CSCI 499 SENIOR PROJECT

CHRISTOPHER BELL

April 15, 2020

In Partial Fulfillment of The Requirements for The Degree of Bachelor of Science

Introduction:

As a brief summary of the project being built, this is an application that allows one person to play the game of bingo in a competition, race type setting with an automated computer player. This app allows said player to daub his own board on an automatically generated bingo board and create new version of this game whenever they wish. Buttons included on this app allow the player to access new versions of the game, start and stop the game, progress the game in step increments, and call bingo if they think they have it on their board.

Motivation and Problem Statement:

The game of bingo is very basic and that makes it fun for all ages and mindsets. However, the ideal game of bingo would be played with multiple people and sometimes being in a place with people can be hard, whether it's a stress disorder that keeps you from being in crowded spaces or schedule conflicts that keep friends from meeting. I know in my situation, growing up as the only child in the family, I would constantly find myself playing games built for one person since I didn't have any brothers or sisters to play with.

Because of this problem, I want to propose an application that lets you play bingo with yourself and not have to worry about finding other people to play with or an internet connection. This would let people with social anxieties, schedule conflicts, as well as any other problem play the fun game of bingo by themselves and not with other people.

Research/Background:

As I was creating this game, I looked to other preexisting Bingo games as an inspiration. What I noticed was that other similar games that already existed were either a pay to play, ad filled, or horribly complex program that made it hard to use for the people like who I used to be: an only child with not much money to spend on video games. I wanted this to be an easy-to-use, free game that could be easily accessed and played on a whim by a single player.

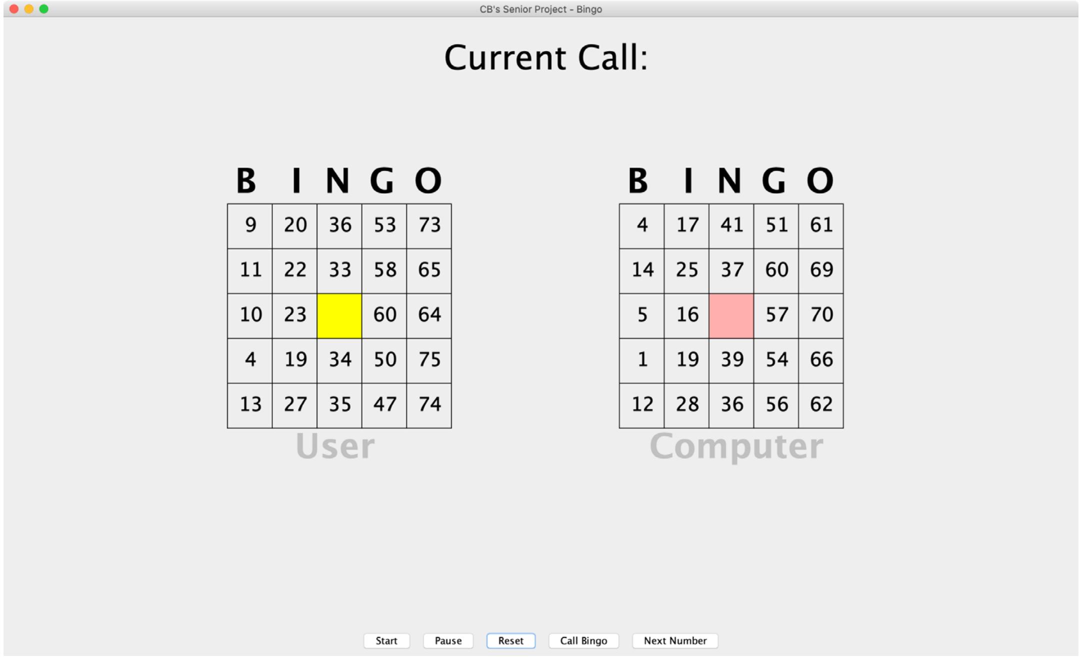
In order to create this application, I used a program called NetBeans, which is an easy-to-use IDE that lets you start, build, and export an application in many languages. I used Java as it was the best Object-Oriented language for me to use in the process. With this, I had everything I needed to create my Bingo application as every part of it was created with Java.

Any software/equipment needed:

As I previously mentioned, I wanted this application to be easily accessed and played by anyone who grew up like me, and I didn't have any fancy computer or game system, so I knew I had to pack this in a way that let you download and run no matter what you have. To make this work, I ensured that all the details packed inside of the game didn't need any extra graphic requirements, RAM, or CPU power. The final product needs a bare minimum of:

- A working and up-to-date Java environment
- Laptop or Desktop Computer

Screenshots and Explanation:



Current Call:

