Orbital 2018 Milestone 3

Gan Chin Yao | Lim Wai Lun

Proposed level of achievement: Apollo 11

Background

This is a project done under National University of Singapore (NUS) Orbital Programme in 2018 by two Year 1 NUS students, Gan Chin Yao and Lim Wai Lun. Our project is an Android game developed in Android Studio during the summer holidays.

Overview of the game

The game can be played either in Single Player Mode, where the player plays alone, or in Multi-Player Mode, where player gets to play with their friends on different devices, or random people who are seeking for opponents. In the Single Player Mode, player tries to score points by guessing a word. A description, or question, will be shown to the player during each game round, and player proceeds to guess the actual word. Scores are awarded for each successful guess, upon which the next question will appear and player will proceed to guess the next question. The game ends in 105 seconds. Elements of the gameplay includes the ability to skip current question, requesting hint for current word, as well as attempting to score Streak to further boosts the player's score.

In the Multi-Player Mode, player competes with up to 7 other opponents in real-time. Player emerges victorious by scoring the highest points at the end of the entire 105

seconds. The gameplay is similar to the Single Player mode, with the exception that players can cast Spells. Spells would either give the caster an advantage, or cause disruption to opponents who are attempting to guess their own words.

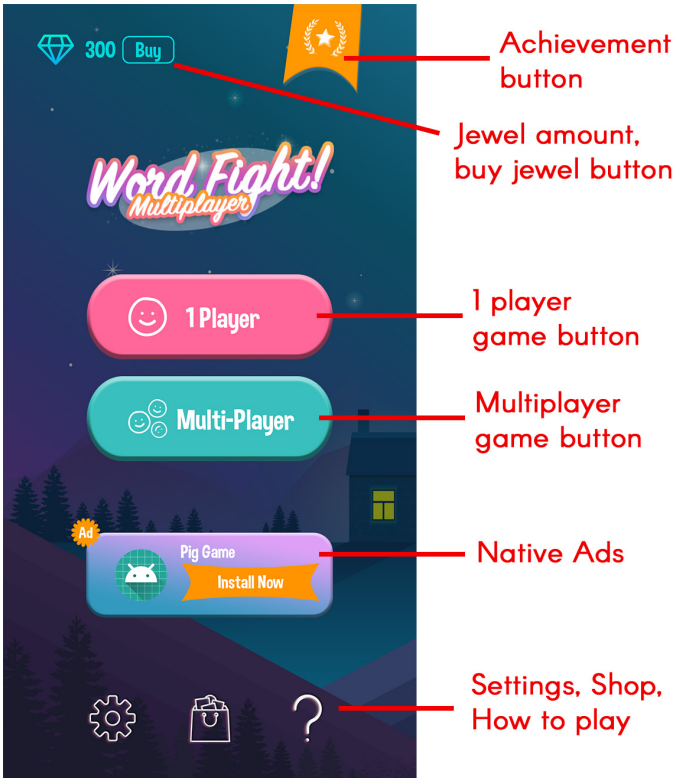
Objectives and target audience for the game

As this is a game, the main purpose is to allow players to enjoy playing the game. As there are little multiplayer titles in the Google Play Store, our team decides to incorporate multiplayer element so that players can play together with their friends, and have a great time. Since this is a Word game, players may even pick up some English vocabulary while playing the game, or gain some knowledge based on questions asked in the game. The intended targeted audience for the game is for all Android users. The game will be published in the Google Play Store in all available countries. There is no age restriction. Everyone from all walks of lives and from any countries, should they be English literate, will be able to enjoy this game.

