| Sellswords & Spellcrafts  Project Requirements  Luke Schnetlage  Mason Evans  Nicolas Towery |
| --- |

# Abstract

We intend Sellswords & Spellcrafts to be a web-based 1v1 deck-building game. Users will connect to a central web server in which they can be matched with other users. We want to allow users to be able to take their turn, while the other user takes their turn once the first user has completed their turn. We hope to achieve this so as to allow each user to see their opponent’s moves or actions in real time. The central gameplay loop will follow 2 players drawing cards from their own randomized decks until they have a number of cards equal to the starting hand size. A random player will be given the ability to play first, after the first turn, players will alternate turns until one player’s life total has been reduced to zero or have zero cards in their library. With the gameplay mechanics in mind, we will implement a system which allows users to better understand the rules of the game and provide helpful tips as needed. Our hope is to provide users with a unique gaming experience and to support multiple diverse play-styles including aggressive, controlling, “midrange”, and other niche options.

# Tools & Technologies

Desktop / Windows 10 - For simplicity sake, All team members own a desktop computer running Windows 10. Given the time and scope of the project, developing for one OS on one platform will be easier.

Firefox - All team members have experience developing and optimizing web based applications for Firefox.

Visual Studio 2019 - Provides the ideal development environment given our use of tools in the .Net ecosystem(C#, Blazor,...) .

C# - C# is a prerequisite for Blazor development as well as a language all team members are familiar with.

Blazor Server - All code will run server side. This will streamline the process, as all code will be written in one language, C#. This will also aid in simplifying stylization.

Discord - Discord provides persistent records of team discussion as well as the ability to designate specific discussions to their own channels.

Google Cloud - Google Cloud’s services are free and links directly to GitHub, streamlining the process.

# Requirements list

## Home Page

* 1. When first loading webpage, user will arrive at home page
  2. Home page should contain the following
     1. A header containing an image of the logo
     2. Subheader text containing team member names
     3. A brief description of how to navigate the website that states
        1. What you will find on rules page
        2. What you will find on game page
     4. A description of why the project was created
     5. A brief description of what the game is
     6. Navigation menu

## Navigation Menu

* 1. Navigation menu should be present on all web pages including
     1. Home
     2. Game
     3. Rules
  2. Navigation menu should include
     1. A link labeled “Home”
        1. On click, will redirect to home
     2. A link labeled “Game”
        1. On click, will redirect to game
     3. A button labeled “Rules”
        1. On click, will redirect to home

## Rules page

* 1. Rules page should contain the following
     1. A header containing an image of the logo
     2. A text block containing the rules and how to play the game
     3. Navigation menu

## Game Page

* 1. The game page must contain the following
     1. A header containing an image of the logo
     2. A window that can be in three of the following states:
        1. Pre game is the starting window
        2. Current game follows the pre-game window
        3. Post game follows the current game window
           1. Can loop back to pre-game
     3. Navigation menu
  2. Alerts user asking for confirmation when they attempt to do the following
     1. Close browser
     2. Close tab
     3. Leave web page

## Pre game window

* 1. Default state of game window
  2. A label for a display name
  3. A textbox for the display name label
  4. Text telling the user how many characters the display name can be
  5. A submit button for display name
     1. If user has not already submitted a display name in current web page session, userID will be tied to player
     2. If user has already submitted a display name in current session, display error message
     3. If user has refreshed the webpage, wipe display name
     4. If user enters invalid display name, display error message
  6. Once user submits display name, users may select one of the following options
     1. A button titled “Join Session”
        1. Once clicked, screen will change to the following:
           1. A label for list titled “Open Sessions”
           2. A list of joinable sessions

List items include the following

A label for the player name hosting the session

A “Join” button

On click, user will attempt to join session

If successful, will enter “current game” state

If unsuccessful, display error message

A button to refresh “Open Sessions” list

If successful, display refreshed list

If unsuccessful, display error message

A button to create a session

On click, enter 5.6.2

* + 1. A button titled “Create session”
       1. Once clicked, screen will change to the following
          1. A header titled “Current Session”
          2. Display user’s display name back to them
          3. Text explaining the game is waiting for another player
          4. A “back” button

On click, bring user back to “Open Sessions” screen (5.6.1)

* + - * 1. Upon another player entering the session.