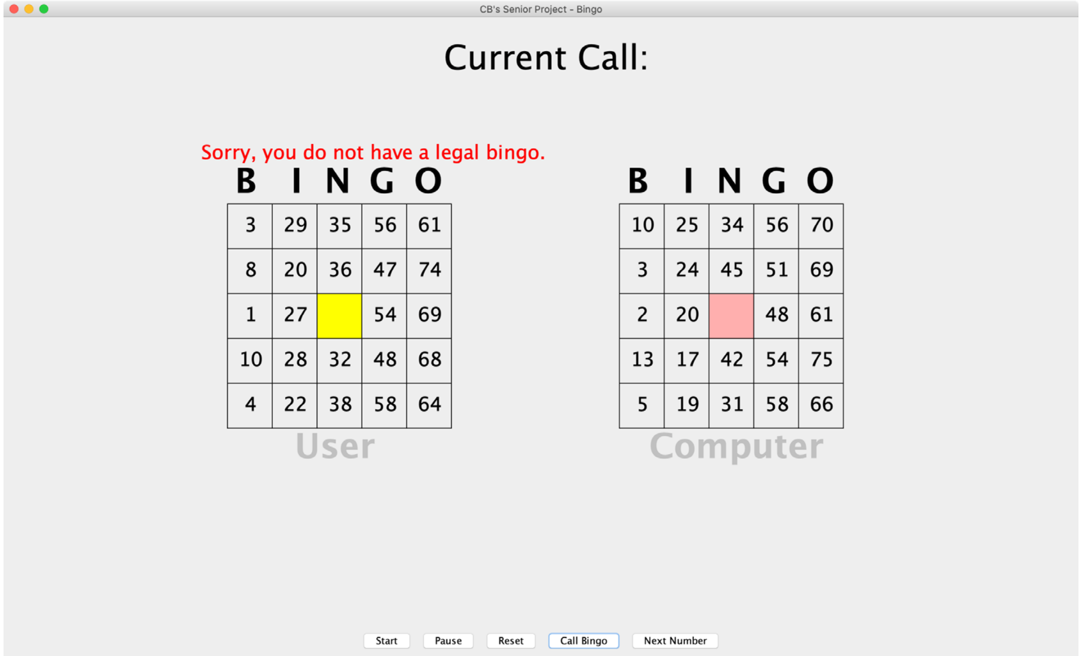
B	I	N	G	O
9	20	36	53	73
11	22	33	58	65
10	23		60	64
4	19	34	50	75
13	27	35	47	74

User

B	I	N	G	O
4	17	41	51	61
14	25	37	60	69
5	16		57	70
1	19	39	54	66
12	28	36	56	62

Computer

Photo 1: Blank Boards – When the app opens and the first set of boards are generated, this is what the user sees. They have access to all the buttons at the bottom and they can mark squares on the board to the left, labeled "User."



Current Call:

Sorry, you do not have a legal bingo.

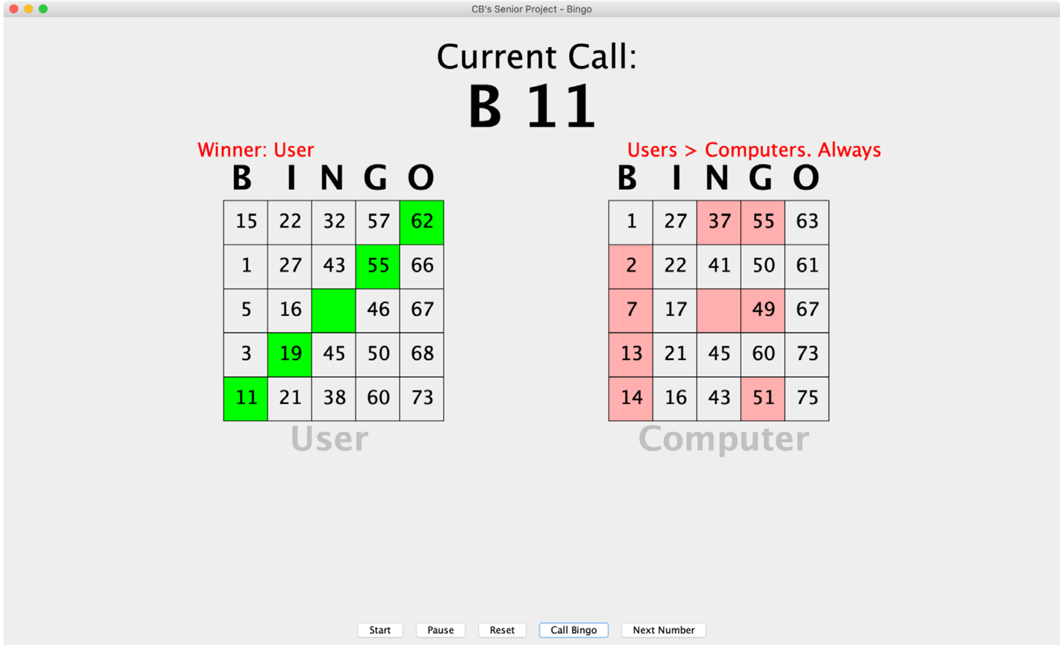
B	I	N	G	O
3	29	35	56	61
8	20	36	47	74
1	27		54	69
10	28	32	48	68
4	22	38	58	64

User

B	I	N	G	O
10	25	34	56	70
3	24	45	51	69
2	20		48	61
13	17	42	54	75
5	19	31	58	66

Computer

Photo 2: False Call – At any point of the game, if the user selects "Call Bingo" and does not have a real bingo to claim, the game will give the error shown here.



