**Project Sprint #5**

The main tasks of this assignment are:

1. Adding the feature of recording a game into a text file. The user story and acceptance criteria of both record and replay are required, but the implementation of replay is for extra credit (up to 2 points in the weighted total).
2. Conducting a code review exercise.
3. Summarizing the lessons learned from Sprint 0 through Sprint 5.

The following is a sample GUI layout of the final product, where “Replay” is optional.

|  |  |  |
| --- | --- | --- |
| SOS Icon  Description automatically generated Simple game Icon  Description automatically generated General game Board size  8 | | |
| Blue player  Icon                              Description automatically generated Human  Icon  Description automatically generated S  Icon  Description automatically generated O  Icon                              Description automatically generated Computer | Chart, line chart  Description automatically generated | Red player  Icon  Description automatically generated Human  Icon  Description automatically generated S  Icon  Description automatically generated O  Icon  Description automatically generated Computer  Replay |
| Record game | Current turn: blue (or red) | New Game |

Figure 1. Sample GUI layout of the final product

**Total points: 16**

1. **Demonstration (6 points)**

Submit a video of no more than 8 minutes, clearly demonstrating that you have implemented all the features in the following table. In the video, you must explain what is being demonstrated.

|  |  |
| --- | --- |
|  | **Feature** |
| 1 | A complete simple game of two human players is recorded |
| 2 | A complete general game of two human players is recorded |
| 3 | A complete simple game of human-computer players is recorded |
| 4 | A complete general game of human-computer players is recorded |
| 5 | A complete simple game of computer-computer players is recorded |
| 6 | A complete general game of computer-computer players is recorded |

If you have implemented the “replay” feature for extra credit, you should include its demonstration in the video.

1. **User Stories and Acceptance Criteria for the Record/Replay Requirements (1 points)**

* **User Story Template**: As a <role>, I want <goal> [so that <benefit>]

Add or delete rows as needed.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **User Story Name** | **User Story Description** | **Priority** | **Estimated effort (hours)** |
| 20 | Record any type of game with any number of real or robot players | As a player I want to be able to record any type of game with any board size, and any number of robot players, so that after the game is complete I can save all the moves made into a text file so that the game can be saved and reviewed with all the appropriate board settings, game settings, and moves made. | 1 | 1 |
| 21 | Replay any type of game with all board, game, and player settings that the original game had. | As a player I want to be able to select a txt file of a saved game that I have viewed, and have the GUI replay the saved game, so that I can watch past games that have already taken place with the correct board settings, game settings, and moves made. | 2 | 1.5 |

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story ID and Name** | **AC**  **ID** | **Description of Acceptance Criterion** | **Status (completed, toDo, inPprogress)** |
| 20 Record any type of game with any number of real or robot players | 20.1 | AC 20.1 Not Recording a game  Given, the record game check box is not selected  When, the game starts and completes  Then no moves should be recorded to an output file. | completed |
| 20.2 | AC 20.2 Recording a game  Given, the record game check box is selected  When the game starts  Then the board and game settings will be saved into a txt file along with the order and details of every move made for that game. | completed |
| … |  |  |
| 21 Replay any type of game with all board, game, and player settings that the original game had. | 21.1 | AC 21.1 Successfully replay any valid txt file  Given the player selects the replay option  When they choose a valid txt file  Then the board settings should be updated to fit the settings of the recorded game and replay the game in the correct order of moves per player. | toDo |
| 21.2 | AC 21.2 Deny a replay of an invalid txt file  Given the player selects the replay option  When they choose an invalid txt file  Then, an error message will display that the txt file selected is either invalid or corrupted and the game cannot be replayed. | toDo |

1. **Code Review (2 points)**

Apply source code review to one or two most important classes (and other classes if time permits) and report the findings. In addition to looking for bugs, the review should check: (1) whether the entire project has followed the coding standard in a consistent manner, (2) whether the project has followed the design principles introduced in class, and (3) whether there are code smells that indicate the need for refactoring. The following checklists provide basic guidelines. You may add new items to each of the checklists.