If successful, enter “current game” state

If unsuccessful, display error message

## Active game window

* 1. A cog icon will constantly be presented to the user
     1. Once clicked, the cog will open an options menu with the following options
        1. Surrender
           1. On click, the user will be presented with the following options

A button labeled “Yes”

On click, automatically lose the game and return to menu

A button labeled “No”

On click, return to options menu

* + - 1. Back to game
         1. On click, the options menu will close
  1. Upon entering “current game” state, both users will be presented with the option to select one of three possible decks.
     1. “Fire deck”
     2. “Water deck”
     3. “Wind deck”
  2. After deck selection, users will be presented with a “confirm” button
  3. After confirmation, users will be presented with all the cards in their “hand”
  4. User can click on a card in their hand
     1. On click, this will select a card
        1. User can click on an empty zone that shares a type with the selected card
           1. On click, the selected card will leave the hand and enter the selected zone
        2. User can deselect card by clicking away from card
  5. Once on the field, your minions cards can be clicked to select them
     1. Once selected, minions will display a context menu with the following options:
        1. “Attack”
           1. On click, the user will be prompted to pay the relevant energy cost to attack

The user will then be given the option to select which terrain they wish to expend

Once valid terrain has been selected to pay costs, the user will be presented with a button labeled “confirm attack”

* + - * 1. Once costs are paid, the user will be prompted to select which zone within its proximity to attack

Once a zone is attacked, the attack will resolve

An attack on a creature that does not kill said creature will indicate how much damage was dealt

An attack that kills the creature occupying the zone or an attack on an empty zone will result in the attacking creature occupying the zone

An attack on an opponent will indicate how much damage was dealt

* + - * 1. Alternatively, the user may press a button labeled cancel

On click, the attack will be canceled and the user will return to the default game state

* + - 1. minions with an activated effect will also display “Active effect”
         1. On click, the user will be prompted to pay the relevant energy cost to activate the designated ability

The user will then be given the option to select which terrain they wish to expend

Once valid terrain has been selected to pay costs, the user will be presented with a button labeled “confirm attack”

Alternatively, the user may press a button labeled cancel

On click, the activation will be canceled and the user will return to the default game state

* 1. Once on the field, your invocation cards can be clicked to select them
     1. Once selected, invocations will display a context menu with the following option:
        1. “Pay Ritual Cost”
           1. On click, the user will be prompted to expend terrain to pay for its indicated cost as outlined in 6.6.1.1.1.
           2. Once costs are paid, the invocation will resolve

Resolving invocations that require targets will prompt the player to select targets before resolution

* 1. An opponent’s hand will be represented by a number of card backs equal to the number of cards in an opponent’s hand
  2. Simultaneously as the user can see their hand, the user will be presented with the default game board containing the following zones:
     1. 2 friendly invocation zones
        1. Players will be allowed to place their invocations to their empty zones to give them their relative time needed to resolve
     2. 2 enemy invocation zones
     3. 3 friendly minion zones
        1. Players will be allowed to summon minions to their empty zones to give them a position to attack adjacent contested zones
     4. 3 enemy minion zones
     5. 1 friendly terrain zone
        1. Players will be allowed to play one terrain per turn to their turn to their terrain zone
     6. 1 enemy terrain zone
     7. 3 initially neutral contested zones
        1. Players will be able to attack contested zones with adjacent minions
  3. An “end turn” button will be superimposed over the board,
     1. When pressed, the user will end their turn
        1. After a user ends their turn, their opponents turn will begin
        2. During an opponent's turn, the “end turn” button will be grayed out
  4. The player’s life total will be superimposed over the board in green text
  5. The enemy life total will be superimposed over the board in red text
  6. A player’s deck will be represented by a card back on the player’s side of the field
     1. A number will be superimposed over the player’s deck to denote how many cards remain in their deck
  7. An opponent’s deck will be represented by a card back on their side of the field
     1. A number will be superimposed over the opponent’s deck to denote how many cards remain in their deck

