

Board Game Builder

Project Overview

This Board Game Builder is a tool that allows users to design and play custom turn-based board games. Users should at least be able to define:

- Board shape and size
- Custom pieces and their movement rules
- Tile events and special conditions
- Win conditions and turn order

This project aims to produce a general-purpose game builder where users define rules using a simple "if this then that" scripting language. Completed games are saved as JSON files that can be reloaded for future play. Unlike other board game builders online that provide only the graphics and physics simulations, our project will have enforceable rulesets rather than relying on the honor system.

Team

Aaron Cook

Sophia Culver

Landon Dennis

Jackson Fixsen

MVP Goals

- Build a working interface for creating and playing chess/checkers-like games
 - Basic rule scripting (piece movement, captures, simple win conditions)
 - Ability to save and load games via text/JSON file
 - Functional turn management system
-

Technology Stack

Area	Tools
Frontend UI	HTML/CSS (Bootstrap?)
Game Logic	Javascript
Save Files	JSON
Optional LAN Hosting	Node.js (express) / Websockets
Vector Graphics	Adobe Stock or Illustrator

Expected Users

Role	Function
Game Creator	Uses UI to design a new game and save to JSON.
Game Player	Loads a saved game file and plays it.

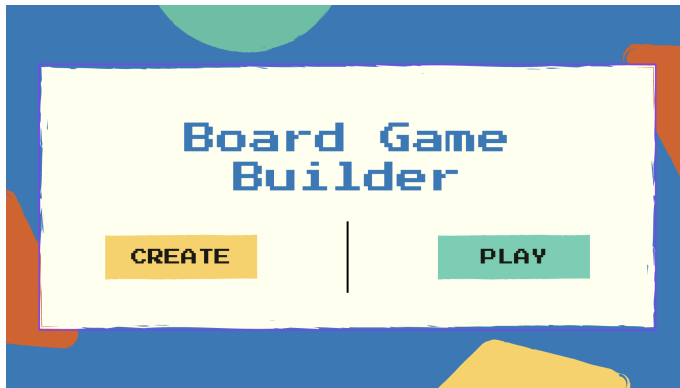
Risks & Mitigations

Problem	Description	Mitigation
Scripting complexity	It is difficult to gauge how difficult the scripting feature will be.	Prototype early and test on real possible triggers and actions.
UI / Sprites	Since we don't have a server database, allowing users to upload sprites seemed unwieldy. So how do users visualize pieces?	Use free stock images. Use text on sprites to differentiate between many pieces easily. Alternatively, allow sprite LINKS.
Scope Creep	Almost infinite feature possibilities. May get sidetracked/overwhelmed	Focus on function before polish Lock-in MVP features early
Team Bandwidth	Many dependencies between pieces of the project.	Google Chat & Weekly Meetings Also, regular Git Commits.

Wireframe Creation UI

1. Main Menu: Create or Play.
 - Click Create: Opens new Page (2)
 - Click Play: Opens Computer File Manager.
 - If json is invalid, stay on Main Menu, display popup "Invalid Code"
 - If valid: Opens Game Screen (8)
2. Create Menu: Load Game or New Game.
 - Click Load: Opens Computer File Manager.
 - If json is invalid, stay on Create Menu, display popup "Invalid Code"
 - If valid: Opens Primary Creation Screen (4)
 - Click New Game: Opens Board Setup (3)
3. Board Setup: Allows you to select grid type (tile shapes) and grid size.
 - Provides a little preview of the different tile shapes based on what you choose.
 - Boards are not extremely customizable, but should at least allow square & hex.
 - Click Create Board: Opens Primary Creation Screen (4)
4. Primary Creation Screen: Center: board, Left bar: (undo, redo, zoom, move), Top Right: save as, preview, edit elements, restart. Moveable: Any opened menus.
 - Click Save As: Open File Manager to save a JSON file.
 - Click Preview: Lets User Preview what the game will actually look like.
 - Click Edit Elements: Opens Elements Editor (5)
 - Click Restart: Open Popup giving option to save as, don't save, or cancel.
 - For any option other than cancel, this will bring us back to (3)
 - You should also be able to click on visual elements themselves!!
5. Elements Editor: Overlaid menu/form for users to add or remove elements to their game and open or hide editing menus.
 - Gives a List of elements, with the option to Add/Delete, Show/Hide, and to Edit.
 - Add/Edit: Opens element menu (6) (add will create class)
 - Delete: Warn this will remove all associated rules. (deletes class)
 - Show/Hide: show or hide the menu.
6. Element Menu: Name, sprite, color, text, classes, etc.
 - Add/Edit Rule: Opens Rule Editor (7)
7. Rule Editor: Dropdown-based scripting interface (triggers/conditions/actions).
 - Maybe select pieces instead of searching the way Excel allows.
8. Game Screen: Displays board, pieces, turn tracker, player resources, and any interactive features. Should also be a place for them to check (user-specified) rules if the user filled those out.

