

VirtuCards

CS 30700 Sprint 3 Retrospective



TEAM 8

June Seo

Ryan Hawks

Aryan Wadhwani

Kade Boltjes

Shayne Marques

Umang Sharma

What went well during Sprint 3?

The time spent determining a specific style and developing prefabs for UI components paid off this sprint as we were able to efficiently and quickly complete tasks associated with both the project's back-end and front-end.

We were able to incorporate more quality-of-life features such as user profile customization through user profile avatars, the ability to play music from within games and sound effects to indicate specific actions within the game. We were able to also add more scripts to provide users with more games to play with unique features - specifically Uno (where special cards can make other users draw cards and skip their turn) and Poker (where users can fold their hands to not interact any further within the game round).

Thanks to the introduction of a bug tracking spreadsheet, we were able to better approach debugging the project. There were relevant sections added to include a member whose area of expertise the bug was in, a field to add who was working on the bug currently (so that multiple people didn't resolve the same bug) and a legend which indicated the priority / severity of the bug and whether it was resolved or not. The spreadsheet helped improve the debugging procedure and helped the team dedicate more time to it.

The advanced research in acceptance criteria helped as our team didn't need to make significant changes to the acceptance criteria to the planning document before the final sprint review.

The change in workflow on documented test cases helped complete them in one go and free up time in the latter part of the sprint for rigorous bug-fixing.

All of the improvements, as well as more familiarity with Unity, helped us tackle an increased number of user stories this sprint. We completed 3 more user stories for a total of 21.

User Story #1

As a user hosting the game, I would also like the option to invite my friends to join my game.

#	Description	Estimated Time	Owner
1	Add an invite player button in the waiting screen	1 hour	Ryan
2	When the invite player button is pressed, a panel appears. This panel lists all the user's friends with buttons to invite each one individually. Additionally a button labeled "Invite All" is placed at the top of the panel to invite all the friends.	3 hours	Ryan
3	When a specific player is chosen to invite, send a message to them allowing them to join the game unless they are already in the waiting room of the game.	3 hours	Ryan
4	On clicking the "Invite All" button, send an invite to all the players in the host's friends list that are online.	1 hour	Ryan
5	Create a test case to show that the option to invite friends is implemented correctly	1 hour	Ryan

Completed

In the game waiting room there is now a button for the host to invite players on their friend's list. When the user clicks the invite player panel, they are greeted with a panel that displays a dropdown of all of their friends. The host can choose to invite someone specifically or invite all of the players on the list. When a client receives this invite, a panel pops up showing the room details and gives them the opportunity to accept the invite or decline the invite.

User Story #2

As a user hosting the game, I would like to have the option to play another game with the same players at the end of a game.

#	Description	Estimated Time	Owner
1	Create a panel that shows to play again button after the game ends	1 hour	Ryan
2	When the play again button is clicked all of the game info is reset (cards, score etc)	2 hours	Ryan
3	When the host clicks the create new game button the clients are automatically put into the new game	2 hours	Ryan
4	Create a test case that verifies the host can play again	1 hour	Ryan

Completed

After a game is played, there is a button for the host to play again with all of the players in the lobby on the same gamemode. When this is clicked, the entire game board is cleared and all of the player's hands are reset. If anyone leaves before the host clicks the "play again" button, the game will account for that and reformat the game board correctly.

User Story #3

As a user hosting the game, I would like to have songs being played from a queue from the common external screen.

#	Description	Estimated Time	Owner
1	Add a music button on the host game screen, which opens a panel on the side displaying the currently playing song and the songs in the queue.	4 hours	Aryan
2	Add a button to pause music playback in the music panel and to trigger music playback, also functionality.	3 hours	Kade
3	Add UI for a Skip Song button.	1 hour	Kade
4	Implement the Skip Song button's functionality to skip the song currently playing.	3 hours	Aryan
5	Once a song has finished playing, remove it from the queue and start playing the next one.	2 hours	Kade
6	Create a test case that ensures that a song can be played from the queue.	1 hour	Kade

Completed

When the host has songs in the queue, it plays them sequentially. After one finishes, the next plays. When the host presses the pause button, it pauses the current song playing. When the host presses the skip song button, it automatically skips to the next one.

