individual-report

individual contribution to the project :

For this group project , I'm doing the record and replay module. It's mainly recording all the changes of the game and when the player wants to replay it, the game can set the game back to the state when recorded. First commit is for record and the other two is for replay
Three important commit used:

Commit 1 : ca54063b https://gitlab.ecs.vuw.ac.nz/course-work/swen225/2020/groupproject/team26/chipschallenge/-/commit/ca54063b6ee92316e81ea2f67a5f6b87cb2a1737	Description: Constructed basic structure for saving the game. Creating a Json object using Json-Sample Library and save basic information of the game into the Json object using keyword, Such as movement and steps. And write the Json object which contains game's information into the SavedGame/record1.json file created previously .
Commit 2 : cd9f33f4 https://gitlab.ecs.vuw.ac.nz/course-work/swen225/2020/groupproject/team26/chipschallenge/-/commit/cd9f33f4d75c53f9ece12c9128c847dbb1f61f24	Description: Updated Replay class in load() method that transform the Json file to json object And read it from the object. Fixed the previous version's problem that movement reading Issue due to one piece of info can only be saved with one keyword. Save all the infos including movements, level and time into the field. And created upload Board() method that use informations loaded to update the board and return a new board that is updated.
Commit 3 : 8af54ebb https://gitlab.ecs.vuw.ac.nz/course-work/swen225/2020/groupproject/team26/chipschallenge/-/commit/8af54ebbbd0ad26a7cba82295e40f8b118042fef	Description: Updated Replay class to deal with the second character in level 2 (The bug). Which read all the movement and position of the bug from the json file. Then update all the movement to the bug with initial position and return a new board that bug's position has been updated.

ranking of the contribution:

- Felix Kain (kainsamu)	kainsamu@ecs.vuw.ac.nz	5
- Qing Lu (luqing)	luqing@ecs.vuw.ac.nz	5

- Daniel Marshall (marshadani) marshadani@ecs.vuw.ac.nz 5
- Oscar Sykes (sykesosca) sykesosca@ecs.vuw.ac.nz 5
- Ruiyang Zhang (zhangruiy) zhangruiy@ecs.vuw.ac.nz 5
- Yun Zhou (zhouyun) zhouyun@ecs.vuw.ac.nz 5

a.What knowledge and/or experience have I gained?

1. learnt how to use version control systems such as gitLab and gained experience with tracking code history, source code management(pushing, merging, pulling ...), issue tracking system…...to collaborate with my team members . Including reverting in gitlab , if someone accidently pushed bad code , we can revert a previous version to avoid crushing the project.

2. I gained experience with how to write my code with a design pattern and code contract such as the iterator pattern and observer pattern.

3. I also gained experience to write my code following the design principle,making my code to be high cohesion and low coupling , means that each module should be independent and there should not be any on-permitted outgoing dependencies in my module. Following the design principle will make the project easier to add new features and avoid too many changes to other module when adding new feature to the project.

4. Because my module is record and replay, so I gained a lot of experience with json objects , I learned how to save the current game state into a json file such as how to create a json object and save all the information of the game into the json object.

5. There are lots of different json libraries out there, I gained experience of picking the right and appropriate library that suits my situation and need for my project.

6. I also gained experience of dealing with files such as creating a json file in java and how to convert a json object into the json file and how to extract a json object from a json file.

7. Gained more experience with finding and solving bugs using spotbugs and gained more experience with javadoc comments.

b.What were the major challenges, and how did we solve them?

1. The first major problem for me is creating a Json object and writing the Json object to the Json file that I created. There wasn't a method in the Json library that I first used that is able to do it . From what we learned, Json is a standard text-based format for representing structured data based on JavaScript object syntax. So to write the content of the Json object into the Json file, I should find a method that convert a Json object that stores the saved game into String object (because the json is text based format) but the library given do not contains a method that able to convert

Json object into Json String, so I have to go online and found another Json library(org.json.simple) that contains the method and solved it using method within the library then I’m able to write the information that I stored into the Json file using :Files.*write*(Paths.*get*(filePath),obj.toJSONString().getBytes(Charset.*forName*("UTF-8 ")));

2. Another problem that we’ve faced is when I apply replay , my replay class reads the Json file (saved game) and should be able to return the board and the time left for the saved game to apply replay. The problem we had is that we can only apply auto replay since the replay class is returning an updated board , we cannot really replay step by step in the replay class because the record and replay module should not have any connection with the rendering module due to MVC design pattern. We fixed the problem and applied replay step by step by collaborating with the application module, which the record and replay module only return the board step by step and the application module can apply rendering the returned board step by step to make replay step by step work.

