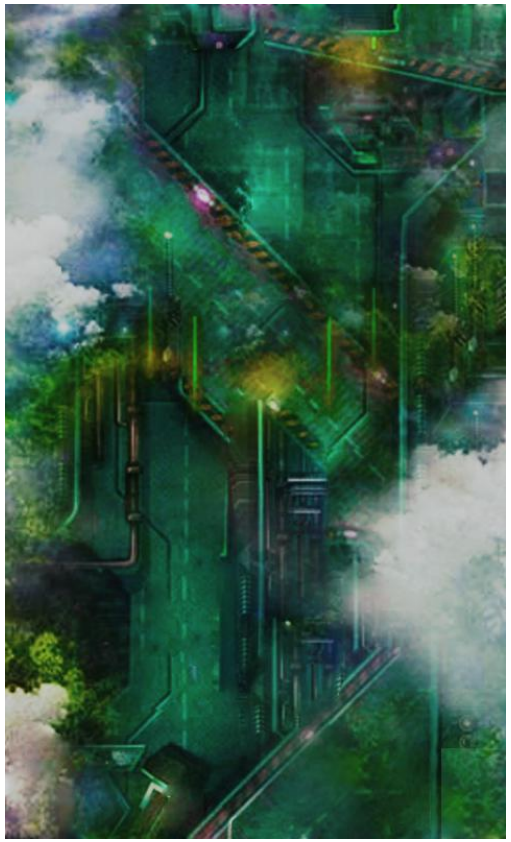
Many years ago, WeChat had a game called Airplane Wars. The interface style is simple, easy to operate, and interesting. It is very popular in China. It seems that its picture quality and gameplay are not out of date. Then, I want to use python to implement this game.

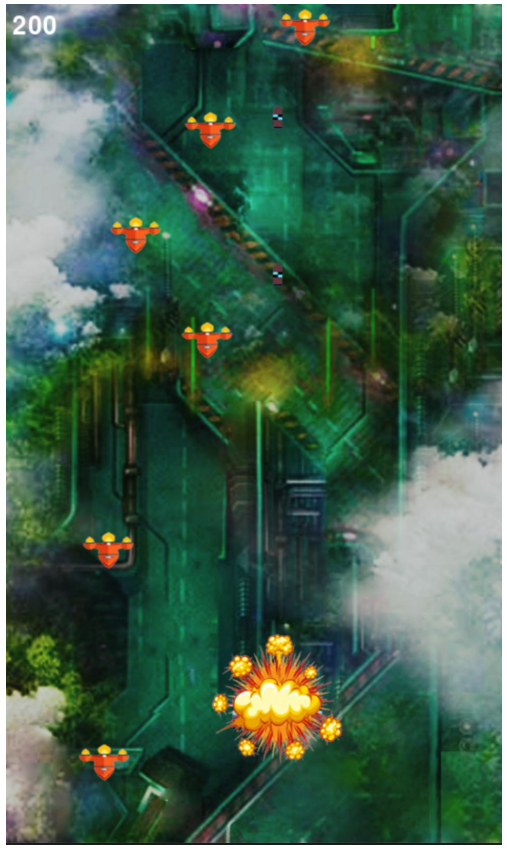




Welcome

I made an airplane war combat game with python. The player controls the airplane by using the arrow keys and fires artillery shells to destroy the enemy airplane. If the enemy airplane hits his own airplane, the player sacrifices.

