HW5: Individual Assignment (130pts)

Replace all the highlighted placeholder text, below. You may change the style of this template, but please provide all the responses requested and keep the same sections / order.

**Questions?** Ask on Piazza or send me an email.

What is **your name**?

|  |
| --- |
| *Adrien Protzel* |

Which **team** are you on?

|  |
| --- |
| *Team Bajas* |

What is **GitHub username**?

|  |
| --- |
| *AJ-Protzel* |

# Sprint 1 Contributions (45pts)

## User Stories (30pts)

What were your contributions to the project user stories during Sprint 1? **Be specific**.

|  |
| --- |
| *Added base board game*  *Added varying board size*  *Refitted color scheme*  *Bug fixes for end game and move functionalities* |

**What specific user story subtasks** were assigned to you for Sprint 1? Provide Asana links to the user stories showing subtasks assigned to you.

|  |
| --- |
| [*https://app.asana.com/0/1182325912594614/1182525247777931/f*](https://app.asana.com/0/1182325912594614/1182525247777931/f)  <https://app.asana.com/0/1182325912594614/1182525247777935/f>  <https://app.asana.com/0/1182325912594614/1182525247777933/f>  <https://app.asana.com/0/1182325912594614/1182525247777953/f>  <https://app.asana.com/0/1182325912594614/1183438829665065/f> |

**Where** can **evidence of your** **contributions** be seen? If you contributed code, provide a GitHub link.

|  |
| --- |
| <https://github.com/Bshah11/Team-Baja-OSU-CS361> |

## Spike (15pts)

**What spike** did you do during Sprint 1?

|  |
| --- |
| *Learn flow of GitHub* |

What were the **results of your spike**? What did you find out? What did you conclude? Be specific.

|  |
| --- |
| Using github desktop, the main point you look at are the "Current branch" in the top middle, the "fetch/push origin" to the right of it. The "Summary (required)" on the bottom left.   The current branch shows which branch you are working on, you can click it to create a new branch or if you’re on the master branch you and click "merge" at the bottom of the drop down.  "Fetch origin" refreshes the current branch with what is currently loaded on github That button will be replaced with a push or commit prompt if you have changes on your current branch.  When you have made changes on your branch, add a header to the "Summary (required)" describing what you did.  in order your should: create new branch or open master > fetch origin if you want the latest github information > work on your code > add a summary and click commit under it > (if not working on master branch) click switch to master branch > click merge into master branch > click the branch you want to merge > click commit next to current branch > profit |

# Class Diagram (25pts)

**Instructions**

* Imagine your project is **finished** and **has classes**.
* Create a **UML class diagram** for this potential finished project.
* The diagram should have at least **4 classes**.

|  |
| --- |
|  |

# Sequence Diagram (25pts)

**Instructions**

1. Imagine your project is **finished** and **has classes**.
2. Create a **UML sequence diagram** showing interactions between classes and users/actors, for this potential finished project.
3. The sequence diagram must include at least **4 columns**.

Note: It may help to start with a use case you wrote earlier in the term.

|  |
| --- |
|  |

# Evaluation and Revision (25pts)

**Instructions**

1. **Evaluate** **your class diagram** based on three **SOLID** design principles.
2. Describe **how the design could be improved** based on each principle you choose.
3. The evaluation **must lead to diagram changes**.
4. **Modify** the diagram based on your evaluation.
5. **Share** the revised diagram with your team in Asana.

|  |  |
| --- | --- |
| **Principle** | **Evaluation** |
| *OCP* | *No matter the board size, the same functions work.* |
| *LSP* | *The game boards are updated separately.* |
| *DIP* | *The current Game only needs to know the score total and if the game == end. But does not need to know the cells individual values.* |

**Revised diagram**:

|  |
| --- |
|  |

# Rubber Duck Debugging (10pts)

How did RDD go this week? For full credit, write at least a paragraph about your experience. You can write more if you want to.

|  |
| --- |
| *We were able to talk through the possible bug fixes that had arose from the code refactoring into a class structure. Talking out load I was able to read each line of code and walk through step by step and see what each value stored, where a function was called, and what information was passed around. This game me a firm understanding of the flow of the code and at what points additional user stories could be implemented.* |

## 