Current Call:

B 11

Winner: User

B	I	N	G	O
15	22	32	57	62
1	27	43	55	66
5	16		46	67
3	19	45	50	68
11	21	38	60	73

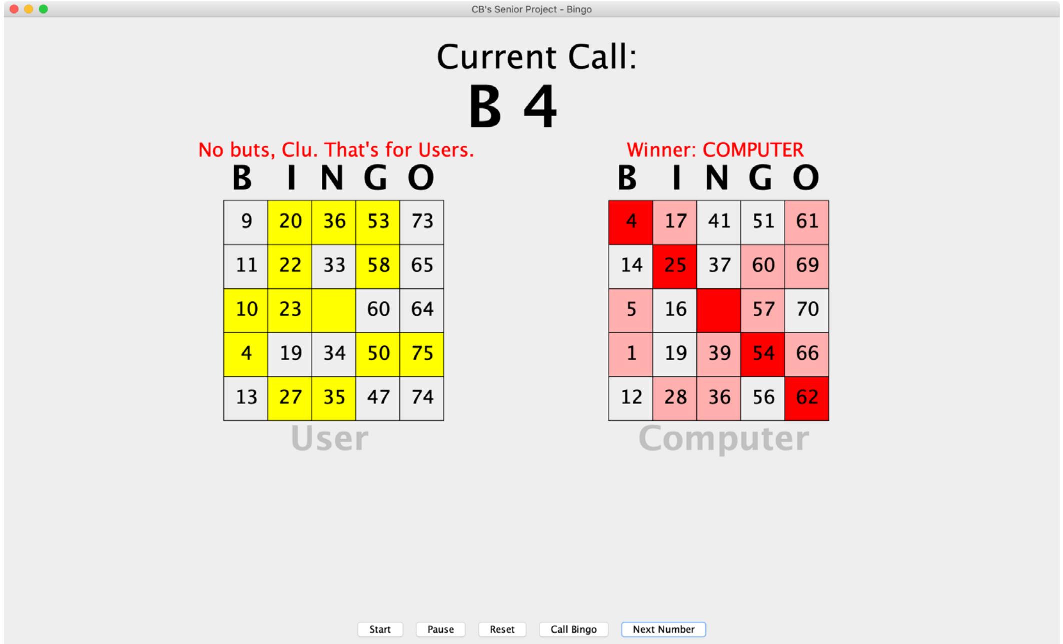
User

Users > Computers. Always

B	I	N	G	O
1	27	37	55	63
2	22	41	50	61
7	17		49	67
13	21	45	60	73
14	16	43	51	75

Computer

Photo 3: User Win – When the user marks all numbers needed for a bingo on their board, the select "Call Bingo" and the game will go through and see if all highlighted squares are legal and if it makes a five in a row. If correct, it shows the above messages.



Current Call:

B 4

No buts, Clu. That's for Users.

B	I	N	G	O
9	20	36	53	73
11	22	33	58	65
10	23		60	64
4	19	34	50	75
13	27	35	47	74

User

Winner: COMPUTER

B	I	N	G	O
4	17	41	51	61
14	25	37	60	69
5	16		57	70
1	19	39	54	66
12	28	36	56	62

Computer

Photo 4: Computer Win – As the computer automatically highlights legal squares on its own board, as soon as it gets five in a row, it automatically claims bingo and shows the above messages.

Test Plan: For the test plan section of this Binder, I have included the Test Plan document submitted in November 2019 titled "Bingo App Test Plan Document". This spans the following two pages.

Christopher Bell

CSCI 498

Bingo App Test Plan Document

Introduction:

As a brief summary of the project being built, this is an application that allows one person to play the game of bingo in a competition, race type setting with an automated computer player. This app allows said player to daub his own board on an automatically generated bingo board and create new version of this game whenever they wish. Buttons included on this app allow the player to access new versions of the game, start and stop the game, progress the game in step increments, and call bingo if they think they have it.

References:

Related documents include:

- The Application (app/*whatever name I decide on*)
- Project Proposal (docs/Project Proposal (v.2))
- Design Document (docs/Product Design)

Features to be Tested:

- Game boards and their interactivity: The two players game boards should both be able to automatically generate bingo boards that would be considered legal to play on, however only the left board should be interactive as the right board is played on by the computer

- Start Button: The button should start a new game and produce randomly ordered bingo letter/number combinations. Should do nothing when the game finishes.
- Stop Button: The button should stop the letter/number combinations, essentially pausing the game.
- Step Button: The button should progress the letter/number combinations by one each time the button is pressed. Should not progress once the game is finished.
- Call Button: The button should check whether or not the human player has a legal bingo or not and lets the player know what to do from that point on.
- Reset Button: The button should start the game over, regenerating fresh boards for both the user and computer sides.