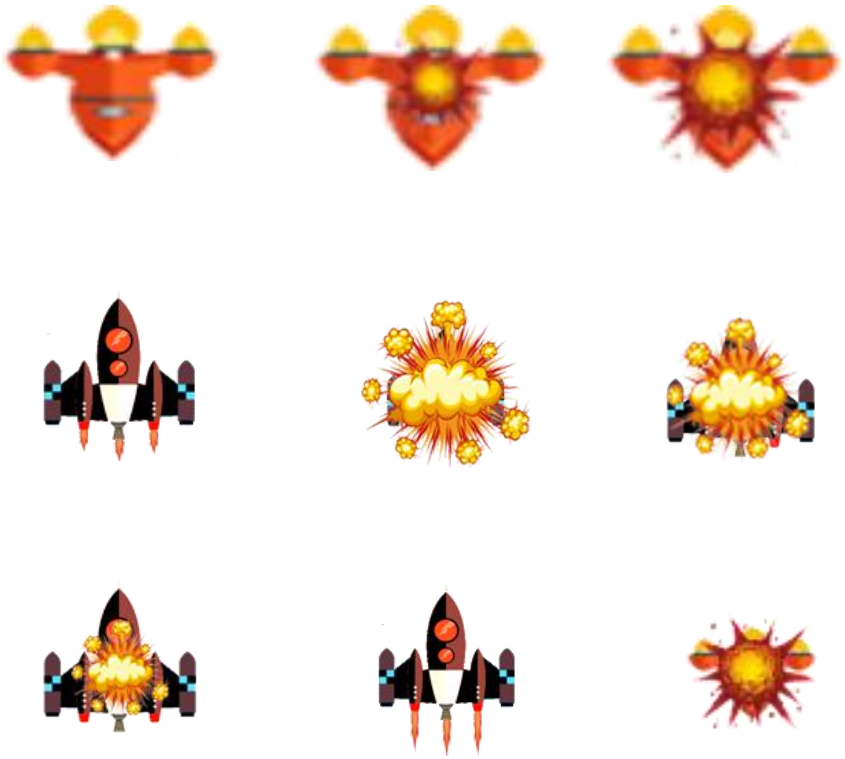


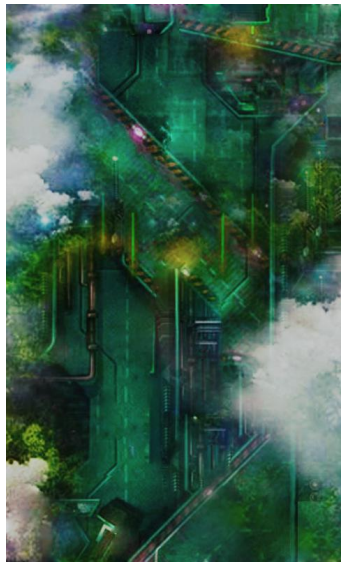


pygame module

The main technical route or method I have adopted is the introduction of the system development environment. The current program is based on python as the programming language, and the main function implementation depends on the pygame module, as well as sys and random. Mainly use the position change between surface objects, and then use event monitoring to let the program run. After the position of the Surface object changes during operation, the interface is refreshed, and when the user operates the mouse and keyboard, the corresponding event is monitored when the operation is completed.

Interface element





GVME OVER!

pygame module

pygame.display	Create and manage game windows	**set_mode(); update()**
pygame.draw	Draws shapes, points, and lines	
pygame.event	Management event	**get()**
pygame.image	Load and store images	**load()**
pygame.key	Read keyboard keys	
pygame.Rect	Manages the rectangular area	
pygame.sprite	Operation, moving image	
pygame.time	Manages time and frame information	**Clock(); set_timer()**
pygame.Surface	Manages images and screens	**blit(), get_rect()**

Module method

1. set_mode() method, to create a game screen

```
screen = pygame.display.set_mode((SCREEN_WIDTH, SCREEN_HEIGHT))
```

2. load() method, load image data

```
plane_bullet = pygame.image.load('resources/image/bullet.png')
```

3. blit() method to load the image on the screen

```
screen.blit(score_text, text_rect)
```

4. update(), method

```
pygame.display.update()
```

5. application- draw backgrounds, heroes, enemy aircraft and display them on the screen

Creation process

1.

The upper left corner (0, 0) of the initialization rectangle is the origin, the x-axis increases to the right, and the y-axis increases downwards.

```
self.rect = player_rect[0].get_rect()
self.rect.topleft = init_pos
```

2.

Class pygame.Rect that depicts a rectangular area.

