

# Team Practice Scheduler

**Created by: Mosab Elagha**

**Version 1.0**

A scheduling program to assign team players practice times that work best with their preferences and availability

## Directions for use

1. In 'RELEASE' download 'Team Scheduler-1.0.zip' and unzip the file
2. Use the **EXCEL** document (not CSV) to fill in
  - Player names
    - As many as you need
  - Time slots (up to 9)
    - If a time slot is being used, fill in data for all cells
    - If a time slot is not being used, leave the *data* for it blank
      - The time slot cell itself can have writing in it
  - Player preferences for time slots
    - 3 = Highly preferable
    - 2 = Preferable
    - 1 = Less preferable
    - 0 = Unavailable
3. Export/Save As to a CSV file
  - When modifying the player info, do it in the excel document then save as CSV again
4. Open Scheduler.jar
5. Click the button 'Go' when you're ready
  - If you saved the Player Info.csv file as a different name, write '{filename}.csv' in the input box
    - Without quotes and where {filename} is the name of your file without brackets
6. In the 'output' folder there are multiple results
  - Open 'Recommended.csv' to see the best results
  - The other files are different schedules that you can choose from, but are not necessarily the most optimal

**If you have any suggestions, please let me know!**

**Note:** *I created this to be used with the Excel file in order to make it easy to paste data into the program. I envision this program being used after a survey sent out to team players (a google doc) asks what times work best for them. The data would be stored in spreadsheet anyways, and so copy and pasting the data would make the whole process go quickly, without the person making the schedule needing to manually input data.*

---

# Building (for developers)

## Installing Gradle

This is a Gradle project. You will need to have Gradle installed.

### Mac

You can use Homebrew to easily install Gradle (<http://brew.sh/>)

```
brew install gradle
```

### Windows/Linux

You can find instructions here:

<http://www.gradle.org/docs/current/userguide/installation.html>

## Building an Eclipse project

To build an Eclipse project, you can run the following

```
gradle eclipse
```

After this, import the project to Eclipse

## Exporing the completed .jar

### Recommended

I recommend using the custom command I created for this project

```
gradle release
```

This will create a folder 'RELEASE' in the same directory as the build.gradle file and that folder will have all the files that are needed.

Additionally, a ZIP file with all the files ready for release is generated. (You'll need to run the command again to do this. Small bug, not sure how to fix)

### Other option

You can also run

```
gradle build
```

in order to run the project. This will create the .jar file in 'build/libs' but you will need to manually move the other required files.

# Todo

- Fix corner cases
- Make variables more dynamic (number of players per slot, level names, number of levels, etc.)
- Make build.gradle better
- [Low Priority] Have this all be self contained (no excel/source file)