Suspension Criteria and Resumption Requirements:

- None. Most testing can be done intermittently and can be tested whatever stage the app is in.

Test Deliverables:

- This Document
- Screenshots
- Test Logs
- Test Reports

Test Environment:

- Hardware: MacBook Pro (Retina, 13-inch, Early 2015)
- OS: 10.15.1 Beta (19B77a)
- Software: NetBeans IDE 8.2 (Build 201609300101)
- Java 1.8.0_141

Schedule:

- User Interface Testing: make sure that two bingo boards and five buttons are visible and are linked to the right information label and coding wise – 12/6/19
- Bingo Board Testing: check autogenerating to make sure that the boards and numbers that are filled in are unique and random; check to make sure that only the left board can be daubed by a user – 12/9/19
- Reset Button Testing: check that clicking reset regenerates the boards to contain new, unique combinations each time – 12/12/19
- Start Button Testing: check that clicking start begins the letter/number calls and that those calls are random, legal, and unique – 12/17/19
- Stop Button Testing: check that stopping the game only pauses the game and doesn't erase all progress up to that point – 12/19/19
- Step Button Testing: check that stepping goes to the next call in sequence rather than creating a new sequence – 12/22/19
- Call Button Testing: check that calling bingo confirms five in a row and that those five are legal numbers that have been called – 12/29/19

- Reset Button Testing Part 2: check that resetting the game fully resets the game and erases past letter/number calls – 1/3/20

Responsibilities:

- All generic testing will be tested by Christopher Bell, creator of this application
- Usability testing to make sure that players can play while making sense will be tested by friends and family members.

Assumptions and Dependencies:

- The test schedule above assumes that coding will be done on time by the end of November. The test schedule also assumes that there will be time needed in between each item to correct any flaws that may arise.

Test Results:

As I carried out the test plan listed in the attached document, I encountered only a few bugs, but those bugs affected the timeline significantly as they were heavy problems that needed fixing (see "Challenges Overcome").

Challenges Overcome:

This project has been filled with challenges and changes as time has unfolded. Originally, this was supposed to be a music-based bingo that allowed a user to listen to a .mp3 file of a song and then they would find that song title on their board. I had the idea for this application back when I hosted a music trivia game that had people do this in real time on paper. I was planning on using materials from my job (.mp3 tracks and boards) until the company revamped the game in a totally new way and in the process, I lost access to all the files. While creating earlier versions of the music bingo game, I focused on elements of the bingo game itself first and decided it would be easier to add music to it later and when I realized I lost all of my resources, I ended up continuing with the bingo theme and trashing the music side of it.

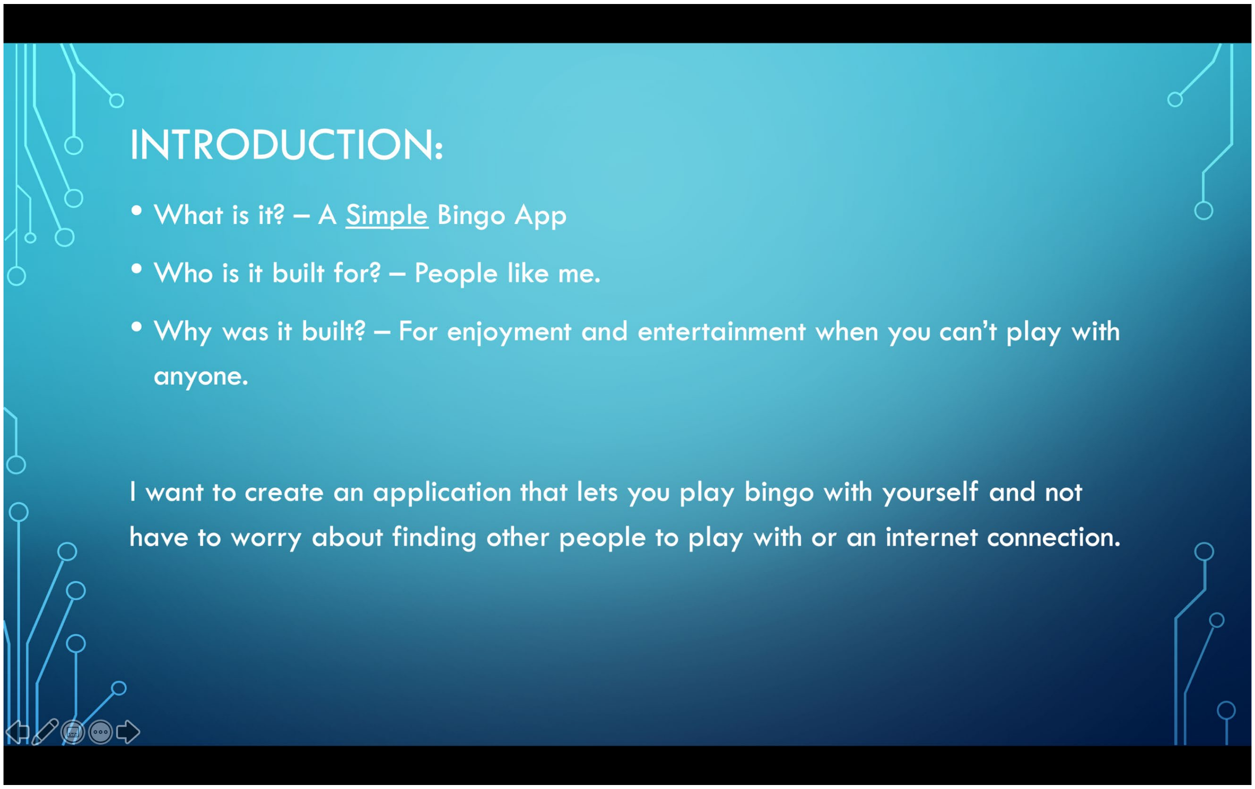
Other challenges I overcome while making this project included figuring out how to relay switches and make register that only certain switches are selected. For some reason, the game boards decided it didn't want to register the User's selections as it was not sure that those letter/number combinations had been registered as "legal." The reset function at the bottom was also not working until I figured out how to randomly generate boards WHILE also making them work as planned. For a short time, I had to quit and restart the application every time I wanted to play.

