# Pair Programming

## Chen Fu and Chen Yang

What we had done is the development of the RobotLabel class which inherits from JLabel creating a label with the image of a robot. At first, we talked about how to develop this class for about ten minutes. Since we did not design this class in the design phase, we analyzed the fields and methods of the class and how to be invoked in other classes, then we designed methods one by one including return type, parameters passed in, and the basic logic. After that, we started to implement this class. Firstly, Chen Yang coded the fields, constructor, and a helper function for the constructor, and Chen Fu sticked her eyes on the code to check if there was something incorrect. Fifteen minutes later, we exchanged the role that Chen Fu coded the rest part of the class and Chen Yang monitored what was written until this class done. Actually, it was the first time for both of us to do the pair programming, and we find it really efficient and productive that it only took us forty minutes to discuss, develop, test and troubleshoot. More importantly, the most efficient part we discovered is one person coding and another navigating because the person monitoring on the code provides many useful reminders so that it reduces a lot of time to revise the code.

## Chen Fu and Jiahuan He

## Chen Fu and Ruida Xie

Chen Fu and Ruida Xie work in pairs to implement

## Chen Fu and Shisong Wang

We work together to implement the SetGameModeView class and GameBoardView class. To make the user interface consistent with our design documentation, we learned about the use of swing and then decided on the best implementation of each section. For example, in the SetGameModeView, we used the combination of labels and radio buttons to solve radio button cannot customize the picture.

## Chen Yang and Jiahuan He

This is the pair programming sessions of Jiahuan He and Chen Yang. During these sessions we progressed the project by developing the hashmap mechanisms to store the coordinates and players’ information after analyzing the time complexity of the software and considering the ease of use of the data type's interfaces, and we also figured out the helper methods in some model class to generate the key pairs of directions and absolute coordinates in order to support controller and view class. We've found a lot of improvements from the experience, including the skills of transferring the information of problems and discussing solutions. After these sessions, we ended up with some satisfying development experience and outcome.

## Chen Yang and Ruida Xie

Chen and Ruida paired together to implement the shooting action for a robot. We completed three methods: the shooting action method in the robot class, the update method in the game class, and the shooting helper method in the controller class. We shifted our roles for three times, and each time lasting for about twenty minutes was for implementing a method. When a person was writing the code, the other was looking at the code and checking if the code was correct. If the person who navigating the code found something incorrect or imperfect, he reminded the coding person immediately and the code was revised. By this pair programming, we saved a lot of time to implement this part because we discovered the fault as soon as possible by the navigating person. More importantly, since we paired together, we had a lot of time to discuss then learned a lot from each other not only the coding skill but also the perspectives to think and work out a problem. Therefore, we both believe it is an productive way in coding, and an effective way to learn.

## Chen Yang and Shisong Wang

In the progress of pair programming of Chen Yang and Shisong Wang, Chen played the pilot role and Shisong played the co-pilot role. We wrote code in controller class and we tried to make the whole game become alive. For doing that, we need to receive user input and handle it by setting data in model classes. Also, we need to update the UI to make user can observe the changes. When we write the logical part of controller which update the data in model classes, it is important to us to be familiar with the model classes. As Chen took part in writing model classes, he could easily figure out how to call the methods in model class to update the data. When we write the code for update UI, as Shisong participated in writing UI, Shisong can assist Chen to figure out which methods in UI classes should be called to update specific information. After the pair programming experience, we realized that two people write code together can accelerate the rate of progress especially when two people are familiar with different part of the project and we need to associate those two parts.

## Jiahuan He and Ruida Xie

Jiahuan and Ruida work in pairs to implement the RotateImage class. This class is served for robot turning, each time a robot turns to a new direction, the icon of this robot will rotate correspondingly. Thus, users can visually see the robot facing direction. At the first stage, Jiahuan is pilot and Ruida is co-pilot. Since both of us lack experience in solving similar problems, we try to search some helpful information on the internet at first. Then we decide to import Graphics2D to solve this problem. We also encountered difficulties reading robot icon, Jiahuan used read file to read robot icon but it did not work. Ruida suggested use ImageIO instead of read file, then we solve this problem perfectly. We think pair programming is an effective way to write code, we can exchange our ideas.

## Jiahuan He and Shisong Wang

In these pair programming sessions, Jiahuan He and Shisong Wang worked together on implementing the design of turn action of each robot and writing the method of run the plays of the game. In order to figure out how to make the robots able to perform turning action, we reviewed the related parts in model and view class. From these pair programming sessions, we found the significance of reviewing all related parts when dealing with interactions between different system components. In addition, we did some debug works on updating war mist and we found that pair programming is not only a good way to write code but also an efficient approach to debug existing code.

## Ruida Xie and Shisong Wang

We implemented the operations of human player such as pressing key to move, shoot, turn, and surrender. As we do not familiar with the key pressed listener and the way to handle the events enough, we were blocked by adding a listener and making it work. Then we tried to writing the logical event-handling code first by one person while the other person reading the document about key pressed event handling. When the person who were writing the event-handling code finish the logical part, the person who were reading the document took over and added the key pressed event listener and make the logical part work. From the pair programming experience, we found that it speeds up our coding progress because if there is only one person doing this work, it is too easy to be stuck. However, in a pair programming process, we could try to solve a problem while the other person is keeping coding, which would massively increase the whole coding process.

# Code Review

# Modifications

Shisong Wang and **Sophia Fu**

Shisong Wang and **Jiahuan He**

Chen Yang and **Jiahuan He**

**Chen Yang** and Sophia Fu

Chen Yang and **Shisong Wang**

Ruida Xie and **Shisong Wang**

**Ruida Xie** and Jiahuan He

**Sophia Fu** and Jiahuan He

Sophia Fu and **Ruida Xie**

Ruida Xie and **Chen Yang**