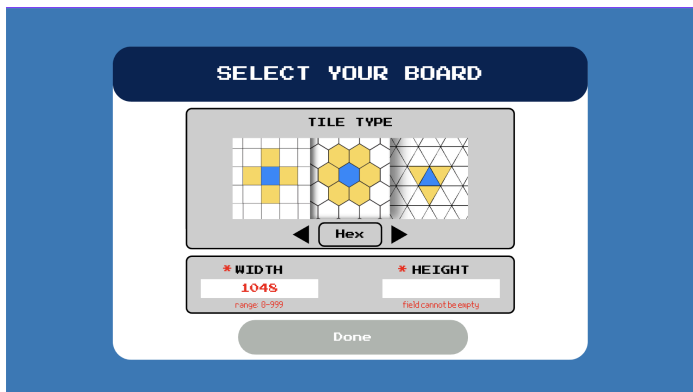
1) Main Menu



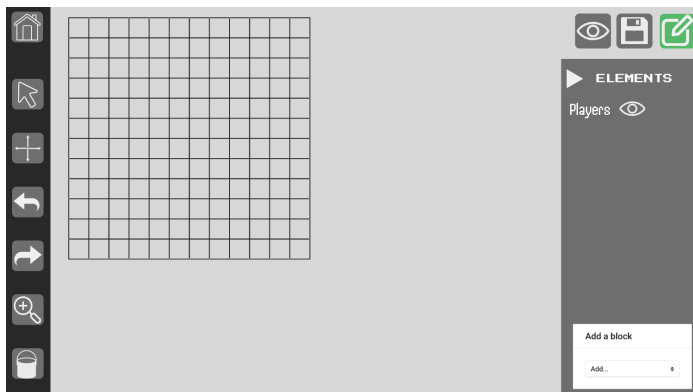
2) Create Menu (Very similar to the Main Menu)