3. Another major challenge that we had is when applying record and replay on the second level because there are more than one character in the second level, we added a bug in the second level as the enemy. My previous code logic is to record every move of the hero for record and apply the movements in replay but with the extra enemy character, the board after replay is not the same as the one that has been saved because the move of the enemy character is randomly generated . My original thought is instead of just storing the movement, I can store the whole board object in a Json file with the Type-Reference of the object, but at the time , I did not have enough time to apply it because it was close to the due date. So I decided to also record the second character’s movements and apply both of the character’s movements at the same time when replaying. In this way, it also worked out.

c.Which technologies and methods worked for me and the team, and which didn’t, and why?

1. We created a facebook group for the group project in which our group members could share ideas and report problems about the project. A group chat is making the whole communication thing easier and I think it is the most helpful thing because one of our group members is taking the course in distance.

2. The spot bug plugin was used by all the team members, which will spot all the potential bugs in our code and for us to solve to avoid potential exceptions and errors, it was really helpful for all the group members.

3. We also decided that we have an in person meeting that discusses the project every week ,which worked very well because there are always going to be some issues that need to be discussed in person, sometimes it’s really hard to get the problem clear by facebook.So in person meeting is very helpful.

4. We also used the issue tracking system in the gitlab, which me personally didn't find it very useful compare to instant message group ,because that you can only see the issue pop up when you are in the gitLab page or doing the project or checking the email, if you are doing something else , sometime you won't even realize there is a new issue. But it's a great thing that we can only assign the issue to the one that needs to know so we do not have to disturb other group members.

5. I personally used the Javadoc plugin to discover all the missing java doc comment and automatic form the comment for me, I just need to modify the auto formed comment to make it more specific .

6. We used the IDE reference browser to find the non-permitted outgoing dependencies in our module and very useful to check if our dependencies meet the requirement.

d.Discuss how you used one particular design pattern or code contract in your module. What were the pros and cons of using the design pattern or contract in the context of this project ?

Also the iterator pattern was definitely used in my module as well, no matter in saving the game or replaying the game, The java offers some "syntactic sugar" for the iterator pattern: the r(T next:collection) {..} syntax is mainly used in my design of code. I used it to iterate through containers,parsing the Json file such as iterating each move and saving them into the Json file .

The pro of the iterator pattern is that this can solve the problem when we walk through two different data structures at the same time in some complex way ,then the normal iterators can actually get in our way, using iterator pattern is fixing this problem because the iterator pattern is providing a generic way of iterating over a collection independent of its type.So the pro is basically providing a unified interface for different collection structures, enabling the same algorithm to operate on different sets of structures since it won't be depending on the internal structure of the collection. In the context of the project, each time when saving and replaying the game, the movements and changes of the game stored in the file is different each time, so if we are iterating each information by step, it is easy to cause the index out of bounds error and the iterator pattern will fix it.

The con of the iterator pattern is that it will increase the complexity of the system to some extent but I did not find it very obvious.

e.What would I do differently if I had to do this project again?

1. First few commits that I made were not really meaningful because I wasn't really familiar with the gitlab system and caused a bit of problems with only pushing the code but didn't push the library imported . If I had to do this project again , I would definitely avoid the beginner issue.

2. I personally did not attend the first two group meetings because I've been busy at other courses, should definitely attend them next time and get more involved at designing and discussing the constructor of the project.

3. I would use the issue tracking system more since that I did not really use it that much but the issue tracking system is more specific because it can state label , weight and type to the issue, so when the issue is assigned to someone , the assigned team member can clear see the type a of the problem and know how much the problem is weighted to know the priority of the issue .

4. For my module, in the record part ,my current code logic is to save all the movements and changes of the game in recording and apply the changes saved when replaying. If I get to do it again, I would find a way to save the board object at each state using the type-reference for recording because as the character number in the game increase , it is getting harder and harder to record every character's movement, but if I can record the whole board object at each states , then I would not need to worry about the increase of the character number.

5. I would apply more design pattern within my module, such I did not apply the composite pattern in my module but as the game getting more complicated , it would be much easier for me to apply composite pattern configure each character's changes by access to composites and leafs through common interface.

f.What should the team do differently if we had to do this project again?

1. I think that our team has no problem in communication with other members except one of our members is in China and he has to use VPN to communicate with us through facebook so to solve this problem I think it will be better next time if we can all use the issue tracking system a bit more next time .

2. The monkey test part of our group project was not very complete at the end , the monkey test was only covered the maze module and the application module was not covered at all because the monkey test author is in China and cannot really talk over issues of the application module with us so we should definitely give him more support if we can do the project again considering the inconvenience of communication for him.

3. The UML diagram was not generated by us because we did not figure out the UML doclet, so we cannot be able to check the structure of the project, and next time we should definitely form the UML diagram because it is very important for us to visually see the structure of the game to discover potential threat and bug with our structure.

4. I feel some game operating logic in our project did not make much sense , for instance the enemy character in level only moves when the hero moves, which is weird because if the hero do not move then the hero will never be threatened by the bug, we can set the bug to move randomly in a particular area of the maze by it self instead of move with the hero.

5. Also , maybe next time we will make the game more complex and more playable if we have more time then we might add visual effects and other layers to make the game more interesting.