Go Gitters Team Charter

1. Purpose and Objectives

Purpose: Develop a clean, responsive, and user-friendly **card-based web application** using HTML, CSS, and JavaSCript. The game will demonstrate core mechanics of Blackjack, with extensibility added through Jokers. It should exhibit proper score calculation and provide audio-visual feedback.

Week 1 (May 17, 2025):

- Implement deck generation, interactable cards, and Hands
- Add core game loop (Send/Discard, basic scoring)
- Build basic UI layout for player/dealer hands and controls

Week 2 (May 24, 2025):

- Enhance UI/UX with animations, responsiveness, and keyboard controls for accessibility.
- Integrate audio effects and score tracking display bar

Week 3 (May 31, 2025):

- Begin basic user testing and collect feedback.
- Write automated unit tests for game logic and log bugs.

Week 4 (June 5, 2025):

- Final polish: Fix critical bugs, optimize performance, update documentation (README, JSDoc, code comments)
- Deploy final build to GitHub Pages and verify cross-browser compatibility.

2. Values and Standards

- **Trust and Respect**: Encourage open communication; critique code, not people.
- **Transparency**: Track all work in GitHub issues, document decisions in PR descriptions and ADR documentation
- **Quality**: Write clean, maintainable, and unified code; ensure user experience is well-defined
- **Adaptability**: Welcome feedback, iterate based on playtesting

3. Roles, Responsibilities and Accountability

Role	Responsibilities	Assigned To
Team Leads	Organize team, manage overall project structure, mediate inter-group communication and discussion, etc.	Nathan Reed, Hanbin Tan
Merge Coordinator	Determine branch structure, branch communication, manage merges	Min Paing
Frontend / Design	Manage the User Interface portion of the game, including designing the elements and creating HTML/CSS	Variable based on demand
Backend	Manage the game logic and implementation of Javascript code in the backend.	Variable based on demand
Documentation Lead	Manage the documentation for the project, including [README](#) management, code comments, and structural documentation.	Sruti Mani and Anthony Velikov

Accountability: Update GitHub Issues in a timely manner, ensure all PRs meet requirements and are properly reviewed.

4. Communication Protocals and Meeting Cadence

- **Channels**:
  - **Slack**: Regular chat, quick questions, showcases
  - **GitHub Issues and PRs**: Task assignments and code discussions
  - **Discord/Zoom Call**: Sprint planning, showcase, and retrospectives.
- **Meetings**:
  - **Stand-Up**: Tuesday/Thursday After Class
  - **Sprint Planning**: Weekly, Tuesdays After Class
  - **Sprint Retrospective**: Weekly, Mondays at 11:00 AM

5. Conflict Resolution and Rule Enforcement

1. **Direct Talk**: Address concerns in a private message first.
2. **Team Mediation**: If unresolved, discuss at next stand-up or DM a Team Lead.
3. **Concensus**: Majority decision, documented in meeting notes. **Enforcement Steps**:
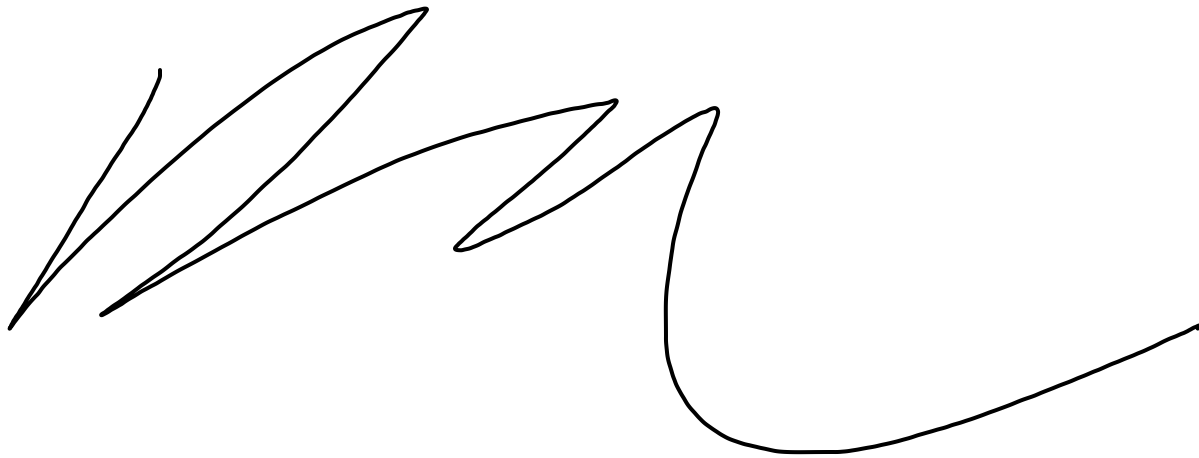
- **Reminder**: Friendly cue in Slack.

- **Check-In**: One-on-one meeting with Team Lead.

- **Escalation**: Document in retrospective and reassign tasks if necessary.

6. Working Methods and Success Metrics

- **Framework**: Agile with 1-week sprints, backlog review on Mondays.
- **Definition of Done**: Specifics defined in GitHub Issue, overall the feature code should be complete, tested, reviewed, and merged.
- **Metrics**:
    - **Velocity**: Issues closed per sprint, adjusted for severity and time consumption.
    - **Bug Rate**: Open vs. Closed Defects, Issue Encounter Rate
    - **Participation**: Attendance in stand-ups and reviews.

Review Cycle: Update this charter in retrospectives to reflect lessons learned.

Signature

Kevin Lee