User Story #4

As a user hosting the game, I would like to be able to remove songs from the queue played by the shared screen.

#	Description	Estimated Time	Owner
1	Add a remove button for each song in the queue	1 hour	Kade
2	Add a method to remove the song in the queue	2 hours	Kade
3	If the song being removed is currently playing, skip to the next song	1 hour	Kade
4	Add a 'Clear Queue' button	1 hour	Kade
5	Implement the functionality for the 'Clear Queue' button.	1 hour	Kade
6	Add a test case to test whether a song can be removed from the queue.	1 hour	Aryan

Completed

When the host has songs in the queue, each song can be individually removed, or they all can be removed with the clear button. When the user removes the song that is playing, it automatically skips to the next one. If the clear button is pressed, all the songs are removed and the current song playing is stopped.

User Story #5

As a user hosting the game, I would like to be able to shuffle the songs currently in the queue if time allows.

#	Description	Estimated Time	Owner
1	Add a shuffle button to the playlist panel	1 hour	Aryan
2	Reorder the queue to reflect the new order of songs playing back.	2 hours	Aryan
3	When the shuffle button is pressed, start playing the new song at the front of the queue	2 hours	Aryan
4	Add a test case to ensure that the songs in the list can be shuffled.	1 hour	Aryan

Completed

A shuffle button is present in the playback options. Upon clicking it, the queue of songs is shuffled and the next song in the beginning of the queue begins playback.

User Story #6

As a user hosting the game, I would like to choose custom backgrounds for the table.

#	Description	Estimated Time	Owner
1	Add a 'Change Background' button.	1 hour	Shayne
2	Add a 'Default Background' button	1 hour	Shayne
3	Add an open file dialog that appears when the 'Change Background' button is pressed that can only select pictures	3 hours	Shayne
4	Display the new selected background in the Game Screen	2 hours	Shayne
5	Add a test case that verifies the background can be changed.	1 hour	Shayne

Completed

When the host presses the settings button, they are given the ability to change the background of the game screen. When the change background is pressed, an open file dialog appears that prompts the host to select an image. Once they have set a background, a button becomes enabled that allows them to reset the background to the default one.

User Story #7

As a user, I would like to select an avatar from a given list to represent me.

#	Description	Estimated Time	Owner
1	Create 5 avatars to be selected from	3 hours	Umang
2	Create a 'Choose Avatar' button, opening a choose avatar panel upon click. Also display the avatars to choose from.	2 hours	Umang
3	Implement the functionality for the 'Choose Avatar' button. This includes showing it on their profile screen.	3 hours	Umang
4	Store the user's avatar selection on Firebase	2 hours	Umang
5	In the host game screen, the new avatar is displayed for the User.	2 hours	Umang
6	Create a test case that checks a player can select their avatar.	1 hour	Umang

Completed

The Avatar designs were created using an online tool called AvatarMaker. 6 designs were made and imported into the Resources subdirectory of the VirtuCards projects. Originally, the Avatars were going to be loaded onto a panel but to improve usability, the panel was replaced with a carousel to allow users to swipe through all the Avatars to view and select them. The Avatar was originally stored as a Base64 string but that was too large for a string on Google Firebase so we added support to store images in our database. The image was then stored in the Avatar field of the user's respective UserProfile object which contains all their information.

User Story #8

As a user, I would like to upload a custom photo as my avatar.

#	Description	Estimated Time	Owner
1	In the Choose Avatar panel, add a button to open an image stored on the device	3 hours	Umang
2	Implement the same functionality as selecting a preset avatar.	1 hour	Umang
3	Store the user's avatar selection on Firebase	2 hour	Umang
4	Create a test case that checks a player can upload a custom photo as their avatar	1 hour	Umang

Completed

The feature is available in the Profile screen of the VirtuCardsClient. Once a user clicks the 'Edit Avatar' button, they are shown the Choose Avatar Panel which was created when working on the previous user story. This contains a 'Custom Avatar' button that now opens the device's native file browser, enabling the user to choose any desired image from their device as their avatar. This was stored in the user's respective UserProfile object in the Avatar field. Originally, this field was capped at 1 MB in Google Firebase which caused some bugs with larger images but the cap was raised to 30 MB.