3) Board Setup

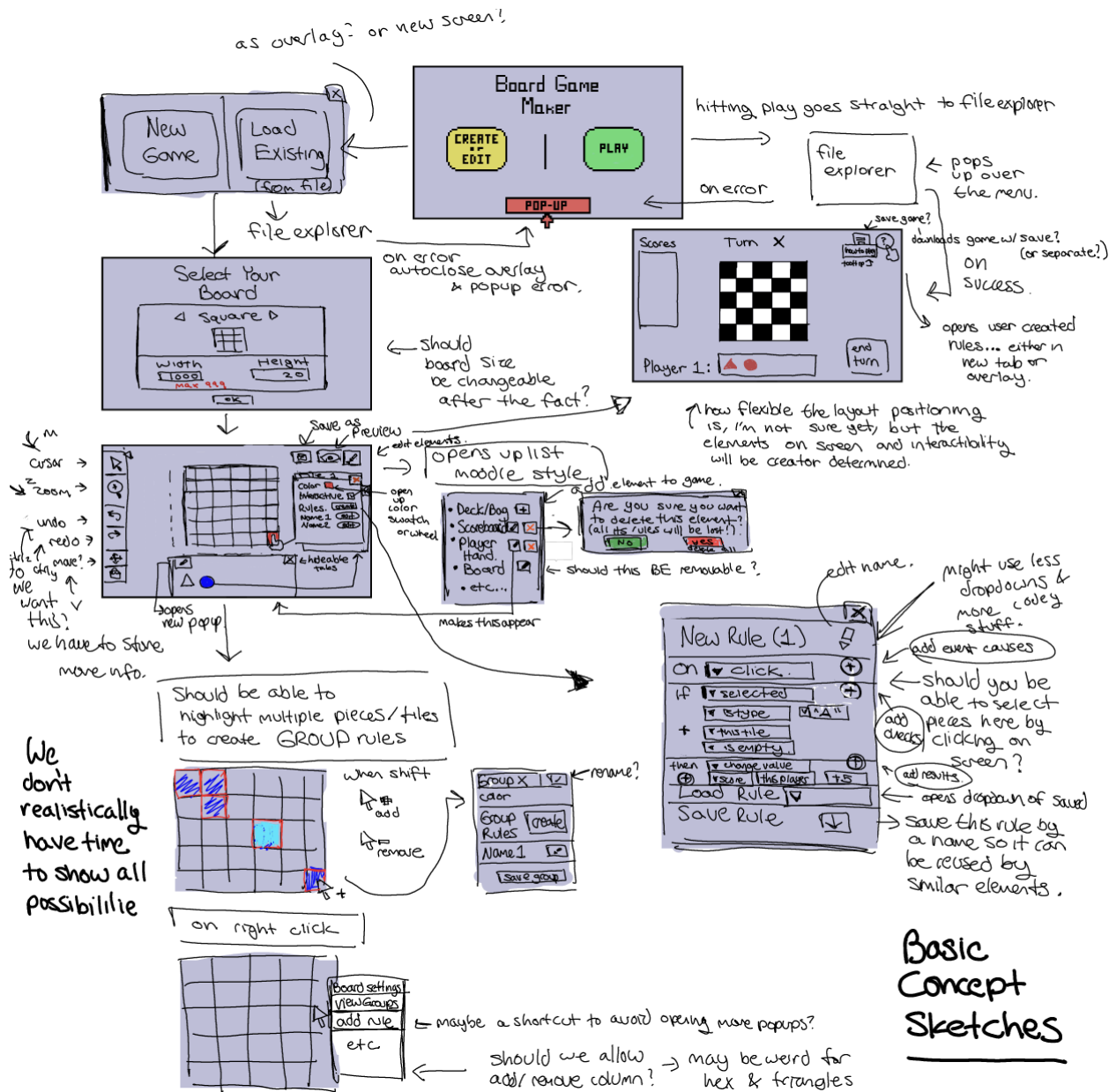


4) Primary Creation Screen (WIP)