## Post game window

* 1. Upon entering “post game” state, user will see one of the following based upon win condition
     1. Relevant message based upon win condition
     2. Button titled “rematch”
        1. Upon clicking, player is “readied up” for rematch
     3. Counter that updates upon players’ choice to rematch
     4. Button titled, “Main Menu”
        1. Once clicked, brings user to Pre-Game window state

## Error Message

* 1. Must be presented when exception occurs
  2. Text relevant to the exception handled
  3. “Close” button
     1. On click, exit error message window
  4. “Home Page” button
     1. On click, the user will be returned to the home page

# Project Timeline

| **Dates:** | **Workload:** |
| --- | --- |
| Week of January 15 - January 21 | **Group -** Designate group members, determine each group member’s experience, agree upon project idea, and prepare a rough abstract. |
| Week of January 22 - January 28 | **Group -** Come together to play-test demos of various versions of potential gameplay, finalize written project proposal, and begin preparing project requirements document and supporting presentation. |
| Week of January 29 - February 4 | **Group -** Discuss what possible tools to use and determine which works best. Finish preparing presentation for project requirements. |
| Week of February 5 - February 11 | **Luke -** Begin creation of class files, develop ERD.  **Mason** **-** Begin design of cards and game-based visuals. Set up Git repository and create easily accessible link between repository and server.  **Nicolas -** Create several potential web designs and confer with group to determine the best option. |
| Week of February 12 - February 18 | **Luke -** Determine and develop the appropriate cycles for core gameplay.  **Mason** **-** Finalize all game-based visual designs. Begin developing an instanceID.  **Nicolas -** Create webpages and begin stylizing each webpage |
| Week of February 19 - February 25 | **Luke -** Develop the system to declare the relationship between the game board and all cards.  **Mason** **-** Migrate all previously created designs to the appropriate format to allow for proper implementation.  **Nicolas -** Finalize web page layouts and designs. |
| Week of February 26 - March 4 | **Luke -** Create interactions among each individual card and related events.  **Mason** **-** Implement art into actual code.  **Nicolas -** Begin writing code of previously completed designs and ideas for frontend. |
| Week of March 5 - March 11 | **Luke -** Test code so all team code can be brought together.  **Mason** **-** Update server-side operations; First routine code test.  **Nicolas -** Test code so all team code can be brought together. |
| Week of March 12 - March 18 | **Luke -** Create and test helper files.  **Mason** **-** Implement instance ID or “Nickname”.  **Nicolas -** Final draft of all web pages completed. Shift focus to the game window. |
| Week of March 18 - March 25 | **Luke -** Develop “Pre-Game” and “Post-Game” menus.  **Mason -** Move front-end and back-end code to server.  **Nicolas -** Create the “Pre-Game” and “Post-Game” scripting. |
| Week of March 26 - April 1 | **Group -** Successfully test and complete a test run or “Alpha” version of the game. |
| Week of April 2 - April 8 | **Luke -** Design additional game-based features to add to UI  **Mason** **-** Ensure user information is properly stored within the server.  **Nicolas -** Create and merge audio files and game-specific animations. |
| Week of April 9 - April 15 | **Luke -** Test interactions and board-states.  **Mason** **-** Test server-side efficiency and errors.  **Nicolas -** Final draft of “Game” page is complete. |
| Week of April 16 - April 22 | **Group -** Run test version during “Poster Presentation”. |
| Week of April 23 - April 29 | **Group -** Any remaining issues, visual flaws, or additional clean-up work completed. |
| Week of April 30 - May 6 | **Group -** Present finished product. Final report and supporting documentation completed. |