Praeses Technical Sample

# Task

You will create a program that plays the card game Blackjack. You may use any technology stack that you wish unless directed otherwise. This is an opportunity for you to show us the caliber of work that you produce.

# Evaluation

### Adherence to Game Rules (High)

Like most card games, Blackjack is heavily based on rules. As part of this task, you will need to identify rules for the core game and implement those in your submission. Bicycle Cards has a good [set of rules](https://bicyclecards.com/how-to-play/blackjack/). But please consider betting, splitting, and multi-player non-essential features. They may be included if you wish, but they are less important than the core of the game. The most important factors here are: “dealing” cards in the correct order, identifying win conditions, handling player actions appropriately, handling the value of aces.

### Overall Impression (Medium)

A holistic impression of the submission. This includes, but is not limited to, the other categories listed here, such as fitness of the chosen technology stack for the final deliverable.

### Program Structure and Design (Medium)

This covers how well designed the application is. Is the code broken up into multiple files, classes, functions, etc? Is each piece of the application focused and cohesive?

### Documentation and Readability (Medium)

This focuses on how easy the submission is to read and understand. Ensure that variables and functions are appropriately named. Code comments are not required, but feel free to add if needed to enhance clarity.

A short README should be included as part of your submission (a simple README.md file in the repository will suffice). It should indicate any additional features that you have implemented beyond the core game rules. Additionally, it should include instructions on how to build and run your application. This should include any information about the tech stack that needs to be installed in order to complete first run.

### UI / Presentation (Low)

### The focus here is on the user experience of the application. While a graphical interface is always a nice touch, it is not a requirement. A solid command-line experience is more impressive than a poor GUI experience.

### Additional Features (Low)

Additional consideration will be given for any features that are implemented beyond the core game. Some examples include multiplayer, betting, and splitting. These are just a few examples of features that you *could* add.

**Note: No additional features are required beyond the core game.**