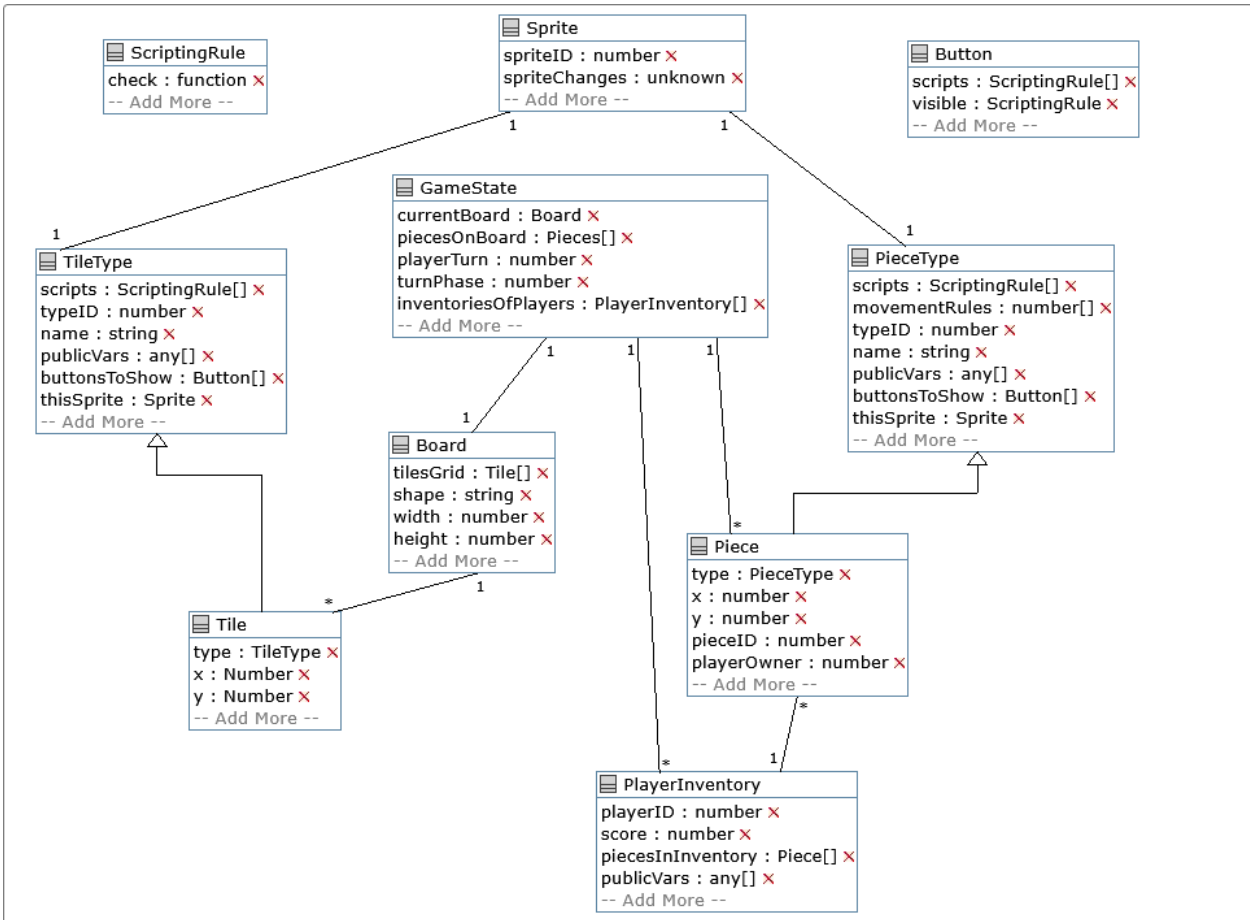


The visual element of this section isn't finished, but the original design sketches are attached.