Make sure your answers resulted from the code review exercise. If there is no finding for an entry, you should provide an explanation. For example, if your answer to “Are the naming conventions violated?” is no, you should describe a naming convention and present an example. You will receive no credit for this part if your answers are simply yes or no without additional information.

Classes that have been reviewed: App, Board

Date/time duration of the code review exercise:12/9/2022 @ 2:00pm – 12/10/2022 @ 11:20am

|  |  |  |  |
| --- | --- | --- | --- |
| **Checklist** | **Checklist Item** | **Findings** | |
| Coding Standards | Are the naming conventions violated? | Some naming conventions are inconsistent, most common is going to be camel case naming, however a lot of the tkinter elements it helped to use underscores as spaces to differentiate regular variables from tkinter variables. Names are all unique with no reusing vars for more than one type of item or object. | |
| Is the ordering convention of method arguments violated? | No, python is specific that self must be the first element inside of a function even though a different function doesn’t necessarily pass it. It’s still required. After that I do not use any default parameters that aren’t handled elsewhere by the code, and everything is passed in every case that I have come across. So ordering is important and upheld. | |
| Any comments meaningless or inconsistent with the code? | No comments are inconsistent with the code every comment lines up with the meaning of the code snippet, however some comments may not be very necessary other than separation of different parts of the code, to know what part of the GUI we are dealing with and what subsection of that part of the GUI is being created or altered. And in the board class there are comments under the scoring condition conditionals to know what the condition is specifically testing in more of a graphical sense than a verbally descriptive manor. | |
| Any code block has an inconsistent formatting style? | I do not believe so, all code blocks have to follow python’s layout otherwise it will not work as python is sensitive to indentations. | |
| Any indentations inconsistent? | In python indentations are very important and even being a space off can cause errors so indentations conventions are held true to python standards. | |
| … |  | |
| Design Principles | Any class/method not well-modularized? | I believe that all my classes and methods are important in the creation of the SOS game. If anything, I believe that some of my code could have been more modularized to have more | |
| Any class with poor abstraction? | No, my code doesn’t not utilize class abstractions but rather two classes that work together that represent two different aspects of the board, the logic and the gui. The gui class creates a board object to reference the board class so the GUI class doesn’t have to determine if cells are already taken, or if the game is over. | |
| Is the visibility of any variable, method, and class inappropriate? | The visibility of variables and methods and classes I believe is all appropriate, the App class which houses the GUI only has access to the Board class through methods of the board i.e getters and setters, and the board class does not have access to any variables or functions inside of the App class. There are some variables that are used for testing that are public and not relevant to the logic of the game itself but serves to satisfy tests in the testing class. | |
| Is design by contract (pre/post-condition) violated? | No in all of my situations that I have been able to test for my pre-and post-conditions have been handled. In all situations where user input is allowed there is usually a minimum and maximum that has been implemented and if somehow a value outside of that range is detected the function will return the nearest valid option. Even for options where no option may be selected such as selecting simple or general game, if nothing is selected a default will happen where simple is selected. | |
| Is the Open-Closed Principle violated? | No the open closed principal is not violated, there are no classes that extend each other only classes that work together. All classes could be extended if need be but cannot be altered outside of the main class itself. | |
| Is the Single Responsibility Principle violated? | This may be violated some, inside of my board class I have some functions that multiple purposes such as generating the board space the robot will be choosing and generating the token that the bot would be placing. This could have been separated into multiple classes for better readability and to diversify aspects out more. | |