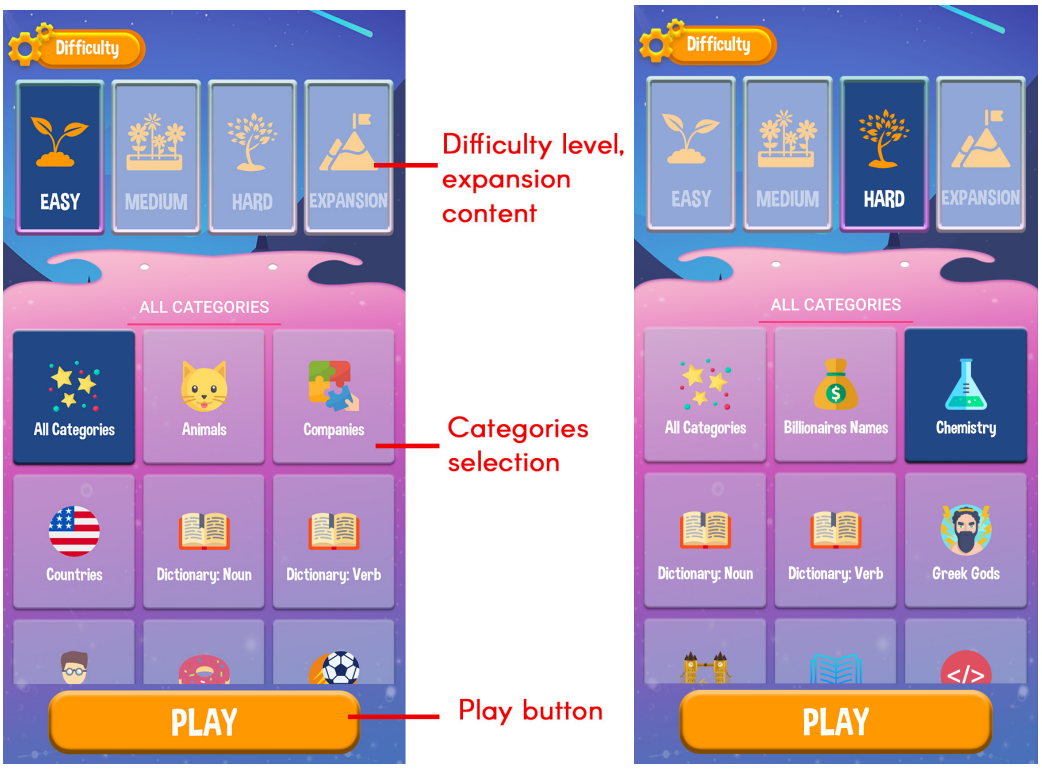
Features of this game:

The following are features that have already been completed as of Milestone 3.

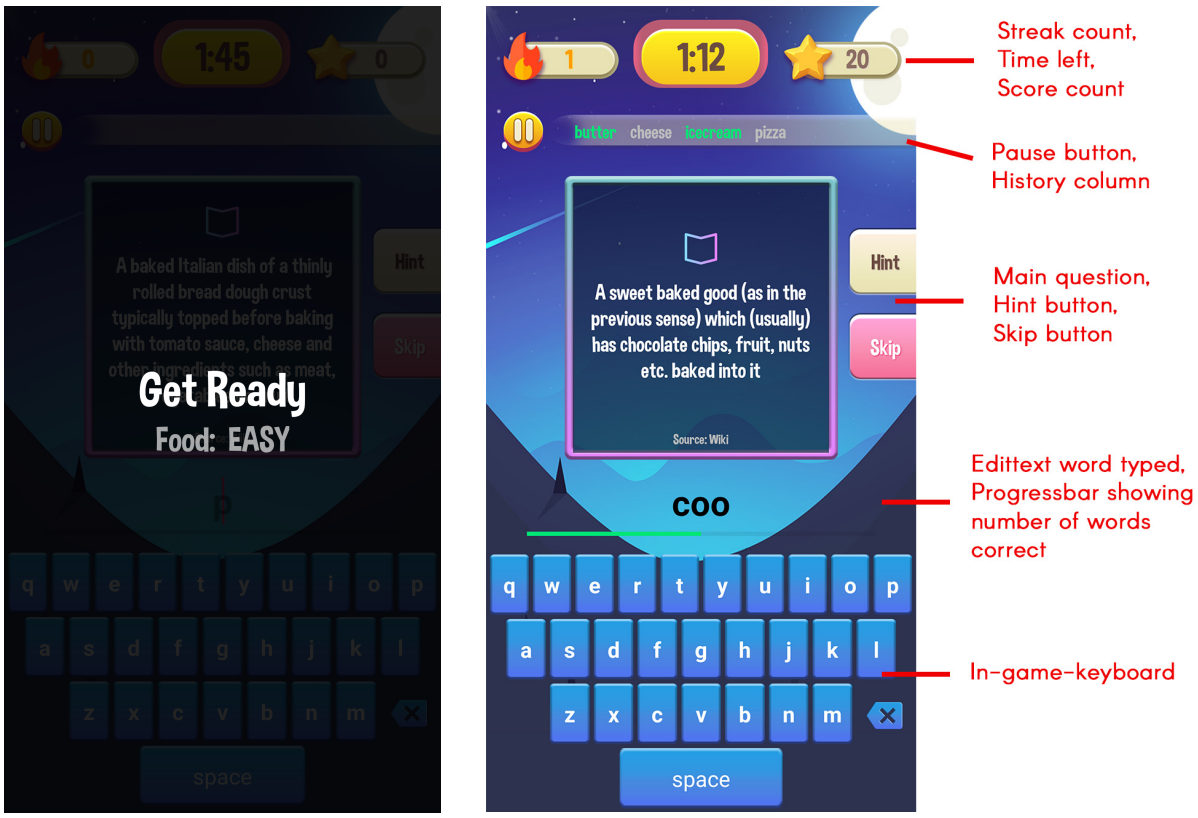
1. Main Screen. Users are greeted with this screen on launching the application

