**Team #1 - Bingo**

**Phase 1 of development (the bare minimum of what we’ll create):**

Upon launch of the game, players are presented with the main screen. This screen should say the name of our game at the top, and offers the player several options: Play Bingo, View Instructions, and View Credits. “View Instructions” displays a page(s) of basic instructions on how to play the game, just in case it’s needed. “View Credits” is basically a page with our names on it.

We might have some fun with the credits. What I envision is a bunch of Bingo cards laid out all across the screen, just like you see real Bingo players doing. But these cards have letters instead of numbers (like a scrambled-up word search), and then letters get called (very quickly, not at the pace we see in the actual game), seemingly at random, but eventually spelling out our names on the cards as the letters are automatically marked. Names could span multiple cards, since the cards only accommodate five characters. This animation would take maybe five seconds and then rest as a still image until the player navigates back to the main screen.

Clicking “Play Bingo” sends the player to the shop to prepare for the first round. Upon launch, players have 20 tickets (enough to prevent going broke too early), which are used to purchase cards in between rounds of Bingo. Cards cost 1 ticket each, although this may be tweaked to something that feels better after play testing. Tickets persist from round to round and can be saved up, but any card purchases are entirely for use in the next round.

After the purchasing phase, a round of Bingo begins.

Before detailing how a round of Bingo works, the following is a description of the cards that are used. Players possess cards containing a 5x5 grid with the letters B-I-N-G-O across the top. Each column of the grid contains numbers that are randomly generated but restricted to a certain range. Column “B” can only contain numbers 1 - 15; Column “I” can only contain numbers 16 - 30; Column “N” can only contain numbers 31 - 45; Column “G” can only contain numbers 46 - 60; and Column “O” can only contain numbers 61 - 75. A number cannot appear on the same card more than once. The center space has no number and automatically counts as a marked space.

At the start of a round of Bingo, a random number of opponents (between 50 and 75) are created, each with 1 to 4 cards (to be determined randomly). The total number of available bingos is based on the number of cards in play (formula is yet to be determined).

During a round of Bingo, the player’s card(s) are displayed. The player can have up to four cards. The player can navigate from one card to another at any time, in a manner that is yet to be determined. The Bingo “caller” who runs the game possesses a range of numbers 1 - 75 that are randomly called along with the corresponding letter every 5 seconds. Once a number has been called, it cannot be called again during that round. It is the player’s responsibility to mark his/her cards accurately, attempting to keep up with the caller. When a space on a card is clicked, the space is “marked;” conversely, clicking on a marked space will “unmark” it. (It is important that the number in the space still be visible underneath the marker.) Any space on any card can be marked, regardless of whether that number has been called or not.

There will be two locations on the screen that display the current round’s history of called numbers. One will display numbers in the order in which they were called, but will have a very limited “memory” that displays only the last several numbers. The other display will be a 5x15 grid (sort of like a “master” Bingo card) that displays all of the numbers that have been called, but it does not indicate which numbers were most recently called.

A player can call “Bingo!” on a card at any time, regardless of whether a correct bingo pattern has been marked (see below for eligible patterns). A check is made to ensure that all of the numbers in the alleged Bingo have actually been called. If the Bingo is valid, the player wins a bingo and the winning card is removed from play; if the Bingo is invalid, that card is blacked out and will not accept any input for 5 seconds. If the winning card was the player’s last remaining card, the round ends prematurely; players with cards still in play can continue playing after their first call of “Bingo,” in an attempt to win multiple prize pots in a single round.

Patterns that win a Bingo: 5 spaces in a row vertically; 5 spaces in a row horizontally; 5 spaces in a row diagonally; all four corner spaces.

The player is awarded 2 tickets for each valid bingo that is successfully claimed. (Again, this value may change after play testing.) Maybe an extra ticket is awarded for the first 10 bingos of each round. The player is taken to the shop again to prepare for the next round. The flow of the game alternates between the shop and the rounds of Bingo.

As a final note, I think we should have some sort of audio in the game. Royalty-free music, a fanfare when you win, an “awww” or angry muttering if you call a false Bingo, stuff like that. A voice that calls out the numbers is another obvious audio element; we could even add an effect like heavy microphone distortion so that players can’t rely entirely on audio. It would sound like a crappy microphone in a bingo hall. Audio will require multithreading, or else code execution will halt until the audio is finished playing.

**Phase 2 of development (things to do if we have time):**

A fun thing would be to record each of us excitedly yelling “Bingo!” When the program is launched and we see the main screen, one of the recordings is picked at random and then played.

Maybe we could allow for some sort of player persistence by saving tickets to a file upon exit, which could then be read upon launch. Encrypting this file to prevent players from cheating could be an interesting exercise. If we do this, the main screen should be modified to display how many cards and how many tickets the player currently has, for immediate reference upon launch.

Think about special items or randomly generated bonus spaces that do something if that space is marked. Or maybe bonus spaces only activate if a Bingo is called using that space, introducing a risk/reward mechanic in which a player might hold off on calling a valid Bingo in the hopes of getting a different Bingo that includes a bonus space (at the risk of an opponent calling a Bingo first).

Power-up ideas:

Something that delays the caller for a bit.

“Voice-throwing” (making an opponent call Bingo prematurely).

A chart showing the history of the player’s ticket wallet at the end of each round could be an interesting way to document the ups and downs of wins and losses, helping the player make purchasing decisions for the next round.

Maybe have a small number of achievements to give something for the player to work toward. A screen where a few badges can be displayed for things like "Getting a First Bingo," etc.

Allow network play against human opponents. The choice between AI opponents or human opponents could be presented with a menu within the game, or perhaps using different executables. Multiplayer gameplay would probably be easier to manage if we restricted it to a local network rather than the full-fledged internet.