| Code Smells | Are there magic numbers? | There are no magic numbers, there is not a lot of calculations needed other than incrementing the score. And all other numbers are selected and limited. The only use for them is updating the GUI size and layout, but just a reference to them is needed no math involving them. | |
| Are there unnecessary global / class variable? | There are some variables that have been used strictly for testing purposes only. Which I do not love that I have, but as I have been learning more about testing frameworks and methods, through out the semester I was able to include less and less instances of needing to use a variable for testing. Other than that I believe all my variables are used at least once if not more than once for the same purpose. | |
| Is there duplicate code? | Yes there is some instances of duplicate code, the primary example would be inside my App class, the clicked function and the RobotMoves function have very similar internals that could have been broken down into functions that each could call, but at the time I needed a method to separate robot automated decisions from human stop and wait decisions that occur. The robot function is a more bare bones version of what goes on inside of the clicked function, however there is still aspects that could be broken into some more functions. | |
| Are there long methods? | Yes the longest method I have is my “buildMainGame” method inside of my App class. This is responsible for putting everything on the window, all the widgets and all the text. This probably could have been broken into multiple functions to help organize and would have helped with the reset function, however I was learning tkinter for the first time and this was a method that worked for the time being, if I had more time this would probably be one of the things I change. | |
| Is there any long parameter list? | No I do not have any long parameter lists, in python classes the self-keyword is always required, other than the only thing passed to functions in a parameter list would be what is required the longest parameter list would be consisting of  Self, row, column, token, and player only the information we would need to keep track of to make logic of the program work. | |
| Is there over-complex expression? | No I do not believe there is any overcomplex expressions, there are not a lot of expressions needed for my program, some examples would be creating the file name if the record option is saved, is just concatenating a few strings together. Another would be in my board class when deciding the size of the board it uses a hash map to determine the window size based on the board size. | |
| Is there switch or if-then-else that needs to be replaced with polymorphism | I do not believe so, I do not use switch statements as not all versions of python support switch statements until fairly recently. And all my if statements are pretty standard, the only place I could see it being useful would be when determining if an SOS was made, where I have a large branch of if, elif, and else determining if any pattern of SOS has been made. | |
| Any variable or method name whose intent is unclear? | I believe all variable names have been chosen carefully and are fairly descriptive of what they should do. All tkinter elements are labeled such as player1\_label or game\_mode\_var to make sure everything is unique and descriptive. I guess some of my variables may not be unique enough such as to keep track of the current player I use a variable called Player, which could be updated to maybe use the name currentPlayer to make it more obvious what it represents. | |
| Any similar methods in different classes? | I believe that all methods are very unique to each class as the App class contains everything related to the GUI, and the board class contains everything related to the logic behind the game. | |
| … |  | |
| **Bugs** | **Buggy code snippet** | **What is the bug?** | **Why is it a bug?** |
| robotMoves Function | Invalid win | Every now and then the game thinks someone has scored but the robot hasn’t made the next move yet on the GUI so the logic doesn’t visually make sense of how someone won. |
| Reset function | Two robot players infinite loop | If both players are robots and the game ends there isn’t really a way to stop the game from ending unless you terminate the terminal underneath the game. |
|  |  |  |

