

# Board Game Spec Sheet

For this project, you will be required to replicate one of your favorite board games. You will be required to understand the basic rules of the game that you choose and you will be responsible for creating a basic interface that allows users to play the game.

The primary objective of this project is to introduce you to the practice of software development. We do not expect you to create something that hasn't been done before. Instead, we want you and your teams to focus on thoroughly designing and executing a well thought out program. This will include thorough testing and documentation. By reading through this document, you should have a full understanding of the complete scope of the project and its requirements.

## Time Requirements

As an undergraduate student, we understand that you are in other classes as well. That being said, we expect that you will be able to spend approximately 8 hrs per week outside of class working on this project. It will be your responsibility as a team to allocate tasks accordingly.

You may of course make choices based on your own time and skill constraints, as well as those of your team overall — this may materialize as choosing to do a simpler board game, and is a perfectly valid decision.

## Milestones

Throughout the quarter, we've created a set of milestones that we can use to ensure that everyone is making progress on their project. We feel that it is important to develop everything in a way that ensures you design and test the code to a standard that would be expected in an actual development process. Here are the milestones we have created for you.

Week	Objective
Week 2	Formulate and Organize Teams
Week 3	UX Design and Prototyping
Week 4	"Hello Players"
Week 5	Player Tokens
Week 6	Iterate Through Player Turns
Week 7	Dice Rolls and Movement
Week 8	Implement Rules

Week 9	Game Completion
Week 10	Sharing/Evaluating Projects

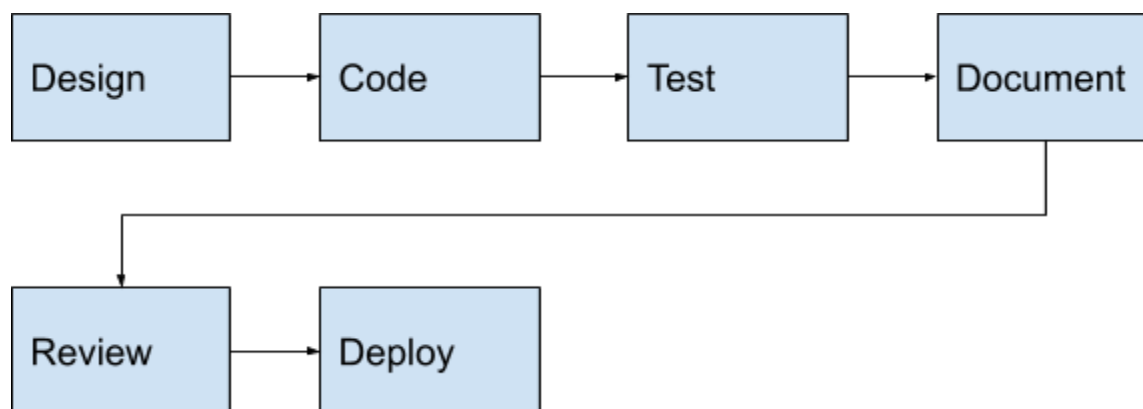
In order to better match the process of industry documentation, we will ask you to use a program like Notion, and track all your milestones there, as well as creating documentation for your own project.

#### Each Week

- Iterate over the pipeline
  - Add testing: unit, functional
  - Versioning: deployments to different stages, environments, etc
- Produce Documentation
  - Depending on the week, a mix of:
    - Use Cases
    - UX Designs
    - Testing Plan
    - UML Diagrams
    - System Design
    - Code Generated Documentation
- Game Codebase
  - Iteratively building out the game's feature set

## Pipeline

For each milestone we want the teams to follow a build pipeline when developing their app. We will give them a bit of leeway in how they set up their pipeline but in general it should look something like this:



# Rubrics

## Week 2: Formulate and Organize Teams

	1	2	3	4
Workflow plan	No plan or vague notions to “communicate over slack”			Detailed plan on how often team will meet, how team will communicate, how issues & tasks will be tracked, and how workflow/progress will be periodically evaluated.
Team Organization	No organization / roles put forth			Team leader/ spokesman decided. Team roles/ division of responsibilities put forth.
Org Level Documentation and Tools	No tools chosen			Tools chosen that best suit the workflow, and tool chosen for documentation.

## Week 3: UX Design and Prototyping

The goal for this week is to have the rules of the game figured out. Additionally, you must show work you have put in towards choosing the set of technologies you will be using to develop this board game, and the processes your team will adhere to when building it.

You will be scored out of 4 for each of the categories below.

	1	2	3	4
Game	Game has been			Rules have been

Rules	chosen, but the rules are mostly in here 🧠			written out, and are documented in the team's org level directory
Wireframes	Team has only a general idea of what the UI will look like. Many components not specced.			All screens have a detailed wireframe including popup modals and interactive components. All parts of the game can be simulated on paper printouts.
Documentation Plan	No plan or merely general notion to "keep track of things" or "comment in the code"			Detailed plan on what types of things will be documented (code hierarchy, design choices, build instructions, etc.), how they will be formatted (UML, documents, etc.) and where they will be kept.
Testing Plan	No testing plan or plan just consists of a few test cases for each milestone.			Detailed plan including test cases for each milestone, what testing framework will be used, and policies regarding unit tests, failed tests, and testing frequency.

## Week 4: Hello Players

The goal for this week is to do a dry run on your build pipeline and end the week with a hello world equivalent of your app. For this stage you will have both the front and back ends functioning in a minimal state with communication between them. Your app in this stage will initialize the front end, draw the board on the screen, make an API call to the back end to retrieve the hello world text, and then display that text in a dialog box.

You will be scored on the following items.

