

December 5, 2019

Yuhang Liang

yuhangl@andrew.cmu.edu

Design Proposal: Interstellar Game

1. Project Proposal

1.1 Project Description

The name of this project is Interstellar, which is an airplane battle game. The project is mainly to establish an online game with the background of aircraft air combat. There are two kinds of play modes, one player mode and two players mode, with different difficulty levels. In the two players mode, two players can play the same game round connected by server. Players can choose the difficulty level recommended by the game based on their play history or by themselves. In the game, players use the keyboard operation to realize the movement and achieve different skill functions to destroy the enemy aircraft. The enemy aircraft are different airplanes with different health volume, speed and flight route, which can also speed up. The game has a boss character with different shooting capabilities and meteorites that rotate constantly. In the game, players need to avoid the enemy's bullets and impacts, destroy the boss, and receive higher points to get a higher ranking on the leaderboard.

1.2 Competitive Analysis

There are games on the similar theme with my project. The game designed in this project have similarities and differences with them. The main similarity is that the rules of the game are similar. The players operate the plane to avoid the attack of the enemy planes and destroys the enemy by firing bullets.

The main difference is that the project is more diverse. The first point is that the game built in this project can be run in two clients at the same time, which can make user experience better. The second point is the diversification of enemy aircraft, the enemy's flight path, different straights or curves, and speed are different, the enemy aircraft have different healthy volume and different bullet flight algorithms, so the game is more challenging. Thirdly, the airplanes that are controlled have different skills, such as shield, space jump and rockets, which are more interesting visually and algorithmically. What's more, The user interface of the game is more comprehensive, with a status bar that updates in real time. And the project has user tracking function, so the game will record the results according to the user's wishes in order to facilitate ranking and game difficulty recommendation. Finally, the project further upgrade the flight mode, and the mathematical formula will make the aircraft have a more realistic flight acceleration change, which enables the playability of this game to be stronger.

1.3 Structural Plan

In order to make the project structure clearer, in the code aspect, the project built some py files separately. And they will be called by the main game py file. In the main game file, multiple modes will be established including the startup mode, game mode, help mode, game end mode and so on. Each mode will need to create corresponding different objects. Through the sprite method of pygame, the project will create an allGroups that manages all sprites, a sprite group of bullets, a sprite group of players and a sprite group of enemy, etc. Each group has some individual sprites. The modes call to these sprites as needed.

On the file aspect, there will be some .py files building the game, an image folder a score folder and a music folder, which will be called automatically when the game is running.

1.4 Algorithmic Plan

The trickiest part of the project is to synchronize screens on different clients. The difficulty of the problem lies in balancing the contradiction between the transmission frequency and the amount of information transmitted in a single transmission. At the same time, some sprites are continuous animations, which makes the transmission diverse and increase the difficulty. If the size of data transmitted at one time is too large, the transmission frequency will decrease, which affect the operation of the game; if the amount of information transmitted at one time is too small, the screen will be out of sync.

To solve this problem, my solution is that each client transmit its own data to the server. After the server stores data from different clients, the server allocates the corresponding data to the corresponding clients. In my project, the server tells the client what number of players it is through the client connection order. The second client only transmits the instructions entered by the second player. The first client receives the data from the second client and processes them. At the same time, it processes the instructions from the first player. The processing includes judgment of the collision function, update of health value, skills updates, etc., and transmit the attributes of each sprite about the displayed image, such as physical coordinates, image rotation angle, etc. The data is stored in a 2D list, and is transmitted after pickled. The second client will delete the original sprite and create a new sprite based on the information transmitted by the server. For the explosion animation effect, the No. 1 client transmits the physical coordinates of the explosion point, the No. 2 client adds a new coordinate point each time, and the update function is executed by the No. 2 client itself. If the animation finished, the sprite will kill himself automatically.

The advantage of this is that it greatly reduces the amount of information transmitted and increases the transmission frequency, because large-size data such as images are stored locally on the client. The server stores and distributes data from the client, which is easier to manage and distribute.

1.5 Timeline Plan

Today - November 22: Complete the basic features of the game and improve them;

November 22 - November 26: Complete the codes of all modes and achieve switching;

November 26 - November 30: Add dual player mode;

November 30 - December 3: Add more online interactions and upgrade the game;

December 3 – December 5: Organize and debug all project files

1.6 Version Control Plan

This project mainly updates the version through time. I update the version constantly and backs up it to the CMU mailbox.

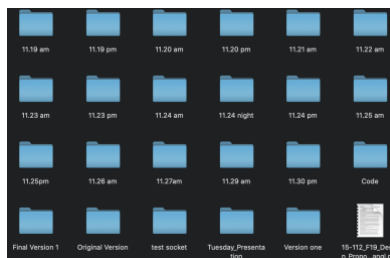


Figure.1 Record of version control

1.7 Module List

This project mainly uses two external modules. The first one is pygame, which is mainly used to implement GUI interface and game control. The second is sockets, which are mainly used in multiplayer mode to achieve online operation of two players.