Beta Design Sketches

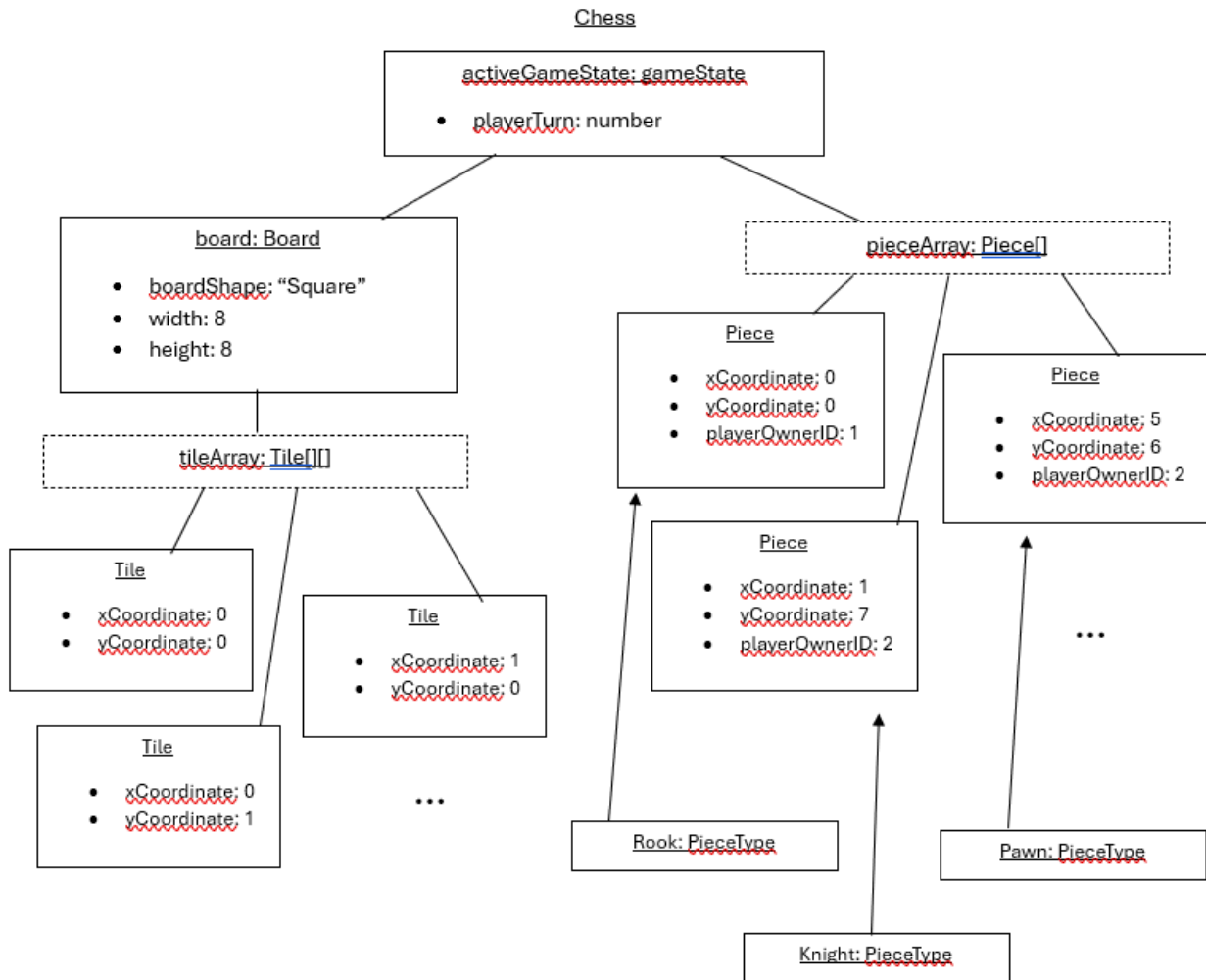


Class Diagram



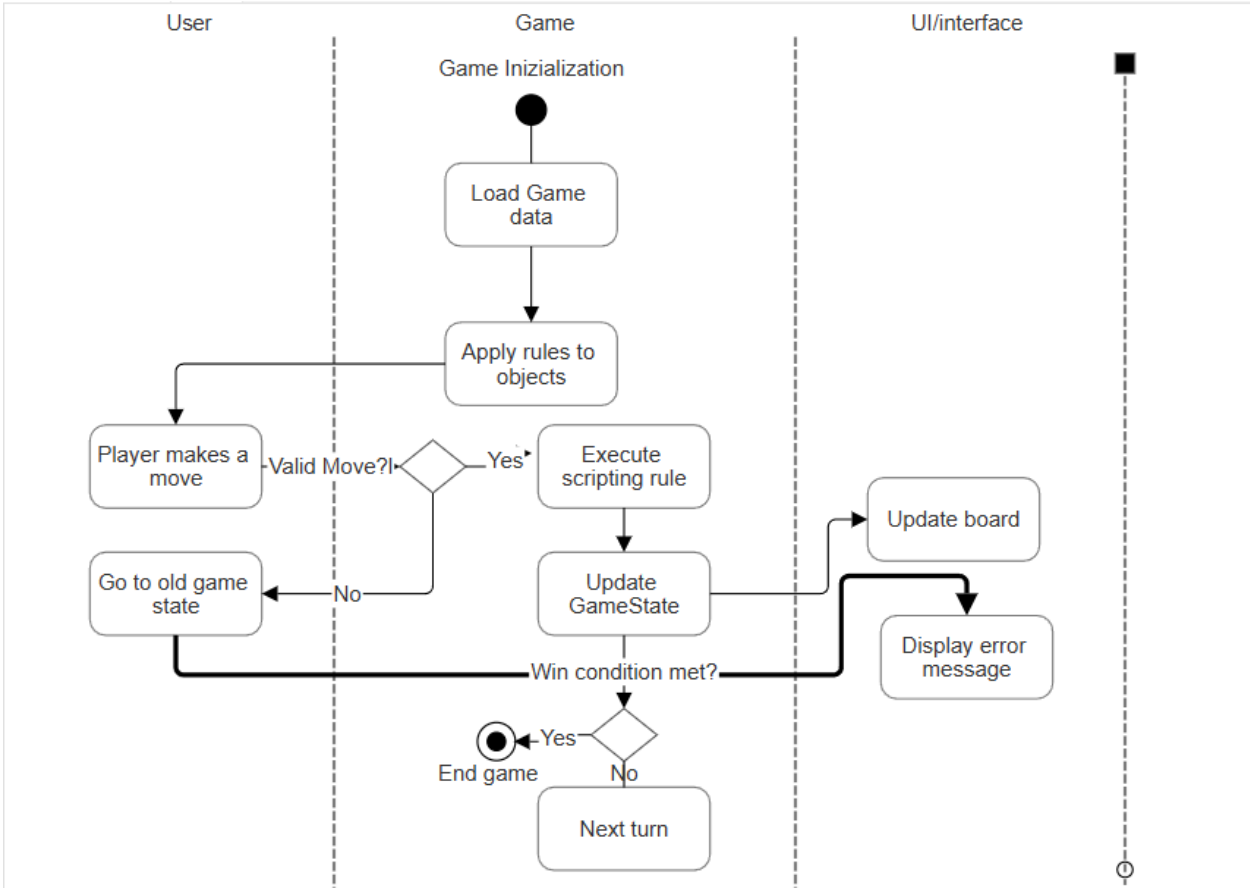
Object Diagram

Note: This diagram focuses on the backend, it does not include the HTML structure