Personal challenges that set me back included getting two sinus infections and a global pandemic that rearranged a lot of my schedule.

Future Enhancements:

- An additional version that lets you play with music clips and titles rather than numbers and letters.
- A call history list or board that shows every letter/number combination that has been called since the last reset of the game.
- A multiplayer version of the game that could have two users play at the same time, one player to use multiple boards, or even one player to go up against several computer players.
- A P2P version of the game that allows multiple players on different devices play.
- A choice of color dauber.

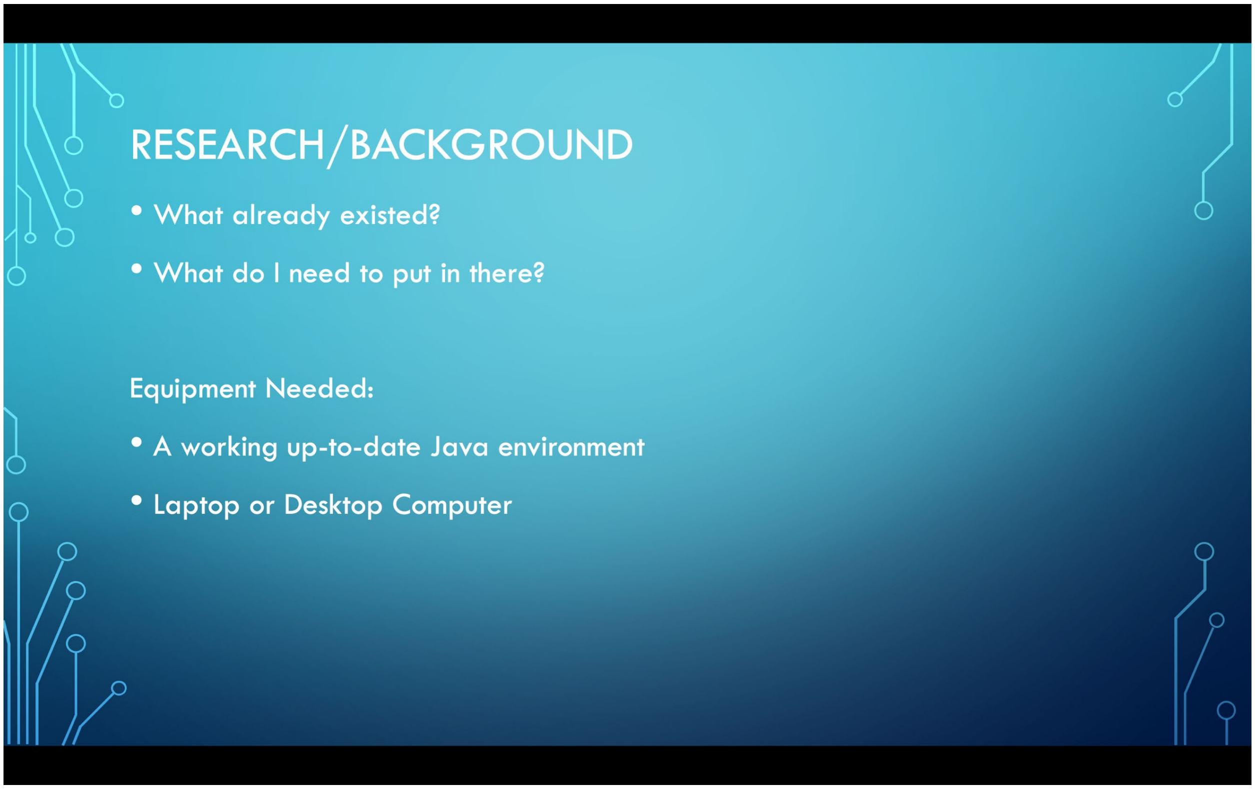
Final Defense Presentation Slides:



INTRODUCTION:

- What is it? – A Simple Bingo App
- Who is it built for? – People like me.
- Why was it built? – For enjoyment and entertainment when you can't play with anyone.