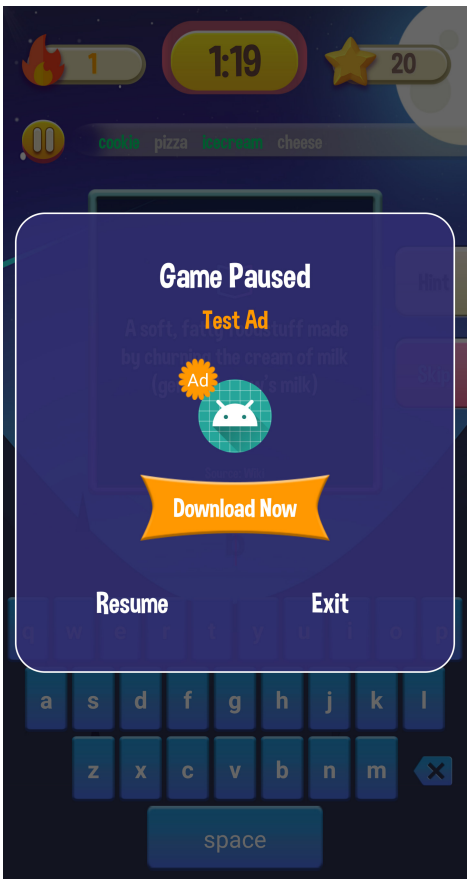


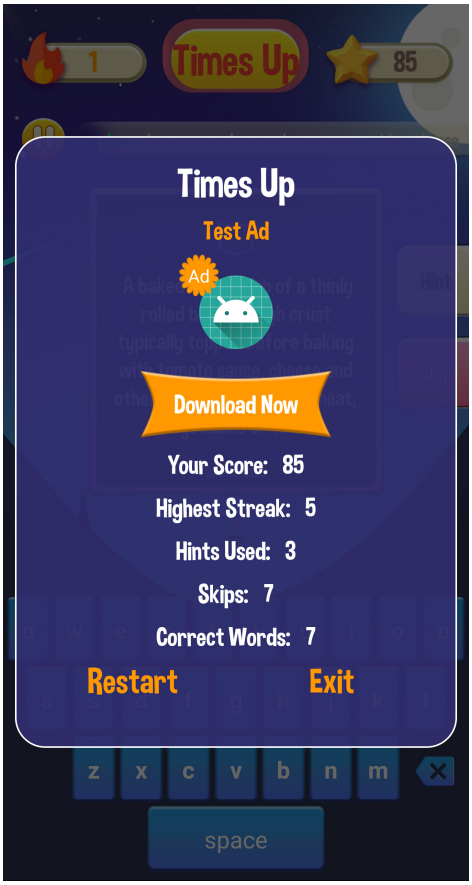
2. Category selection page. Upon clicking the 1 player game button, player will be presented with this page.



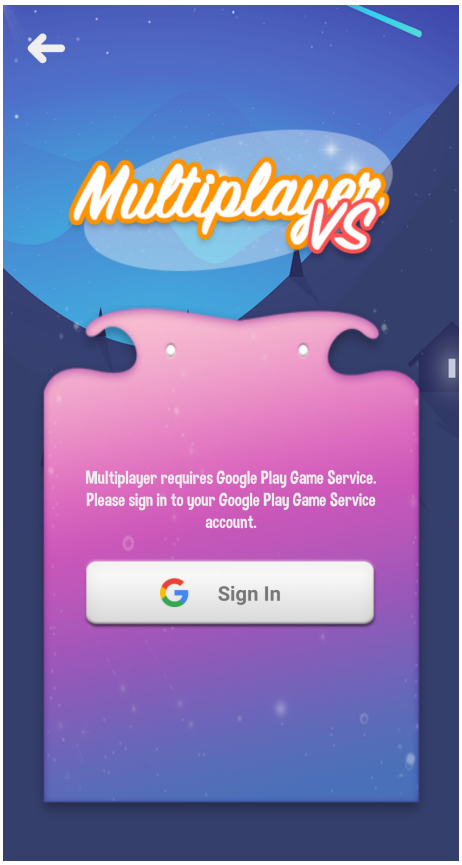
3. 1Player game play.