User Story #9

As a user, I would like to have my avatar and other details linked to my account, allowing me to sign in to different devices with the same account.

#	Description	Estimated Time	Owner
1	Update Firebase with profile settings and selected avatar.	1 hour	Umang
2	Add functionality to import user preferences from Firebase when logging into VirtuCards.	1 hour	Umang
3	Create a test case to ensure that when a player changes their avatar on one device, it will be edited if the user logs in from a different device.	1 hour	Umang

Completed

Most of the Firebase update code was implemented when establishing the Preset Avatar functionality and was reused here. The code will store the user's desired image in the Avatar field of the instance of UserProfile retrieved from Firebase on login into VirtuCardsClient. This will then be pushed to Google Firebase as an update to the user's database entry which can then be retrieved on future logins.

User Story #10

As a user, I would like to search for my friends on VirtuCards with their usernames.

#	Description	Estimated Time	Owner
1	Add a search bar in the Friends screen.	1 hour	June
2	When the user types in the search bar, a list appears showing all the usernames that match	3 hours	June
3	Add a button to add the selected username as a friend	1 hour	June
4	Update the user's friends list in Firebase	2 hours	June
5	Create an error message for non-existent username	2 hours	June
6	Add a test case that verifies if users can search for their friends.	1 hour	June

Completed

The search for friends is available on the Client Friends screen and Host landing page. You need to search for the usernames of the player with correct spelling and capitalization. If the person does not exist, it will show a sign saying that the user does not exist, if the person is him/herself, it will show a sign saying the person cannot add themselves, and if the person does exist, the add button will appear. If the person is not a friend already, it will show a sign that says the person was added, but if the add button is pressed again, the sign will say you are already friends with that person. When the user searches for a player, the player's stats for VirtuCards will also appear below the name.

User Story #11

As a user, I would like to be able to accept invites from a game host to join their game.

#	Description	Estimated Time	Owner
1	Connect Host and Client to Photon “Waiting Room” chat upon joining lobby	2 hours	Aryan
2	Receive an event from the Client receives an invite from the Waiting Room chat	2 hours	Ryan
3	A notification panel is displayed with the host name, room name, and options to accept and decline.	3 hours	Aryan
3	When a client accepts the game invite, they are brought into the correct waiting room and leave the Waiting Room chat.	2.5 hours	Ryan
4	When a client rejects the game invite, the invite is disposed of.	1 hour	Aryan
5	Create a test case to ensure that a client can accept a host’s invite to their game.	1 hour	Ryan

Completed

When an invite is sent to a friend, a popup screen appears asking whether the player would like to join the game. The name of the host and the game being played is also included in the invite. If the client accepts the invite, they are taken to the waiting screen. When the client is not in the join game screen, the invite is not displayed.

User Story #12

As a user, I would like to be able to fold my hand.

#	Description	Estimated Time	Owner
1	Add a fold hand button in the game on the client.	1 hour	Shayne
2	Only show the button for games that allow hand folding like poker.	2 hours	Shayne
3	When the fold hand button is pressed, implement the functionality. For example, remove that player from the game logic (don't make it their turn or give them any cards)	3 hours	Shayne
4	Create a test case that ensures that player is able to fold their hand	1 hour	Shayne

Completed

When the player is in a game such as Poker, the fold button is displayed. When a player folds, their cards are removed from their hand and are only added to the deck after the round is over. If everyone but one player folds, that player is automatically selected as the round winner. If a player folds, and the round ends, they are dealt back into the game on the next round.

User Story #13

As a user, I would like to be able to see the number of games I have won and lost.

#	Description	Estimated Time	Owner
1	Add a display for the the number of games won and lost	1 hour	Kade
2	Make sure the display is up to date with the latest data stored in the database	1 hour	Aryan
3	Wins and losses are updated with each game played	1 hour	Aryan
4	Create a test case that checks if the profile data is being updated	1 hour	June

Completed

In the profile screen, information about games won, lost and played are tracked. This information can also be seen in the leaderboard and the friends list. The information is updated whenever a player starts a game, wins a game, or loses a game.

