Team 13: SpinningPanda

	CMPE202 Section3	CMPE202 Section4
Name GithubID	Chu-Yuan Yang judyyang40	Teng Jin jt101015
Name GithubID	Tao Geng taogeng3	Qi Li kateleecee
Name GithubID	Yunli Wang ericclone	

Github Repository
https://github.com/judyyang40/SpinningPanda

Task Board
https://waffle.io/judyyang40/SpinningPanda

Kanban CFD Google Sheet
https://docs.google.com/spreadsheets/d/1m0W40pCr6pZkWF20apoYZwC8bKDmMZWt2C0cwzZCxw4/edit?usp=sharing

Feedback (Chu-Yuan Yang)

To achieve test-driven development, our team should be writing test cases for our user stories and testable features. We still don't have the code to pass the test, but we should have the tests written first and have failing tests in the beginning. Then as we progress along in writing our code, we will start passing more tests. Also, it will be really helpful to write tests against bugs we have encountered. The team should pay attention to the feedback our system is giving us through these tests. The tests will give us a clear indication of what functions under our design are still missing, as well as which parts of our code are broken and needs fixing. All this should take place in parallel with code development. In the long run, our team will have a decent test coverage for our project.

While deciding to use Greenfoot or Processing, we've taken into consideration if either has built-in support for unit testing like Blue. We assumed there would be since Greenfoot was built on top of BlueJ, but it turns out that neither Greenfoot or Processing has it. So, based on our state diagram of the finite state machine, we decided we will be writing our own JUnit test classes for our participating Actors in the project, which will mainly be the Player, and the "Islands".

Respect (Teng Jin)

This week, we began to design our project including how to draw the state diagram and the class diagram.

The thing I want to mention is all our team members perform excellently on respect value. During our discussion, we have a divergence on how to design our classes. In my point of view, we should firstly have an interface named IIsland and then have some class Islands to extend the interface IIsland. In this case, every islands can be regarded as a class which represents a state in the state machine. On the other hand, Yunli Wang have a contradicted idea that it would be very complicated to design a class for every islands, because the behavior of every islands is similar. So he pointed out that every island should be an object of the class Island and every object island has its own representation as a state in state machine. Even though we had different approaches on designing the classes, we showed respect to each other during our discussion. I mentioned many evidences to persuade him to follow my idea. Instead of interrupting me, he listens to what I say carefully and then talks about his opinion. Finally, I changed my mind that his idea is better.

After the project meeting this week, I realize that the respect value is a very important value during our project, especially when we have different ideas, because showing respect to others or opposite ideas can help us find a better solution of our project. Therefore, I will keep encouraging our team member to do well in respect value.

Simplicity (QI LI)

We strictly stuck to the instructions: do the necessary part first, no extended exercise. Our discussion and implementation focused on these three aspects this week:

Map drawing: Whether or not let the program to generate a random treasure map for the players? No, for easier implementation, we should first consider the big picture of the project, not the small details like how many treasure maps that we can generate. If we still have some energy left, then we can do some research on this situation. For now we just need to generate a fixed island map for the purpose.

Multiplayer process: In the current stage, we focused on implementing a single player version of the game. Later when we move forward to extend it to multi-player, the transformation would be easier for us.

Game logic: Where should we put the choosing buttons (direction A or B)? One Button class corresponds to multiple locations? Or one island corresponds to two choosing buttons? We chose the latter. We then can save the trouble of recording what are the A and B button stand for every time we arrive at an island. We may be able to also avoid some performance issues as the location list becomes longer and longer.

This week we took these small steps towards successfully implementing our project. All these decisions in the long term are great for the project. Our simplicity value is well kept.

Eliminate Waste (Tao Geng)

This week, our team discussed the design of Finite State Automata in a short stand-up meeting, and described our design through two types of UML diagrams: class diagram and state machine diagram before writing code. Using class diagram, we clearly present the classes involved in the game, their properties, and various relationships among them, while state machine diagram demonstrates the lifetime behavior of a single object (i.e., an island in our case).

Starting from a detailed description of the process by UML, we are able to use these diagrams to allow rapid and effective communication with each other and understanding of the detailed design information in a graphical manner to prevent potential time wasting. In particular, this technique is a powerful tool to “eliminate waste”, because the diagrams not only can be later followed to efficiently write code, but also enables the identification of activities without adding any value such as those unnecessary or even incorrect classes.

With agile development, we made a clear work vision to develop a game with a single player in our preliminary design. As such, its primary functions could be rapidly implemented, while multi-player game could be accomplished by adding extra small features into our current design. The iterated process (i.e., sprint in Scrum) allows us make small progresses regularly, while completing the entire project more efficiently in a given time period by making any beneficial changes immediately and avoiding low-quality and useless work.

Communication (Yunli Wang)

Our team's goal for this week is to agree on a design for the team to implement. This is a very difficult, if not the most difficult, stage during the whole project life cycle. The reason is that we are starting from scratch with so many ideas in each team member's mind. Yet we need to all come to the agreement about what we are going to do in order for us to move forward. It is especially important for us to communicate clearly and effectively during this stage. Clearly, the quality of our design will determine the end product we produce several weeks down the road. So I tried my best to facilitate a smooth communication among our team members.

During our discussion, the team seemed to be glossing over some important aspects of the specifics in the design. I think this might be due to the fact that this is a new territory for all of us and we do not have enough experience designing everything from scratch instead of following some well defined goals in a homework or lab assignment. I myself felt lost and confused from time to time in terms of what we can do or cannot do and how we can do it in such a way that it maximize the team's productivity and produce the best end product. So what I did was I tried to ask as many questions that I think is important as possible. It seemed to elicit more discussion as the team started to discover all sorts of ambiguities that was not fully discussed. As the discussion progressed, all of us had a better understanding of each other's ideas and concerns. And we were also able to compare different approaches in a more informed way instead of just based on each other's different assumptions. In the end, we reached a initial agreement on how the classes in the world would be and how the state diagram should be presented.

In conclusion, this week's experience reinforced my belief that communication is one of the most important value in a team project. We were able to achieve our goal and produced a initial working class diagram and state diagram thanks to superb collaboration and sometimes heated but respectful arguments between team members. I believe we would be able to continue to meet our goal along the way if we keep the communication open and get a better picture of the whole project.