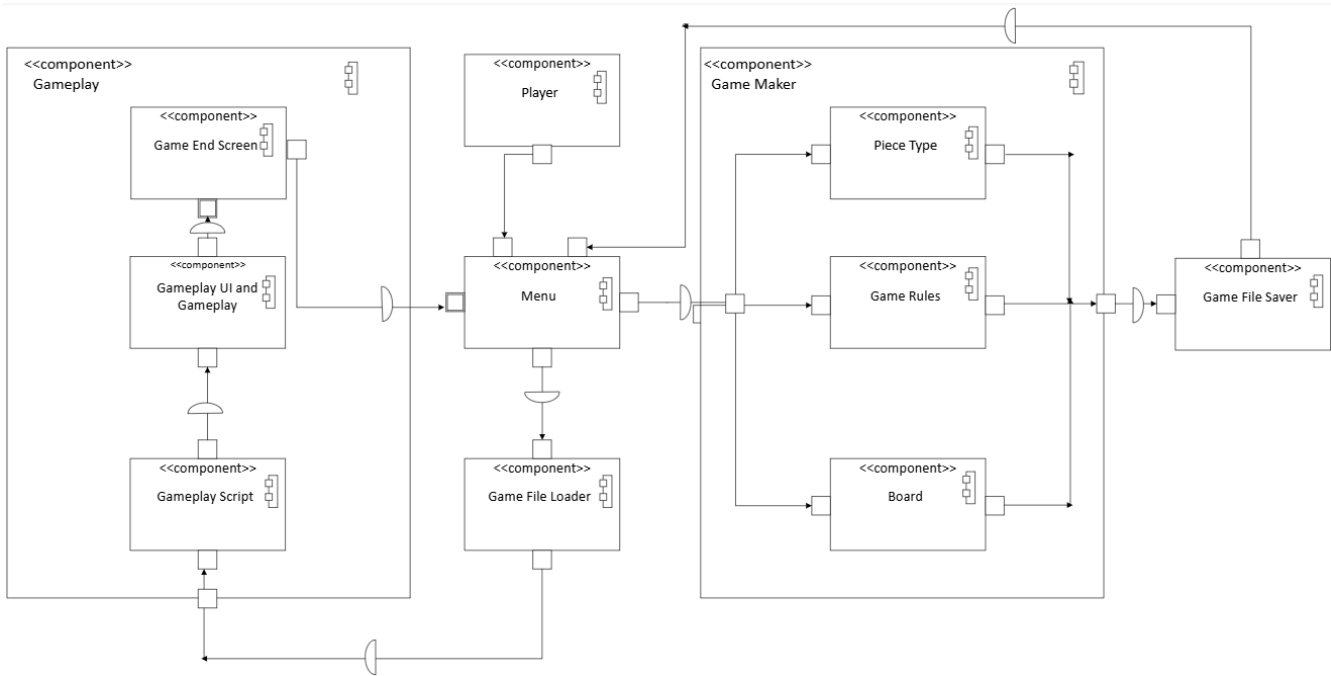


Activity Diagram

For gameplay

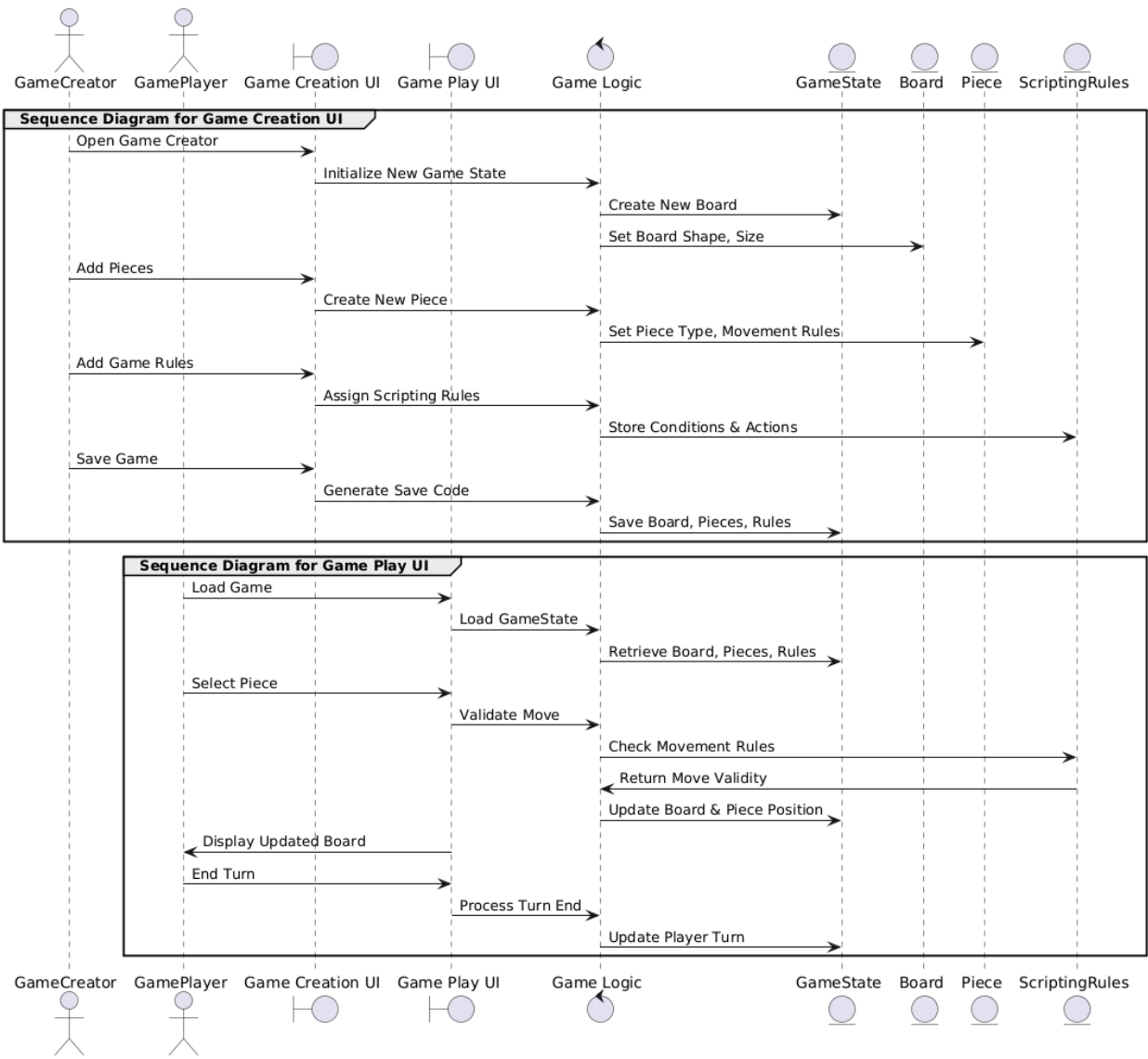


Component Diagram



Sequence Diagram

For both game creation and gameplay.



Scripting Rule Design

Entries in [brackets] can be plugged into.

Entries in {braces} can be referenced by the script (can read value, relationships, methods, etc)

Triggers

- When this piece moves in {direction}
- When this piece lands on {tile}
- When this piece is removed
- When this tile is landed on by {piece}
- When turn begins on turn {turn number} {player number}
- When turn ends on turn {turn number} {player number}
- When this piece is added to the board
- When this tile has {piece} added onto it

Actions

- Remove this piece
- Move this piece in [direction]
- Change owner of this piece to [player]
- Move this piece to inventory of [player]
- Add type [type] to piece/tile
- Remove type [type] from piece/tile
- Declare variable of this object
- Set/increase [variable] of player/piece/tile by [number] (score, for example)
- Add [piece] to this tile
- Change turn phase to [phase] (ending the turn is one possible phase to move to)
- End game in victory for [player] (set [player] to some non-player value for a draw)
- Declare game state valid
- Revert to previous valid game state

Value Reporters

- Variable [number] of this piece/tile