User Story #14

As a user, I would like to have the option of censoring profanity in the chat.

#	Description	Estimated Time	Owner
1	Add a 'Censor Profanity' toggle in the settings panel	1 hour	Umang
2	Add a list of profanity that shouldn't be sent.	2 hour	June
3	Create a warning message for users that use profanity in the chat	2 hours	Umang
4	Create a method that scans for profanity in the sent chat message after "Send" is pressed.	3 hours	June
5	Create a test case that ensures that profanity is filtered.	1 hour	June

Completed

We were successfully able to implement censoring profanity in the chat. It didn't go as smoothly as planned because Photon had a chat censor implemented already, but we had to pay 48 dollars, so we manually made a list of profanity and had each text scan through the list. The user will get two very obvious warnings before the user is completely banned from the chat.

User Story #15

As a user, I would like to have default animated reactions I can use, like “Boiler Up!”, if time allows.

#	Description	Estimated Time	Owner
1	Create 2 animated reactions	3 hours	June
2	Create buttons to send the reactions to other users	1 hour	June
3	Add music in the background after the button as been pressed	2.5 hours	June
4	Add a cooldown before the reaction can be sent again	2.5 hours	June
5	Implement a signal that sends from one client to every other client in the game	2 hour	June
6	Create a test case that ensures the different reactions are sent and displayed correctly.	1 hour	June

Completed

Every user will be able to see the two IU and Purdue buttons on the bottom right of their device when they hide the chat. If one person in the chat presses one of those buttons, Purdue marching band and an animation will pop up on everybody’s device. After that, everybody will get a cooldown on their button and if they try to press one of the buttons again, a little sign will show up saying it is on cooldown.

User Story #16

As a user, I would like to be able to see where I stand compared to my peers.

#	Description	Estimated Time	Owner
1	Add a 'Leaderboard' button to the Profile page	1 hour	Kade
2	Add UI for a Leaderboard page	2 hours	Kade
3	Retrieve each friend's statistics from Firebase	2 hours	Kade
4	Display the ranking of users in a list box, sorted by wins, losses, or games played.	3 hours	Kade
5	Create a test case that verifies the leaderboard works properly	1 hour	Kade

Completed

When a user is signed in to a profile, not anonymous, they can click the leaderboard button. When they do, they are brought to the leaderboard page where they can choose to sort their friends by games won, lost, and played. When clicking a friend's name in the leaderboard, that friend's statistics are shown in more detail below the leaderboard.

User Story #17

As a user, I would like to choose a custom card sleeve/back for my group or myself.

#	Description	Estimated Time	Owner
1	Add a 'Change Card Sleeve/Back' button.	1 hour	Shayne
2	Add a 'Default Card Sleeve/Back' button	1 hour	Shayne
3	Add an open file dialog that appears when the 'Change Card Sleeve/Back' button is pressed that can only select pictures	2 hours	Shayne
4	Display the new selected Card Sleeve/Back when cards are flipped over.	2 hours	Shayne
5	Add a test case that verifies the card sleeve can be changed.	1 hour	Shayne

Completed

When the client joins the game, they can open the settings panel and select a custom cardback from their gallery. This cardback is now visible to them. They can click another button to reset to the original cardback. The host can similarly pick a cardback, and the selected cardback by the host is also shared with all clients. The cardback is stretched to fit the size of the card.

User Story #18

As a user, I would like to add songs to the queue played by the shared screen.

#	Description	Estimated Time	Owner
1	Check if the message sent by the client matches the “!play <song_name>” format, if so, redirect the message only to the host.	2 hours	Aryan
2	On the host, search the song specified by the user on Youtube and add the best match.	5 hours	Kade
3	Check if the song specified is able to be found and display an error accordingly if not.	3 hours	Aryan
4	Update the host’s queue UI to include the newly added song	1 hour	Kade
5	Ensure the displayed song queue is in order of insertion	1 hour	Kade
6	Send a notification on the client that a song has been add	2 hours	Aryan
7	Create test cases that ensure songs are added to the queue when players add more songs.	1 hour	Aryan

Completed

After a game starts, the players can request songs by typing !play song_name in the chat. When the song is added, a notification on the host and the clients will appear stating the same. The song will be added last in the queue of songs to be played on the host, or begin playing if it is the only song in the queue. Requests greater than 7 minutes in size will be rejected.