1. **Summary of All Source Code (1 points)**

|  |  |  |
| --- | --- | --- |
| Source code file name | Production code or test code? | # lines of code |
| Gui.py | Production | 435 |
| Board.py | Production | 242 |
| Start.py | Production | 13 |
| Tests.py | Test | 168 |
| Total lines of code | | 858 |

**You will receive no credit for this assignment unless your complete source code is submitted.**

1. Summarize the lessons learned from the entire project by answering the following questions from the perspectives of development processes, coding, design, refactoring, and testing **(6 points)**:

* What did you personally gain from the project?
* What does your project do well, and what could your project do better?
* How could you improve your development process if you develop a similar game from scratch?

Minimum requirement for (5): One full page single spaced, font size no bigger than 12 points.

Throughout the 449 course I truthfully learned more than I thought I was going to. I had been told by friends who had taken the course before that the class was easy and you don’t have to try that hard that it was just another class you must take. So, I honestly went into it with not the highest of expectations. However, that changed quickly in my mind. I do not believe the course is designed to be hard but one thing everyone failed to mention was how education that it would truly end up being. I had heard of some of these things we have learned in class like scrum and test vs production code, but I never was really exposed to them until I took this course, and I learned quite a bit about the development process, coding, design, refactoring, and testing.

I have personally gained a lot of knowledge about all these topics throughout the course. When it comes to the development process, I had no idea how in depth the planning and organization of everything was. From the amount of people who are involved to having different roles. I originally would have just thought that most companies just have tasks for a couple of engineers to do and they just go work on it until they’re done with it. But it goes into a lot more detail than just that. Having meetings about what has been accomplished, talking about what’s holding your team back. From even activities like scrum poker where we talk about how long we expect something to take us and try to find middle ground between all the proposed time. Most of my time I’ve spent coding in the past was on personal projects where I just thought of an idea and went with it until it was complete. So, I really gained a lot from this dividing into separate parts idea and going in an order. I will greatly benefit from a more planned out schedule verses just doing something and then something else without thinking. I feel as if I did improve in my coding over time as I was building a project that took the whole semester, I was learning new techniques and learning the tkinter framework better overall the practice just really helped me out. For design, tkinter is very primitive but it taught me a lot about practicing structure and learning how different GUI aspects can interact with each other. Refactoring was a very useful tool to learn to do this semester. For the first group of projects my SOS game was just meh, everything from GUI to logic was inside of one class over a couple methods and it worked but it was efficient. It wasn’t until we started talking about refactoring in class, I thought of some ideas that could improve my project, the main idea being how to break up the logic and GUI successfully, that was by far one of my biggest breakthroughs for this project. Finally, I gained a lot from testing. I remember talking about unit testing in CS101 where we used the build in unit test framework in python. At the start of the class my tests were not that great, just scenarios where they couldn’t really fail, however as I became more comfortable with assertions and understanding what a good oracle would look like, I was able to write tests that would utilize my logic for my classes and test real scenarios that would occur.

There are a lot of things that my project did well. When designing the goals for my project I was pretty accurate with development timing and the processes. I was able to whip up a quick concept and build on it to make it better over time. For coding, I think overall it was well it meets all the requirements for the project but there are few quality-of-life things that tkinter struggled to handle that I would like to have changed if I had more time for the project. For starters, my project still doesn’t include lines when a SOS is created. This is something that I would have liked to have added, but sadly tkinter being very primitive will not let me draw lines over widget objects so that option would not have worked. Another idea I played around with was changing the color of the buttons, however I had to use tkk buttons for this project so I could update the text on the buttons because regular tk buttons don’t allow you to add text on them after they have already been created. The issue is tkk buttons don’t allow a single button to have a single color, all buttons are the same color all the time. Switching back and forth between tk and tkk buttons could have been an option if it didn’t break the user interface. Sadly, there wasn’t an idea that I could come up with that would work with my current build, in the next attempt at this project I would have chosen maybe a different language with a different framework to build a GUI, one that is a bit more modernized. Another thing in the coding I would have liked to have added but with the current build was not doable through tkinter, would have been multithreading for the human player and for the robot player. Multithreading would have allowed me to use what I have learned from my operating systems class to use locks to force waiting for the robot to place its piece. This could have helped me include time in between robots move to make it more natural and readable. Instead of everything just appearing all at once. As for the design I’m personally very happy with how the layout worked, everything is responsive and works for all game board sizes. Refactoring as I mentioned I am very happy with because it helped improve the quality of my game a lot by adding the second board class to the main GUI class, because it allowed me to separate logic and visuals. And I am also very happy with my testing code, as I learned a lot throughout the semester on how to write better and better tests that truly do measure if something is working the way that it should be.

I think the best way I could improve my development process would be to diversify my self out and try a new language. I have proved that I can make a project in python that meets the minimum needs for my project, however there is more that I would like to do that python mixed with tkinter could not accomplish. I would take more time in the development process and look at more frameworks and other languages to decide what could be best for my goal. For example, this was supposed to be an object-oriented project, which mine was, but it was very difficult to do with tkinter and python. Maybe if I had used pygame instead it could have gone better since that is made for actually making video games inside of python and can create some really amazing projects, but still for an object orientated language I think Java would have been the best way to go, but since I didn’t know java at the start of the semester and I didn’t have a lot of free time this semester I had to make a lot of compromises. So overall for my development process if I were to make a different but similar game, I would have taken more time to look over more frameworks and tried to learn them instead of just relying on something that I have a little experience in.

I really appreciate all the skills that I have learned throughout this semester, I have a job lined up working with business applications, more specifically helping clients with Oracle NetSuite, at an accounting firm and part of my job will be writing scripts and code for their needs inside of NetSuite. I also have the goal of moving up in the company and transferring to the Application Development team where I will do even more coding in many languages. These skills are valuable, and I see how I will be using them every day in my career.