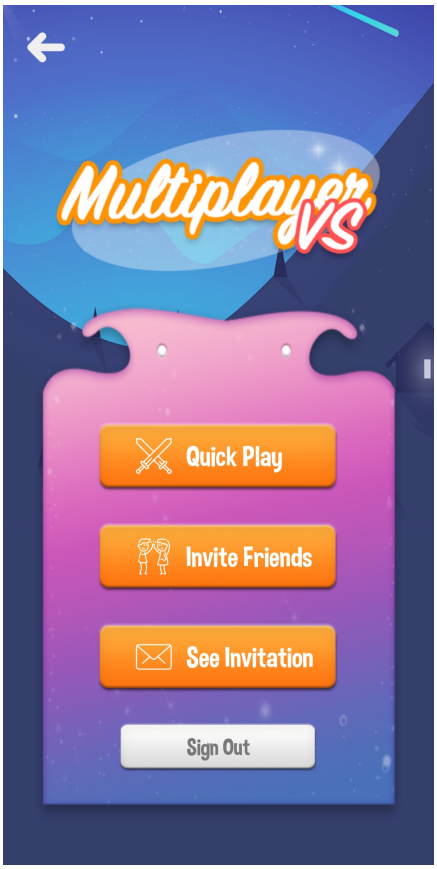


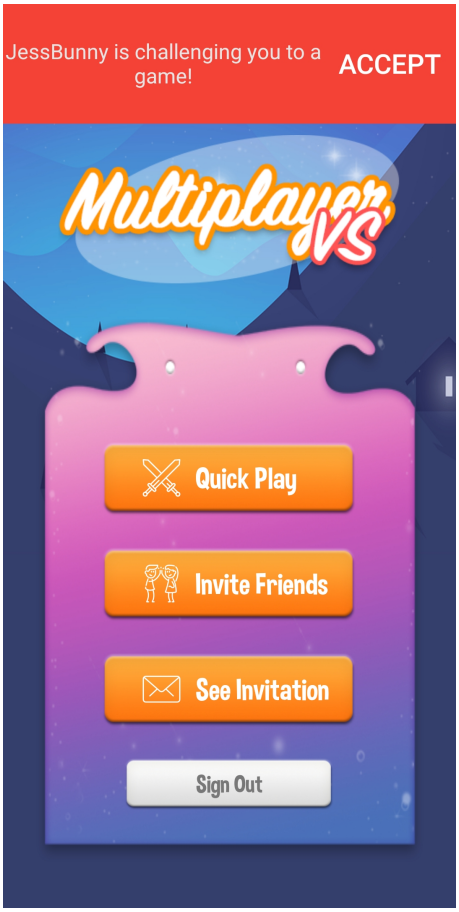


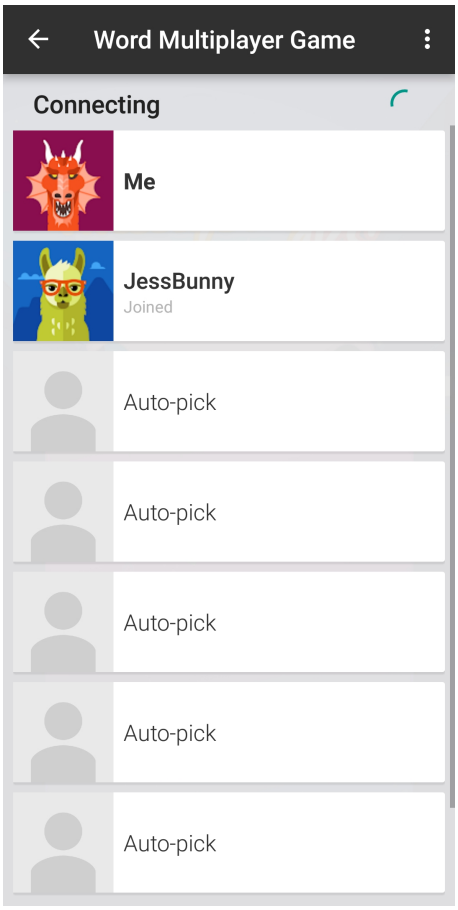


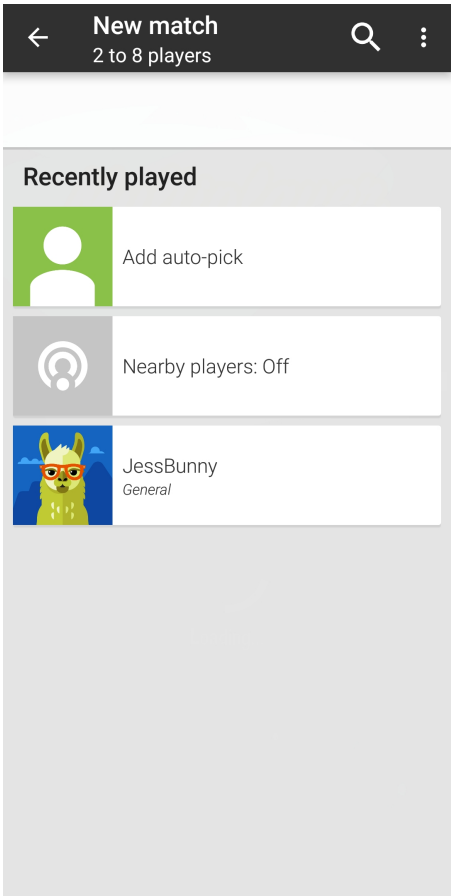
4. Multiplayer Game Play.

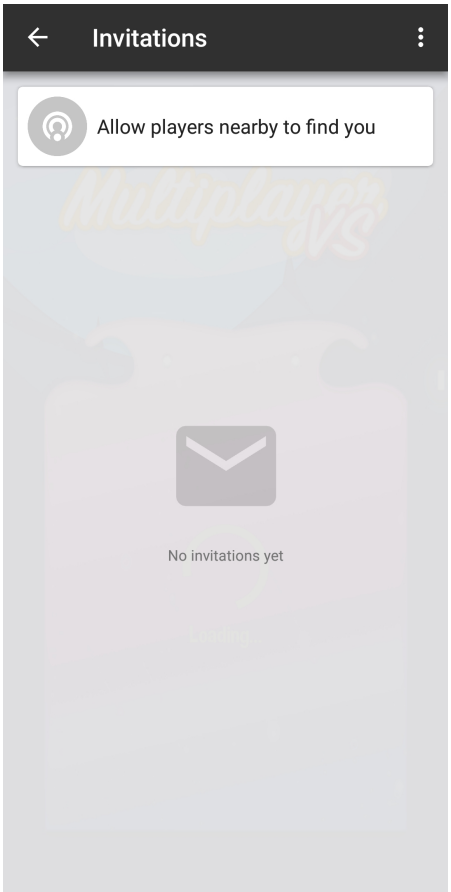


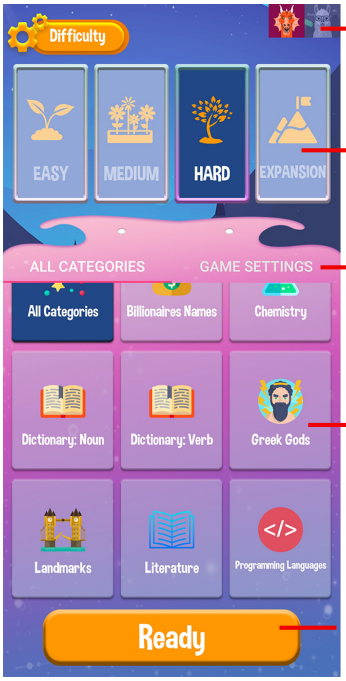



JessBunny is challenging you to a game!
ACCEPT
Multiplayer VS
Quick Play
Invite Friends
See Invitation
Sign Out


Word Multiplayer Game
Connecting
Me
JessBunny
Joined
Auto-pick
Auto-pick
Auto-pick
Auto-pick
Auto-pick


New match
2 to 8 players
Recently played
Add auto-pick
Nearby players: Off
JessBunny
General