User Story #19

As a user, I would like sound effects when an action happens on the screen.

#	Description	Estimated Time	Owner
1	Create or choose sound effects for the different actions	2 hours	Umang
2	Add script components that play the required sounds when the action is performed.	2 hours	Umang
3	Create a test case that ensures that the sound effects occur in sync with the game actions.	1 hour	June

Completed

When a game starts, a sound effect is played to alert all connected players. If a player plays a card, or skips their turn a different sound is played to notify the player. Also if a player is not allowed to play a specific card, but they attempt to anyways, a sound effect will play alerting them that it is prohibited. Additionally, a sound effect is played upon the winner being declared.

User Story #20

As a user, I would like haptic feedback when it is my turn to play or I make an illegal move.

#	Description	Estimated Time	Owner
1	Implement the haptic feedback functionality of an appropriate strength and duration.	3 hours	Shayne
2	Make a method to create some haptic feedback	1 hour	Shayne
3	Add script components that generate haptic feedback through the method when required.	2 hours	Shayne
4	Create a test case that ensures that the haptic feedback is triggered correctly.	1 hour	Shayne

Completed

A haptic feedback is implemented, such that it is generated whenever the client attempts to play an invalid card. It is also generated whenever it becomes the client's turn.

User Story #21

I would like to be able to kick players from the game when necessary.

#	Description	Estimated Time	Owner
1	Create a kick player button	1 hour	Ryan
2	Add a drop-down to select players from.	2 hours	Ryan
3	Upon selecting a player to kick, Raise an event notifying the player they have been kicked.	2 hours	Ryan
4	Create a panel that shows that a player has been kicked from the client side	2 hours	Ryan
5	Send the player to the Join Game page.	2 hours	Ryan
6	Create a test case to ensure that a player is kicked from the game when necessary.	1 hour	Ryan

Completed

Once a host has started a game, there is a button for the host to kick players. When the host clicks this button, they are greeted with a panel that has a dropdown of all the players currently in the lobby. The host can select one of those players to kick. Once a player has been kicked, the game will be reformatted as needed. Once a client has been removed from the game a pop-up panel will show up on their screen telling them they were kicked from the game.

What did not go so well in Sprint 3?

Overall, we think this was a great Sprint in terms of completing more than the required amount of tasks, bug fixes and presenting during the Sprint Review. We all worked well together and completed our individual tasks and test cases in a timely manner.

Due to the nature of our application being a game, there are a lot of edge cases and random scenarios that are unaccounted for during development. This leads to us noticing a considerable amount of bugs. While we were able to fix a good amount of bugs, there are circumstances we cannot consider as developers while play-testing our game.

Being we had so many tasks, we did struggle towards the end to complete them all on time. They were all completed, but some were finished after our rehearsal, which introduced a couple uncaught bugs. Furthermore, as our game is a multiplayer game, there are various components of the game that can only be tested with groups of players, and not one developer. This makes it harder to detect bugs, since despite having nearly daily meetings during the sprint, group meetings are yet too short to find all the existing bugs.

A better system of rigorously testing our project could be better as a fresh perspective during testing may help locate more weak points within the project.

How should we improve?

For future project work, we've learnt that better bug tracking would be very useful. Currently, we have used a shared Google Sheet to report, test and eliminate bugs. Having bug tracking integrated with Github Issues, which we used for project management, would be valuable.

Setting milestones for each week to achieve would also be valuable in ensuring a consistent effort is placed. Also, having a set meeting where we do no coding whatsoever and just do playtests for finding bugs would be useful for finding edge cases.

Another consideration during future projects is better planning while setting up the repository. Some of the files currently in the repository simply trigger merge conflicts, and don't have any real importance to the project. Furthermore, some of the files in the repository are user path-dependent, and hence should have not been added to the repository. These files lead to issues in importing assets in Unity, and the reimporting process which Unity undergoes can take above 10 minutes. Originally, we were willing to reimport all the assets given the time for the procedure wasn't long but it took longer and longer as the semester went on.