Rect(x, y, width, height) -> Rect

```
    if self.rect.left >= SCREEN_WIDTH - self.rect.width:
        self.rect.left = SCREEN_WIDTH - self.rect.width
```

Creation process

3. Initialization and exit of the game

pygame provides two methods to initialize and exit the game.

pygame.init() pygame.quit()

```
261         pygame.display.update()
262
263         for event in pygame.event.get():
264             if event.type == pygame.QUIT:
265                 pygame.quit()
266                 exit()
```

Creation process

4. Game loop and game clock

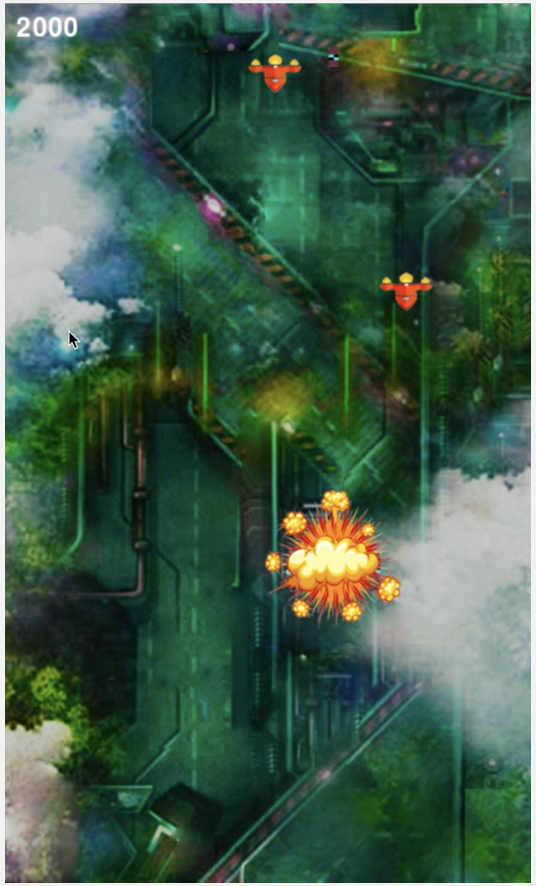
Change image position-animation effect.

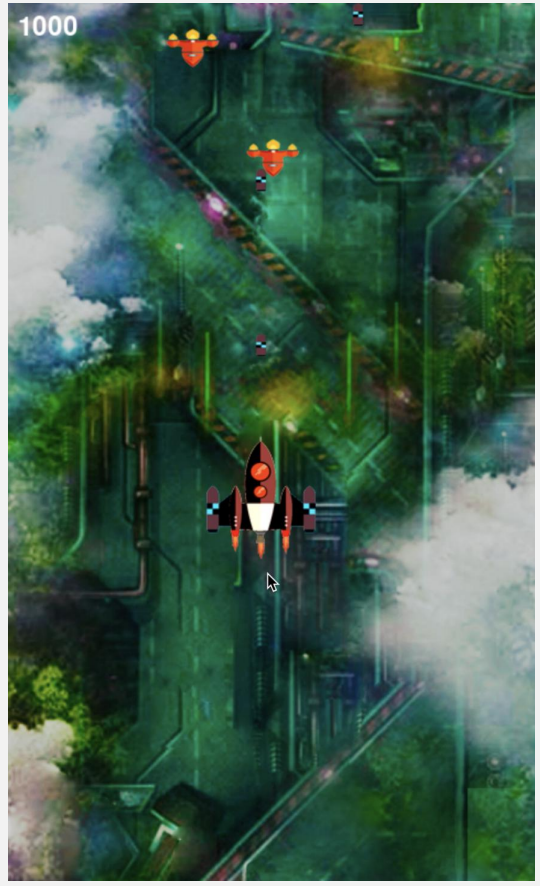
Move the position of all images every 1/60 seconds

call pygame.display.update() to update the screen display

```
178    while running:
179        
180        screen.fill(0)
181        screen.blit(background, (0, 0))
182
183        clock.tick(60)
```

Interface display





THANK YOU

Lanxin Li 20004530