# ENG6 Fall 2020 Final Project

Up to now in ENG6, we have focused on teaching you the “how” of programming. In the team project you will be taking your own ideas and bringing them to fruition through your knowledge of computer programming. Software development is rarely a one-person effort, so the project will be team-based. You will be randomly placed into a 3-4 person team with members from any section.

*Important: We discourage the use of other Matlab code written by someone else, however, if you use it you must reference the code creator. Furthermore, do not use projects done in previous quarters, as we have a record of these.*

## Project: Dice Game

**Project Description:** Implement a MATLAB computer program of dice game of your team’s choice. For example, possible candidates are Farkle. Or Left Center Right. An example of a computer dice game implementation is Yahtzee. (it is clear what is expected of the players, however with respect to the visual and auditory aspects, it is not a good example). The dice game must involve one or two players. A player should be able to make strategic decisions to win the game. The program should roll the dice upon request, make apply the rules and score the game. Each team makes decisions about what programming elements will engage the users.

**Links to external resources:**

- Controlling Random Number Generation.
- Play Audio

ThingSpeak Documentation

**Collaboration Policy:** Once the teams are formed you are only allowed to talk and collaborate with members within your team. Team members are expected to equally participate, and collaboratively work towards the completion of the project. Other than contacting the teaching

assistants for clarification, you may not seek the assistance of other persons to complete your team's project. (Of course, general discussions about how to implement a GUI, OOP, and other programming constructs could be discussed with other students, but your team must be totally responsible for the implementation of code used in your project).

**Grading Criteria:** The projects are open-ended. As long as your program can perform the assigned tasks, there will be no correct or incorrect approaches. Certainly there will be more acceptable and attractive solutions. The final project will be graded in 7 parts:

1. ***Project proposal***: Each team submits a 1-3 page project proposal via Canvas describing the project they have selected, a general description of how they will implement the main components of their project. Essentially, the scope of the project should be both challenging enough to merit full credit and doable within the timeline. An Appendix should contain a breakdown of programming tasks, and who will be responsible for what, along with a timeline that will meet the submission deadline (we suggest you make use of a Gantt chart). The expectation is that each team member must take responsibility for a specific aspect of the project and grading for each member will be adjusted according to how the project tasks were delegated and who was responsible for what aspects of the project. The more specific you can be in defining the programming tasks, what functions should exist, and what each function should accomplish, the better.

2. ***Core***: Complete the basic project as outlined in the project description. This is the basic functionality of your game and will be graded earlier as a milestone.

    - **Features of the GUI:**
        - Roll dice button
        - Display scores edit fields
        - Anything else you need or want to add depending on the game you have chosen.

At this point, the game should be playable and run without errors.

3. ***Reach/Special Features***:

    - **Animations**
        - Rolling dice
        - Earning points
        - Others
    - **Sounds**
        - Rolling dice
        - Earning points
        - Others

- **Originality**
    - Is your application a well-known game or does it make interesting modifications to existing rules or have completely unique rules? You can also get credit for interesting gameplay, even if the rules are simple. For example, instead of being turn-based, the game can require players to react in real-time.
- **Artistic Design**
    - Your game should be aesthetically pleasing, e.g. GUI, images, sounds, with animations being especially encouraged.

4. ***Remote Gameplay*****:** The game should allow remote gameplay between two or more players in different locations.

- We will be using ThingSpeak to enable different instances of your application to communicate over the internet.
- You will be learning ThingSpeak throughout the last weeks of the quarter, so this aspect of the project will be an appropriate proportion of your grade.
- The remote functionality should not make the game cumbersome to play. For example, the GUI should update automatically when receiving new data without additional user input.
- If you have an idea for an alternative way to implement remote functionality other than ThingSpeak, bring it up with us and it will be considered.

5. ***Code Requirements*****:** Your code should be well-organized with properly named variables. The code should be well-documented, especially the part for remote functionality. Also consider whether certain data structures are better for implementing certain features: classes, structs, tables etc. The game should run without errors.

6. ***Youtube Video Requirements*****:** Youtube has several examples of ENG6 videos (search ENG6). The format of the video is entirely up to your team as long as the following criteria are met:

- The maximum length of the video is 10 minutes
- Each team member must be seen in the video to present their work and contributions
- There should be a clear and easy-to-follow demonstration that shows the correct functionality of your program (show your program actually working in the video – not screen shots of before and after.) This is especially important for the remote functionality, and your team members should be shown playing remotely with each other.
- Be honest if a certain feature is not working 100%. More leeway is given for honesty over a broken feature than for a feature that is shown to be working when it actually doesn't when the application is run.