	1	2	3	4
Functionality				
Game board	App does not draw a board or board is incomplete			App draws a complete game board.
API call	App does not call the backend or the API call fails.			App successfully calls backend and retrieves hello world text
Hello World Message	App does not display the hello world message			App successfully displays the hello world message from the backend
Testing				
Unit Test getMessage()	getMessage does not return hello world text			getMessage returns text hello world
Integration Test API call	GET request fails or returns the wrong thing			GET request to /msg returns hello world text
Functional Test Hello World	No message is triggered or message does not say hello world			Launching the app triggers a hello world message
Documentation				
Build & Deploy Process	Not documented or only weakly described.			Detailed description of how to

				build/run/deploy app.
Code Generated Docs	Sparse or nonexistent documentation of code			All functions and classes have a meaningful docstring. Docs effectively convey purpose of functions and how to call them.
API Documentation	API paths are not documented			API paths are documented with a clearly defined purpose and correct usage for each

## Week 5: Player Tokens

The goal for this week is to draw a player token and place it on the board. Your pieces don't need to move around the board but you should be able to place it on any square on the board. Additionally you must show us that you've tested these components and documented your design.

You will be scored out of 4 for each of the categories below.

	1	2	3	4
Functionality				
Player pieces	App cannot draw player pieces or pieces are generic place holders.			App draws player pieces with sprites. Player pieces can be assigned any position on the board.
Testing				
Unit Tests for Game Board/game model	Tests are weak / nonexistent			Tests are thorough, have good coverage, and are passing

Unit Tests for Player pieces/player models	Tests are weak / nonexistent			Tests are thorough, have good coverage, and are passing
Functional Test for Player Pieces	No Functional test described for showing player piece			Detailed test for showing a player piece
Documentation				
Data Models	No class hierarchy defined			Class hierarchy for data models is documented.
Code Generated Docs	Sparse or nonexistent documentation of code			All functions and classes have a meaningful docstring. Docs effectively convey purpose of functions and how to call them.
API Documentation	API paths are not documented			API paths are documented with a clearly defined purpose and correct usage for each

## Week 6: Iterate Through Player Turns

The goal for this week is to implement turns for players. Nothing needs to happen during each turn — you simply need to allow each player the ability to click a button that ends their turn and begins the turn of the next player. You should have a visual indicator for whose turn it is, and UI reflective of the player's status.

	1	2	3	4
Functionality				
Turn passing	Turn does not pass from player to player or is buggy.			Turn passes from player to player correctly.
Documentation				
Turn Sequence (including turn passing)	Not documented or only vague/high level description			Detailed description of turn sequence and the corresponding interactions between modules.

## Week 7: Dice Rolls and Movement

The goal for this week is to implement the rolling of a dice, and the movement of the player. You will want to think about “where” the modifications are made to your data model, and also create the visual aspects of the roll and the player animation. You may choose your own level of complexity for representing the roll of the dice, as well as the animation of the player.

	1	2	3	4
Functionality				
Dice roll / card draw simulation	No turn setup/ non functioning			Player can roll dice/draw card to start their turn. Action is intuitive/easy to figure out. Rolls/Draws are



				randomized.
Player movement	Player cannot move around the board or does not move according to rolls / draws.			Players can move to all squares (per game rules) and move according to rolls/draws.
Testing				
Unit tests for player movement	Tests are non-comprehensive / nonexistent			Tests are thorough, have good coverage, and are passing
Unit tests for turn passing	Tests are weak / nonexistent			Tests are thorough, have good coverage, and are passing
Unit tests for game controls	Tests are weak / nonexistent / not passing			Tests are thorough, have good coverage, and are passing
Unit tests for dice rolls / card draws	Tests are weak / nonexistent / not passing			Tests are thorough, have good coverage, and are passing. Tests ensure proper randomization.
Documentation				
Player Movement	Not documented or only vague/high level description			Detailed description of movement process / logic and corresponding interactions between modules.

## Week 8: Implement Rules

The goal for this week is to have the game rule logic completely implemented. You must complete all the game mechanics needed to play a game at the back-end. You need to think about all the cases that would happen during the game. Additionally you must show us that you've tested these components and documented your design.

You will be scored out of 4 for each of the categories below.

	1	2	3	4
Functionality				
Rule Book	The game doesn't provide rule books and doesn't teach users how to play the game.			The game gives players enough information to play a game.
Game Play	Some rules are still unimplemented for the game play. Some edge cases make the game crash.			All of the rules of the game have been implemented. All of the edge cases have been covered.
Testing				
Unit test for all the rules for the game.	Tests are weak / nonexistent / not passing			Tests are thorough, have good coverage, and are passing.
Documentation				
Rule Book	Not documented or only vague/high level description			Detailed description of all game mechanics and how to play the game.

## Week 9: Game Completion

The goal for this week is to have the game completely implemented and playable. Additionally you must show us that you've tested these components and documented your design.

You will be scored out of 4 for each of the categories below.

	1	2	3	4
Functionality				
Game Setup	Number of players / player characters cannot be chosen.			Number of players & player characters can be set at the beginning.
Game Play	Some rules/features are still unimplemented. Game needs intervention from the developer to play through.			Game can be played from start to finish without intervention. All rules/features are implemented and working.
End Game	Game does not end or ends unexpectedly. Game end message missing/incomplete.			Game ends under the proper circumstances. Winner/score board shown upon completion.
Testing				
Unit tests for game setup	Tests are weak / nonexistent / not passing			Tests are thorough, have good coverage, and are passing
Unit tests for game end	Tests are weak / nonexistent / not passing			Tests are thorough, have good coverage, and are passing
Documentation				

Setup sequence	Not documented or only vague/high level description			Detailed description of setup sequence and corresponding interactions between modules.
Game end	Not documented or only vague/high level description			Detailed description of end game and corresponding interactions between modules.

## Week 10: Sharing/Evaluating Projects