

Culminating Project Worklog

Anthony Chen

June 6, 2022

- Created UML diagram for project
- Started research on code to use for functions of game
- Possible code for chess board:

```
1 public class Board {
2     Spot[][] boxes;
3
4     public Board()
5     {
6         this.resetBoard();
7     }
8
9     public Spot getBox(int x, int y)
10    {
11
12        if (x < 0 || x > 7 || y < 0 || y > 7) {
13            throw new Exception("Index out of bound");
14        }
15    }
```

June 7, 2022

- Did research on various online chess programs in Java and APIs
- Looked into ways of reading PGN files and translating moves onto a board

June 8, 2022

- Created file for home page
- Realized that there would most likely not be enough time to create both the visual and backend components of a chess training program
- Decided to focus on backend development
- Researched on possible alternatives to simply downloading PGN files and reading them

June 9, 2022

- Began to appreciate the difficulty of designing a fully functional chess board + pieces in Java
- Decided to search for alternative culminating ideas

June 10, 2022

- Changed plan for culminating project to iOS app

- Self-enhancement app with features that allow users to overcome bad habits or develop new skills
- Switched from jGrasp to XCode to use SwiftUI
- Learned how to connect structures from various classes to one another
- Created MilestonesView, AddMilestoneView, MenuContent

June 11, 2022

- No progress

June 12, 2022

- Created SteppingStonesView and Assets file for storage of potential assets
- Sketched home page and other views on paper, decided on overall mountain theme for app
- Learned a bit more about adding background images to app
- Created basic functionality for MilestonesView and MenuContent

```
struct StrokeText: View {
    let text: String
    let width: CGFloat
    let color: Color

    var body: some View {
        ZStack{
            ZStack{
                Text(text).offset(x: width, y: width)
                Text(text).offset(x: -width, y: -width)
                Text(text).offset(x: -width, y: width)
                Text(text).offset(x: width, y: -width)
            }
            .foregroundColor(color)
        }
        Text(text)
    }
}
```

June 13, 2022

- Completed architecture for all basic views
- Previewed program for the first time, encountered various issues with loading MilestonesView
- Added text formatting and background colour changes
- Had significant difficulty implementing tips and edit buttons
 - Formatting distance was especially hard, and layering every element properly
- Inserted scrolling mechanisms into both the MilestonesView and SteppingStonesView codes

June 14, 2022

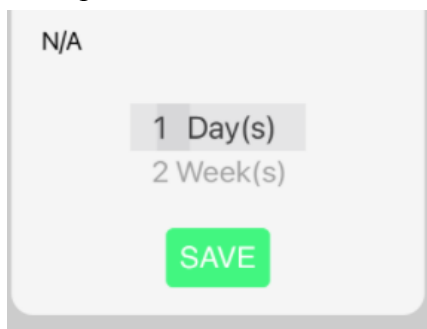
- Cleaned up formatting for homepage and other main views
- Adjusted certain colours manually using HEX values
- Did some research and fixed issue of overlapping buttons on the MilestonesView page

June 15, 2022

- No progress

June 16, 2022

- Started working on AddMilestoneView
- Learned how to connect buttons in various views
- Designed user friendly interface to select various lengths of time to set for each goal
- Learned how to implement text box for user to input name of milestone
- Created text box for this purpose, initially had difficulty with positioning it above two date options below



N/A

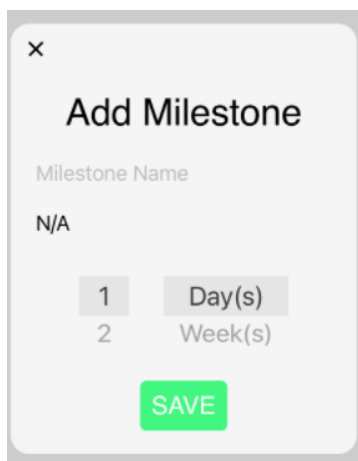
1 Day(s)

2 Week(s)

SAVE

June 17, 2022

- Finished user interface for adding milestone



×

Add Milestone

Milestone Name

N/A

1 Day(s)

2 Week(s)

SAVE

- Completed updated project proposal

June 18, 2022

- Submitted project proposal (wrong version)

- Researched methods of saving variable values in SwiftUI
- Discovered UserDefaults feature, saved for coding on a later date

June 19, 2022

- Integrated AddMilestoneView into MilestonesView
- Enabled user to add their own milestone through button in MilestonesView
- Fixed error that caused offset value for AddMilestoneView to register as null when shown

June 20, 2022

- Started working on allowing the user to actually add milestones
- Updated macOS to Monterey in order to use updated swift documentation compiler (DocC)
- Completed commenting and documentation for MilestonesView, created .doccarchive file to be submitted at a later date
- Submitted updated, final version of project proposal

June 21, 2022

- Completed slideshow and preparation for presentation
- Added function that allows user to add their own milestones
- Researched how to add storage variables to code
- Figured out how to create new remote repository from XCode
- Ran into issues with old credentials, had to delete old information, fetch Github access code, and re-login in order to create remote
- Accidentally deleted entire culminating project by replacing it with a file with the same name
- Salvaged all files using iCloud retrieval tool
- Uploaded .doccarchive file, realized that it wasn't accessible without the Xcode program installed or a dedicated website
- Took screenshots of the most important parts of the DocC file, added to supplementary document
- Wrote basic reflection on use of modular programming and DSDM approach and submitted supplementary document
- Added UserDefaults variables to document, ran into errors that prevented the program from running and temporarily postponed this initiative

June 22, 2022

- Attempted to upload .doccarchive file to Github but was beyond the 100 file limit
- Uploaded file to drive and added link to supplementary document, resubmitted
- Added background images to SteppingStonesView and MilestonesView
- Edited SteppingStonesView to display stepping stones more accurately

- Presented Milestones app to class, discussed its various features and next steps

June 23, 2022

- Completed reflection, submitted both reflection and this work log