I want to create an application that lets you play bingo with yourself and not have to worry about finding other people to play with or an internet connection.

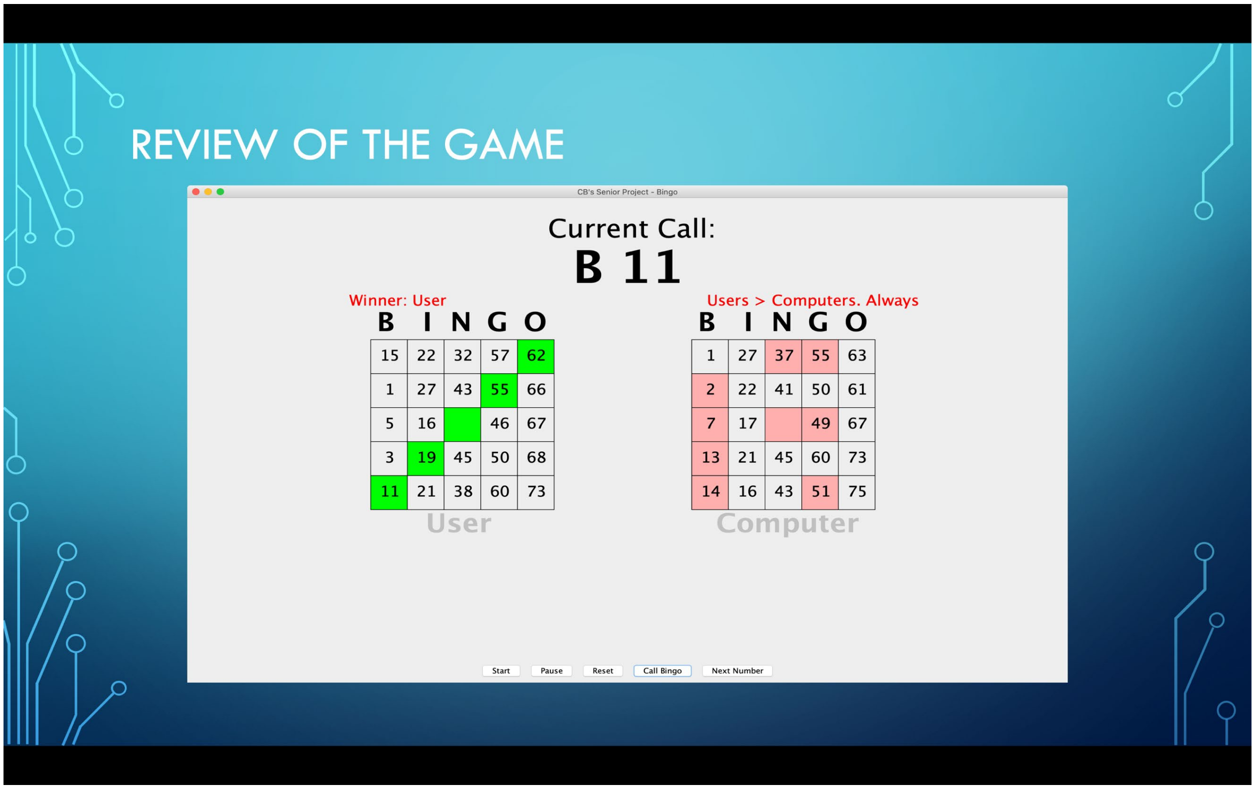


RESEARCH/BACKGROUND

- What already existed?
- What do I need to put in there?

Equipment Needed:

- A working up-to-date Java environment
- Laptop or Desktop Computer



REVIEW OF THE GAME

Current Call:

B 11

Winner: User

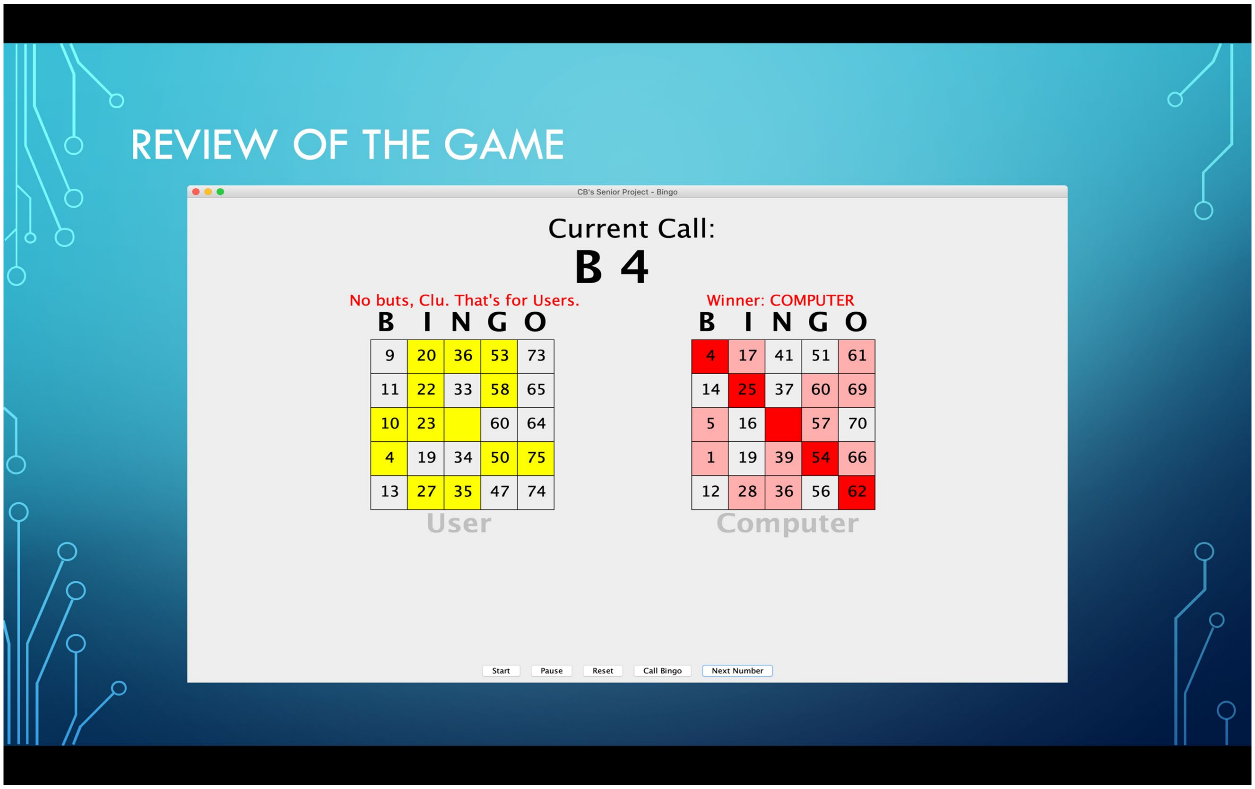
B	I	N	G	O
15	22	32	57	62
1	27	43	55	66
5	16		46	67
3	19	45	50	68
11	21	38	60	73

User

Users > Computers. Always

B	I	N	G	O
1	27	37	55	63
2	22	41	50	61
7	17		49	67
13	21	45	60	73
14	16	43	51	75

Computer



REVIEW OF THE GAME

Current Call:

B 4

No buts, Clu. That's for Users.

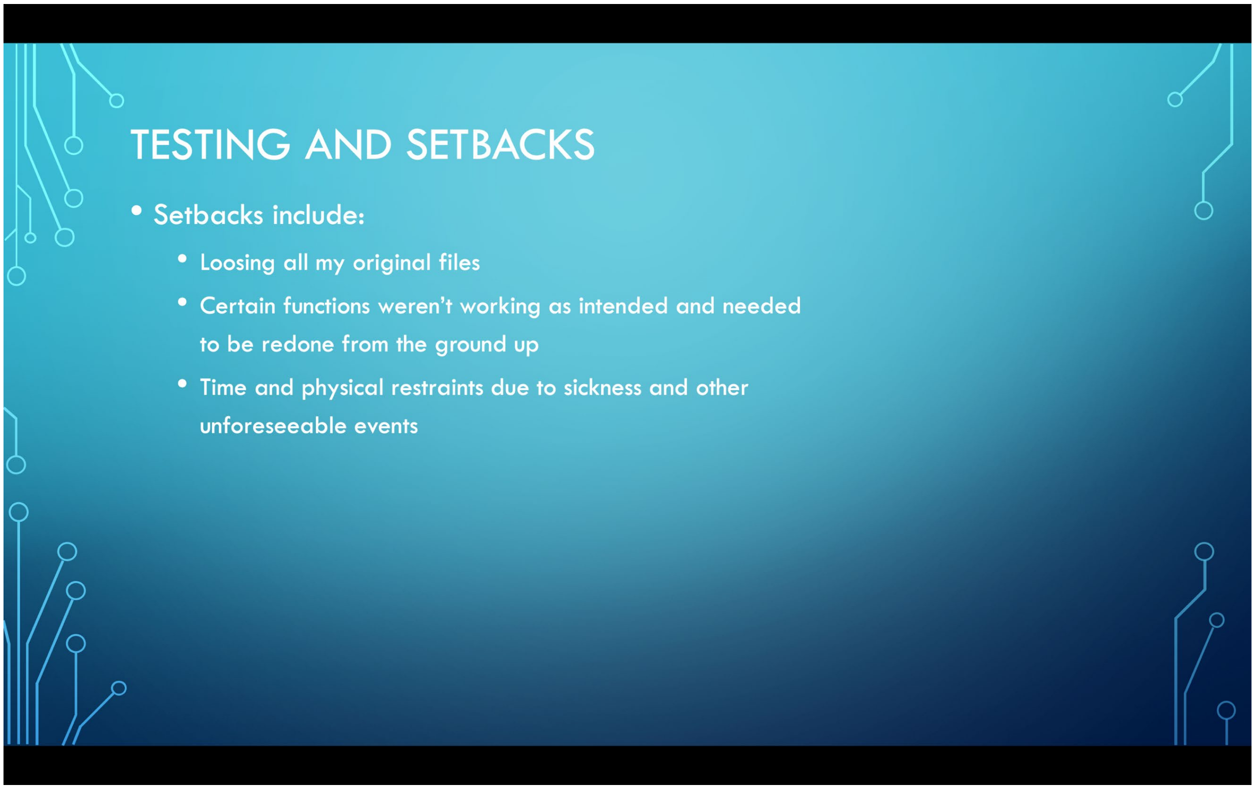
B	I	N	G	O
9	20	36	53	73
11	22	33	58	65
10	23		60	64
4	19	34	50	75
13	27	35	47	74