7. ***Team Evaluations***: Each member must provide a brief personal summary of their involvement and contributions. Each team member is required to submit evaluations of their teammates' contribution. For example, if your team has members A, B, C, your evaluation can be similar to the following for a single member.

Team Member A: was in charge of writing the code to execute the equalizer filters. For the Reach features, A was in charge of adding 2 different analysis plots that could show power spectral density plot and frequency content of audio file. Team Members B, C agree that A performed these tasks for the project.

**Project Deadlines**:

Deadline #1: Friday, November 6, 23:59 PM: **Submit the Project Proposal:** A team member must submit your team name to canvas with the project proposal. Only one team member should do this!

The submission **must** contain an image of the design view with the main components of your game. The image must show the buttons, axes, images, edit fields, etc.

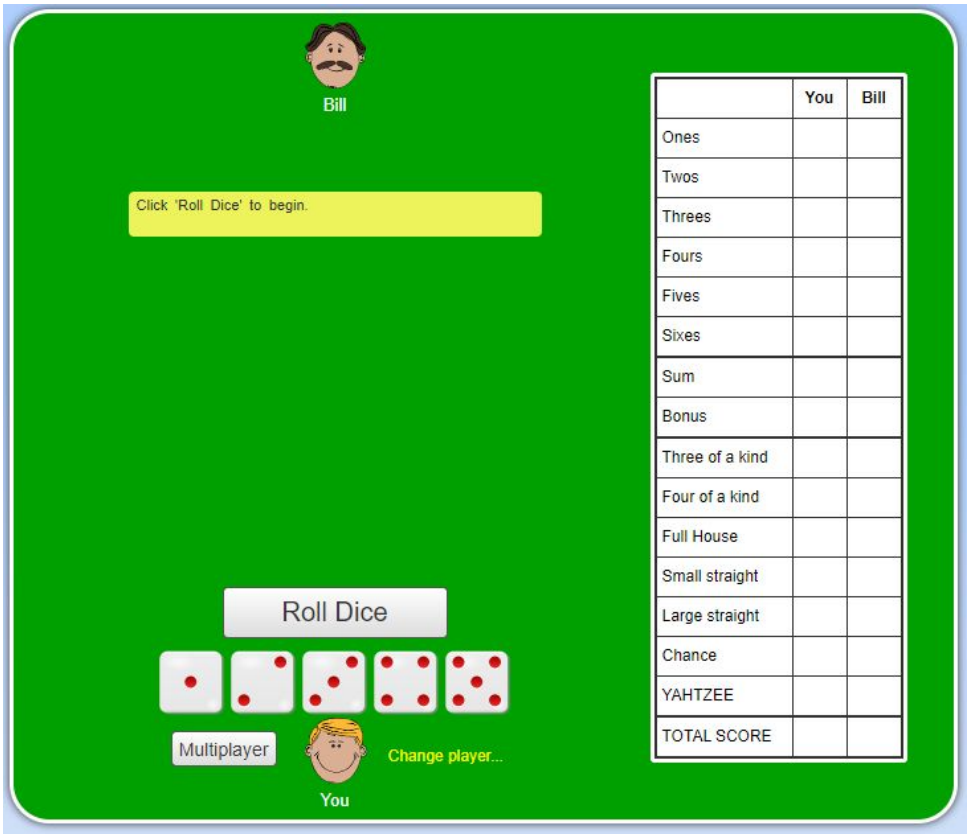


**Figure.** Example of a design view: Yahtzee dice game

Deadline #2:  Week Nov 30-Dec 4: **Present the core component to your assigned project TA in a scheduled consultation**. A single team member should email all relevant code files to the TA prior to the consultation. The zip file will have the mlapp. and all .m files, all .img files and any other files that are needed for the game to run.

The submitted .mlapp GUI must at least allow the game to be played by one or two players using the same computer.

Deadline #3:  Friday, December 11, 23:59 PM: **Submit the Final Project.** The final project must include the **Reach/Special features.** A single member of each team will submit all relevant coding files, a link to Youtube video and team evaluation materials. That person should also submit a zip file of all the code, the zip file will have all .mlapp files, all .img files and any other files that are needed for the game to run. In addition, the submission should contain a PDF of the team evaluation document. The link of the Youtube should be accessible to all those who use the link.

**Team Issues**

Issues involving team members should be resolved early in the project timeline. These can include: team members not communicating/responding, team members not attending meetings, team members not doing their assigned portion of work, etc. These cases need to be brought up with your TA or instructor early in the project timeline, so that the necessary adjustments and interventions can be made to address them. **Issues with your team members do not provide a valid excuse to turn in inadequate work, as these issues should be brought up and resolved early in the project timeline.**