Invitations
Allow players nearby to find you
No invitations yet



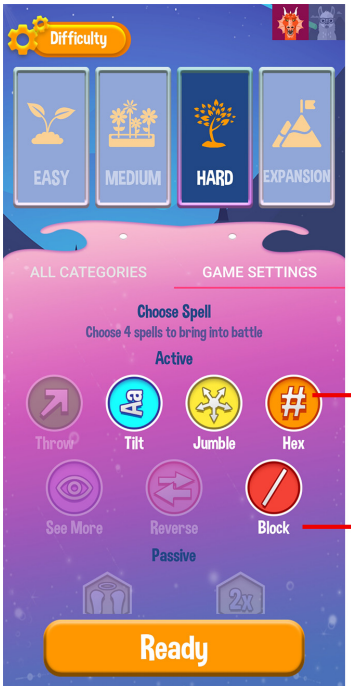
Player icon highlighted as player readies

When a player changes difficulty, all connected player's screen will sync immediately to show the new difficulty selected.

ViewPager swipe to toggle pages

Choose category. When a player changes a category, all connected players' screen will sync immediately, and show the new category

Click to Ready.. Game auto commence when all players are ready.

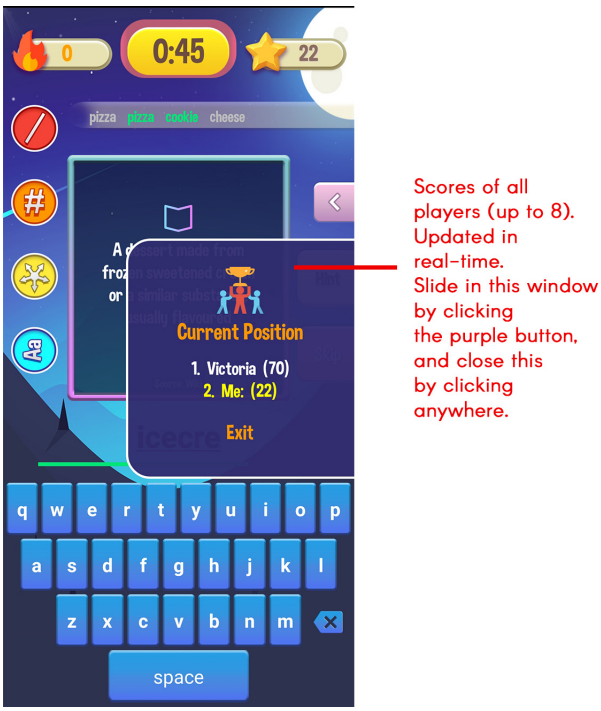


Clicking on the icon will select the spell. Clicking on a new spell will auto deselect the 1st spell, using a FIFO queue. Spells are not sync, i.e. you and your opponents will choose different spell to bring into battle.

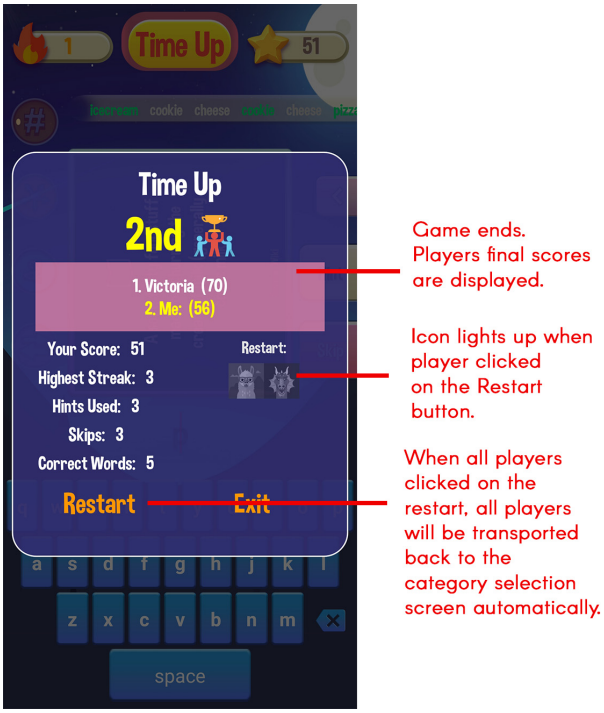
Clicking on the text will bring up a popup explaining the spell.

Spells. Hexagonal spells are passive (will activate automatically). Circular spells are active (need to manually click them).

Click to see current real-time ranking. Scores for all players are updated in real-time.



Scores of all players (up to 8). Updated in real-time. Slide in this window by clicking the purple button, and close this by clicking anywhere.