User

Winner: COMPUTER

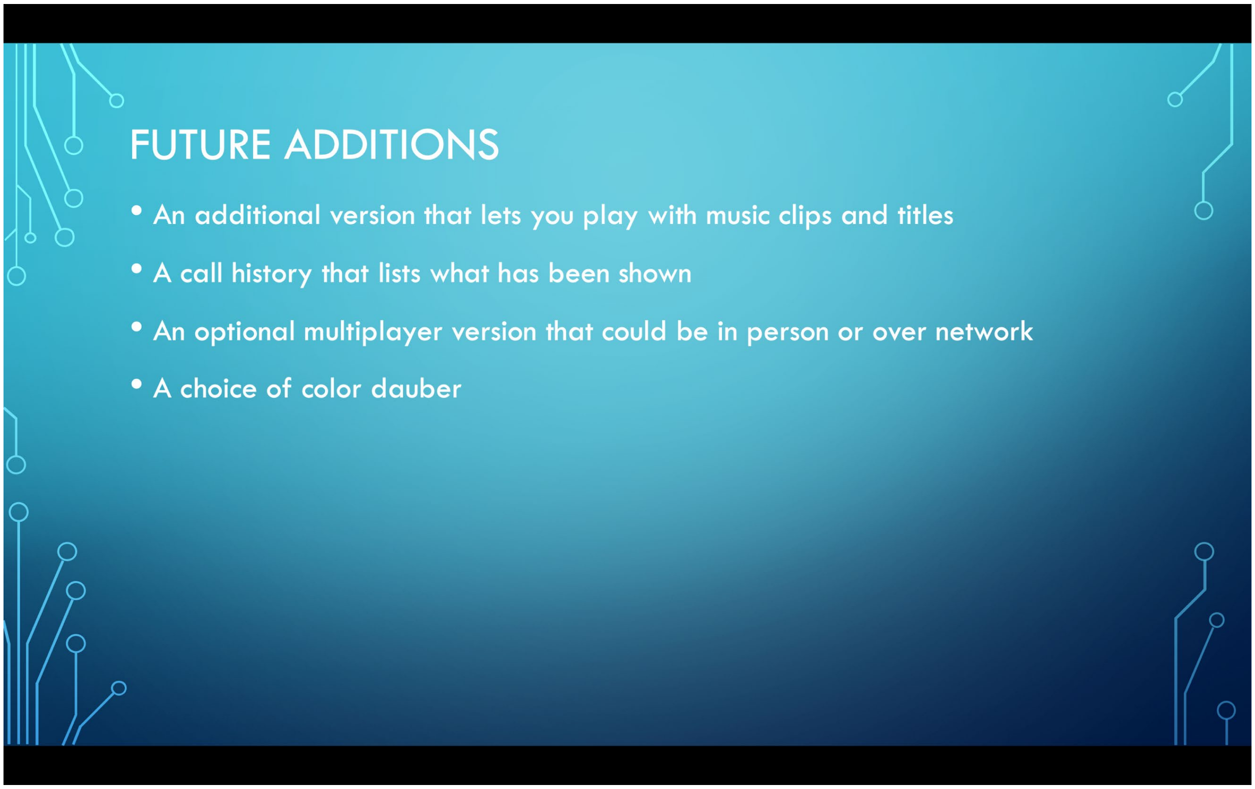
B	I	N	G	O
4	17	41	51	61
14	25	37	60	69
5	16		57	70
1	19	39	54	66
12	28	36	56	62

Computer



TESTING AND SETBACKS

- Setbacks include:
  - Loosing all my original files
  - Certain functions weren't working as intended and needed to be redone from the ground up
  - Time and physical restraints due to sickness and other unforeseeable events



FUTURE ADDITIONS

- An additional version that lets you play with music clips and titles
- A call history that lists what has been shown
- An optional multiplayer version that could be in person or over network
- A choice of color dauber