

## Warmup Exercise - Team and Technology DeRisk: Group 16

Team Number: 16

Team Name: Sweet 16

Github Repo Link: <https://github.com/cse110-sp25-group16/warmup-exercise>

Youtube Link: <https://youtu.be/xeA5QYWfpOM>

### SWOT Analysis:

- Strength:
  - We feel that there was a lot of good communication throughout our group. We were able to delegate the work between the team, splitting into groups by steps. This allowed for clear expectations per member to be set. Within the groups, there was a lot of guidance in explaining parts of the code and set-up to one another, which also helps equalize the knowledge base between members who may be coming in with more or less web development experience.
  - Clean code management! Our team did a good job of making sure they branched from main and did any edits on their own local branch. This ensured that the main branch had clean working code, and any experimentation did not cause conflicts. We were also able to work more collaboratively by visiting each other's remote branches, aiding in teamwork within and between groups. We also made sure to always write documentation for our functions and code, and sort it into files accordingly to keep everything organized.
  - Tools: Our team did not have a hard time setting up VSCode to interface with GitHub, and we did not run into issues with the HTML incorrectly displaying.
- Weakness:
  - Setting aside sufficient time for tasks. While groups 1 and 2 were able to finish their code within the set deadlines, given that the final deadline of the project was quite soon, group 3 had a very short timeframe to implement Step 3. Anticipating that this step would take more time and testing may have led to more time being reserved for it.
  - Finding times within a large group to schedule meetings. With a group of 11, it is difficult to find a significant amount of time that teams can meet up and work together. It is easier to find times for groups of 3 or 4 to meet up, but this can sometimes result in the larger team being left out of discussions on important project elements.
  - Breadth of Web Development Knowledge: Since we have many developers on our team, we also have a wide and varying range of skills. We have several team members familiar with back end development and/or front end development, but they have different levels of experience with CSS, JS, and HTML. Being able to

navigate expectations between members is definitely something we can improve on in future projects.

- Opportunity:
  - In order to gauge the skill level of members in our group, we attempted to not use any assistance from AI as much as we could. This means that our development team has a greater understanding of the code and its implementation, as they are responsible for directly coding and ensuring its functionality.
  - We have a very modular code base, which allows for flexibility in functionality, making it easy to implement different games and rules.
- Threat:
  - Time: Not having enough time to dedicate to both implementing the next step while also thoroughly understanding those that come before is a large threat. In order for divide and conquer programming to truly work, we must have a good idea of what we are reasonably able to achieve with our given skills and available working time. This may mean cutting features or not achieving our most advanced goals, but it will mean that our team will function better and be able to present better code that follows software engineering principles.
  - Using AI to write code. While this will help us finish steps faster, it may lead us to write code that is not functional, or code that contains a lot of bugs as it is not familiar with all of our files, structures, databases, and libraries.
- Summary:
  - This warm up helped us learn where our team's strengths and weaknesses lie, as well as what steps we can take in future projects to improve our code. We recognized that areas such as communication and coding practices fell in our strengths, while managing expectations on differing levels of experience and knowledge was a weakness to anticipate in future sessions. We did attempt to use AI as little as possible in our code, which is a good practice for producing thorough code that the team has a consummate understanding of, and while appealing, we should continue to try and use as little AI as possible.