1.8 TP2 Update

Based on TP1, this project added three new modules, including boot mode, splash mode, and the two players mode in game mode. A new character box has been added to the game mode to display the player's status in real time, and new function keys such as shield and space jump have been added. And the project added background music and so on.

The two players mode uses sockets to build a small server and achieved an online game. The two screens start at the same time and are updated in real time, and players can only control their own aircraft. The double player mode has more enemies, higher health bosses, new character boxes, and more.

1.9 TP3 Update

First, the project fixed a bug that the screen was not synchronized in the multiplayer mode during the period from TP2 to TP3.

Second, the function of selecting the difficulty level of the game and the user tracking function have been added. The user can enter his/her name to obtain the system's recommended difficulty level, which is based on the user's historical game records. And after the game, the user can choose whether to record the score of this game. If users are not satisfied with the recommended game difficulty of the system, they can choose to adjust it by themselves.

Third, the project added ranking functionality. The game results will be ranked based on the game history of all users, and the names and scores of users will be displayed in order on the leaderboard.

Fourth, the project upgraded the original enemy algorithm. The generation and flight of enemies no longer simply on the random function. The enemy's flight path changes from a straight line to a curve, and the enemy's flight speed will increase with the dwell time in the interface.

Fifth, the project added more dynamic effects. When the bullet hits and destroys the enemy, there will be a dynamic explosion effect and the interface is more gorgeous.

Sixth, the project added the ability to pause and quit the game. While the game is in progress, the user can press t to pause the game and display the help interface. The user can press ESC to exit the game midway and return to splash mode.

Seventh, the function after the character's death has been improved. In the multiplayer mode, if a player dies, his airplane can continue to exist in the form of a tombstone rather than disappear. It has no function other than flying. This further enhances the user's gaming experience.

Eighth, the project added ranking mode, textInput mode, and help mode, etc., and achieved flexible switching between different modes.

On Tuesday's discussion group, the group members suggested that I optimize the text input interface and add pause and exit functions in game mode. These features have been implemented in the final version.

2. Storyboard

The game takes place on earth invaded by alien civilization. Due to the attack of the alien force on the earth, you take the last hope of mankind, drive the spaceship to find the Galaxy fleet of earth, and lead the fleet to carry out the final counterattack. However, you are currently experiencing the interception of the alien fleet. And the airplane battle begins...



Figure.2 Alien Fleet

Citation: Image were downloaded from: <https://ru-bykov.livejournal.com/3941071.html> and <https://www.kinokopilka.pro/movies/23191-igra-endera?page=2>

Figure.2 shows the main menu interface of the game. The user can retrieve the difficulty level of the game and choose different modes.

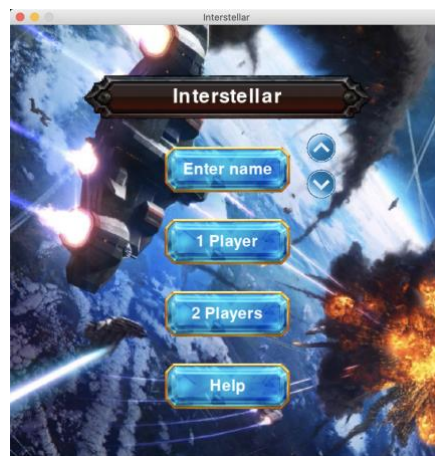


Figure.3 Splash Mode

Citation: Background was downloaded from: <http://pictures.4ever.eu/cartoons/digital-art/space-collision-226972> and <https://pngio.com/images/png-32423.html>

In the game, we can see player1, player2, boss with high healthy value, different normal enemies and rotating meteor, explosion animation, etc.. They are flying in different direction and speed.



Figure.4 Game Interface

*Citation: Image Materials were downloaded from:
<http://www.java1234.com/a/kaiyuan/sucaai/2016/0907/6667.html>*

After the game is over, you can choose whether to record the game results of this game and check the leaderboard!

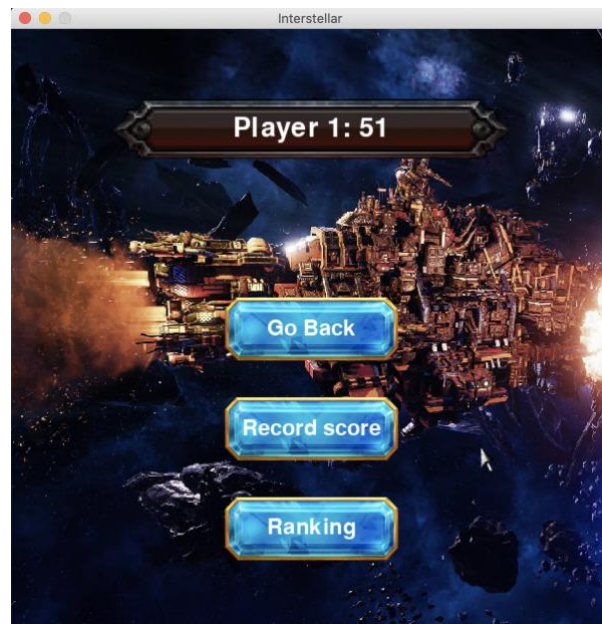


Figure.4 Different enemies with different shooting way