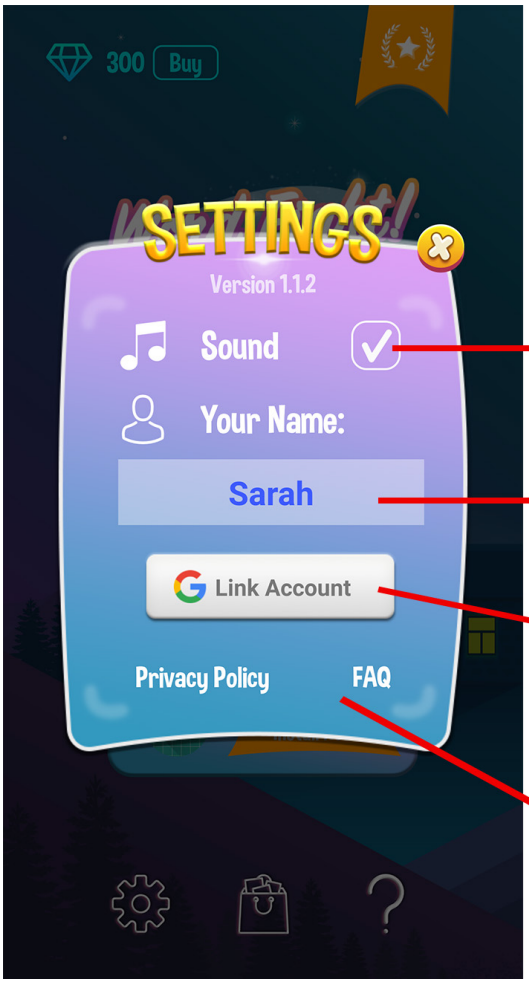


Game ends. Players final scores are displayed.

Icon lights up when player clicked on the Restart button.

When all players clicked on the restart, all players will be transported back to the category selection screen automatically.

5. Settings Page.



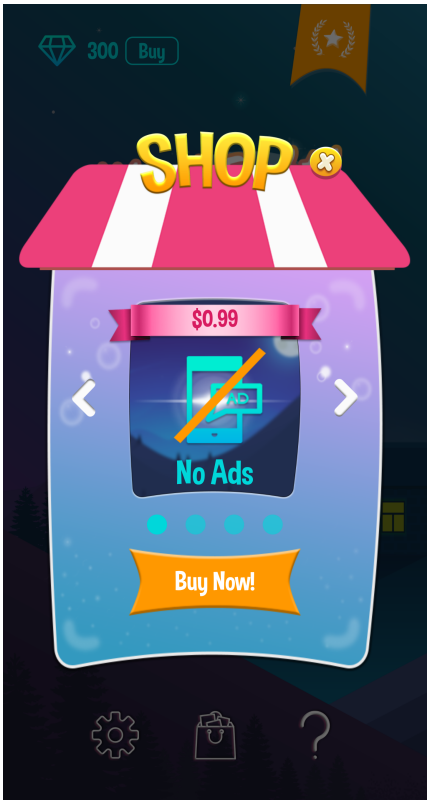
Tap to toggle sound. Store user's preference in SharedPreferences.

EditText to store user's name in SharedPreferences.

Link to a Google Account so that user can access his game state in any devices.

Access the Privacy Policy and FAQ page.

6. Shop Page.



Technical explanation for features developed as of Milestone 3:

1. In Category Selection page, player can select different difficulty level, of which the results are stored and fetched in SharedPreferences for subsequent game play. Category packs are loaded via RecylerView, with a custom layout that consists of an ImageView and TextView.
2. CountdownTimer is used to display the time remaining in the game. Upon pressing the pause button in 1 Player mode, CountdownTimer is cancelled and later restarted.
3. HorizontalScrollView is used to load history text in the game screen dynamically. For every question answered correctly / skipped, a green or gray TextView is created and added into the HorizontalScrollView.
4. Various custom PopupWindow and AlertDialog are used to display popups to users in Settings, Shop, Game Pause, Game End states.
5. Google Play Game Service Real Time Multiplayer is the main API used for Multiplayer mode. User signs in to his Google Play Game Service account, after which he can press Quick Play, Invite Friends, See Invitation, or Sign Out

button. Quick Play and Invite Friend supports up to 8 players in total. Invite Friends supports finding players nearby using Google's API.