- X coordinate of this piece/tile
- Y coordinate of this piece/tile
- Types of this piece/tile
- Turn number
- Turn phase
- Player whose turn it is
- Value of a variable (this might be score of a player, a counter for some piece, etc.)
- Board width and height

Object Reporters

- Tile at [x coordinate] [y coordinate]
- Pieces on this tile
- Tile this piece is on

Control

- If-then-else
- Comparison operators (=, !=, >, <, >=, <=)
- Boolean operators (&&, ||, !, maybe XOR)
- Addition, subtraction, multiplication, division, modulo, etc.
- Array operators (length, entry n of, index of, etc.)

Example Milestone Games

- **Chess/Checkers** (core MVP)
- **Tic-Tac-Toe**
- **Connect Four**
- **Battleship** (if time allows)
- **Qwirkle** (if time allows)

Testing Plan	
Type	Description
Unit Testing	Start by testing just individual triggers, interactions, etc.
Turn Testing	Begin Testing Full Turns and Turn Cycles (even if the game is not in a state such that it can be "won," we can still start making sure the turn cycle works correctly.
Game Testing	Begin to test our milestone games. Attempt to break the rules and play in potentially unexpected ways to find bugs and oversights.
User Testing	We may potentially invite players who are not part of the development team to try creating or playing games to make sure the UI is semi-intuitive.

●