**Requirements/Analysis**

* Project Description:

So, the plan is to make a digital version of Bunco, you know, that dice game where you roll and try to get specific combos to score points.

* Known Requirements:

1. We gotta make sure the game can handle rolling dice and keep track of everything.
2. It should let a bunch of us play at once.
3. Gotta recognize when someone gets a Bunco (you know, three of the same number) and give them the points.
4. And of course, it needs to be easy to play with a cool interface.

* Questions for the Client:

1. Any special rules or versions of Bunco we need to include?
2. Should we offer different Bunco types or extra game modes?
3. Got any preferences on how it should look?

* Topics We’re Unfamiliar With and References:

We’ll check out Bunco Rules (the official site) and forums for Bunco fans to make sure we get everything right.

* Users of the Software:

1. Anyone who enjoys Bunco, from casual players to serious fans.
2. People hosting game nights or parties looking for something fun to play.

**User Interaction**:

1. Players can roll dice by tapping a button or shaking their device.
2. They’ll see their rolls and points on the screen and use the interface to keep the game moving.

* Required Features:

1. Make sure the game follows Bunco rules and scores correctly.
2. Let lots of people play together at once.
3. Give options to start, exit, and maybe change settings.
4. Make it look good and easy to play.

* Other notes:
  + Maybe add themes, sounds, and tutorials to make it more fun.
  + Consider adding features for different players, like changing fonts and colors.

**Software Requirements**

The project seeks to develop a digital rendition of the Bunco game, where players aim to roll specific combinations of dice to score points. The software will feature a user interface (UI) for player engagement, including options for initiating new games, exiting, and potentially adjusting settings. The game will cater to players of varying preferences and experience levels, promising an entertaining experience for dice game enthusiasts.

Requirements:

1. Dice Mechanics:

* 1. Implement dice mechanics that accurately reflect the rules of Bunco.

- Ensure clear visualization of dice rolls within the UI.

2. Player Interaction:

* 1. Enable players to take turns rolling dice and recording their scores.
  2. Ensure smooth transition of game states based on player actions.

3. Scoring System:

* 1. Develop a scoring system that accurately calculates points based on Bunco rules.
  2. Detect when players achieve specific combinations and update scores accordingly.

4. User Interface:

* 1. Design an intuitive UI for seamless navigation.
  2. Provide accessible options for starting new games and exiting.
  3. If needed, incorporate settings to allow players to customize their gaming experience.

5. Additional Features:

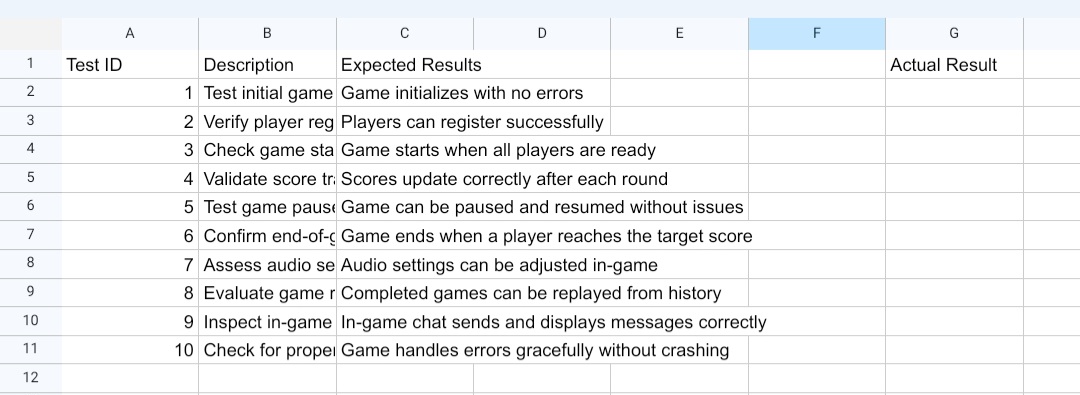
* 1. Incorporate visual cues to enhance player engagement during gameplay.
  2. Consider implementing optional sound effects to enrich the gaming atmosphere.
  3. Explore accessibility features, such as adjustable font sizes, to ensure inclusivity for all players.

**Black Box Test Plan: Bunco Game Application**

Objective: The purpose of this black box testing is to verify that the Bunco Game Application satisfies the criteria and operates as intended based just on inputs and anticipated outputs, without taking into account the specifics of the internal implementation.

Test Items:1. Start Game 2. Roll Dice 3. Score Calculation 4. End Round 5. End Game

**Test Cases**

Test Environment:To verify compatibility, the Bunco Game Application will be tested across a range of screens and resolutions.

**Journal**

1. Input for the Software: In the Bunco game application, input involves player actions such as rolling dice and recording scores. Each input corresponds to a player’s turn, which includes rolling dice and entering the resulting scores. These inputs are recorded throughout the game.
2. Output for the Software: The output of the Bunco game application includes updates to the game state after each turn, such as the current round number, scores for each player, and notifications about special occurrences like rolling a Bunco or reaching the winning score threshold.
3. Equivalence Classes for Input: Inputs for the Bunco game application can be categorized into valid inputs (correctly recording scores according to game rules), invalid inputs (attempting to enter scores out of turn or incorrectly), and special inputs (achieving a Bunco or meeting game-ending conditions).
4. Boundary Values for Input: Boundary values in the Bunco game application include entering scores for each round within the specified range (usually 1-6), adhering to the sequence of play, and correctly handling special cases such as reaching the winning score exactly.
5. Additional Cases for Testing Requirements: Testing for the Bunco game application should cover various scenarios, including verifying the calculation of scores, checking for accurate progression of rounds, ensuring proper handling of tie-breakers, and validating any additional features such as online multiplayer or customizable rule sets.