6. Main multiplayer game plays are synced in real-time using Google's Reliable and Unreliable messages. Communication messages include selecting difficulty level, selecting category packs, on ready clicked, on spell activated, on score updated, on restart clicked.
7. State selectors are implemented for all game buttons so that users can see feedbacks when pressing on any buttons.
8. Sounds are implemented via SoundPool API for button clicks, and MediaPlayer is used for main background music. Upon muting the sound in the Settings, all sounds in the game will not play.
9. Facebook Audience Network is used to display native advertisements to the users. When Ad is not loaded successfully, the entire Ad's view will be set to Visibility.GONE so that user can have a better overall experience. As the app is not live yet, currently Facebook Audience Network ads could not be loaded. Instead, a custom dummy text and dummy imageview is shown.
10. In-app-purchase relies on Google Play Billing API. Currently, user can purchase 4 items: No ads, 50 Jewels, 200 Jewels, 500 Jewels.
11. All TextViews are loaded via a CustomTextView with custom fonts.
12. Fabric Crashlytics is used to log app crashes.
13. FirebaseAnalytics is used to log events so as to better understand users' engagements.
14. Players can resume his/her last game states by signing into his Google account in Settings page via Link Account. Consumable in app purchase (jewels purchases) cannot be restored, but non-consumable in app purchase (no ads version) can be restored by linking to the player's Google account.
15. App supports any screen dimensions, both tablets and phones, by ensuring that the UI component looks the same in terms of proportion throughout any screen sizes.
16. Restricting the game to only portrait mode as landscape mode is awkward to play.

Features to be developed from Milestone 3 to Splash down:

1. Integrate Google's Achievement API for players to unlock achievements and earn jewels.
2. Lock certain category packs which require users to unlock using Jewels.
3. Create a How-to-play screen that explains to users how the game works.
4. Add more animations, especially for spells used / affected.
5. Allow user to use default keyboard instead of custom in game keyboard

User Testing

Prior to milestone 3 submission, we have tested an early version of the game with our family and friends, and asked for their feedbacks.

User	**Feedbacks**	**Actions taken**
Family members	- App graphics are good - App is smooth with little to no lag - App is intuitive to use - Some questions in the game are too long. Consider shortening the texts. - Dictionary category is fun - Some categories (company, literature) are too hard to guess.	- Ensure that questions are capped at 25 words.

Friend A	- Consider providing default keyboard as some users may be more familiar with that. - Custom in game keyboard's position is a bit funny. Some touches feel awkward. - Overall the app is beautiful and nice.	- Adjusted the in-game keyboard sizes - Perhaps allowing users to have an option to use default keyboard, but that will mean accounting for the UI changes as some default keyboard may cover the game content now.
Friend B	- Consider a sign in option for user to save his game states. - Consider implementing leaderboard to showcase user's high score. - Have a help page with step by step guide to teach user the game mechanics.	- Integrated Google Saved Game API to allow user to resume his past game states, even on other devices. - If time permits, integrate Google Leaderboard to showcase user high score. - Help page will be done after milestone 3

Friend C	- Overall the game is quite fun - Only downside is that since there are no players now, quick play cannot find any players to play with in multiplayer mode.	- Maybe code out an AI to assume the role of a real player, matching with people looking to play multiplayer mode, as there are little players base when the app launches. This way, users will always get to play quick play mode.

Source control

Our team uses Git for source control since the start of developing this app. Whatsapp is the main communication channel for our team.

Problems encountered up to Milestone 3

Along the way, our team faces various technical challenges as well as time constraint due to other commitments. Nonetheless, we did well in overcoming the challenges and ultimately produce a working product.

Problems encountered	**Actions taken**
Unfamiliar with real time multiplayer concepts, device handshakes, securing player connections, etc. Basically, foreign to the entire concept of real time multiplayer.	Since we have little knowledge on real time multiplayer, it will not be wise to create our own architecture. Therefore, we employ Google Play Game Service Real Time Multiplayer API that does many of the behind-the-scene connections for us, as well as taking care of security issue. What we need is

	to learn the API and apply them to the project.
Getting the questions for the game. Since the main game play requires a huge amount of question database, we need to come up with a system to get these data efficiently.	We wrote scripts to scrap Wiktionary and Wikipedia for texts to feed the question. Thereafter, a simple Java programme transforms the scrapped text into a java file to be called in the app. Care has to be taken to take only texts from websites that are free to use commercially.
Various bugs along the way, NullPointerException, unexpected behaviours etc.	Source stackoverflow for answers. Read official API. Log game states to console and try to figure out what went wrong. Take some coffee and fixed those nasty bugs.
Limited time. Since our team started the project quite late (as we pivoted from our initial idea in milestone 2), we have limited time on our hands.	Prioritize the important core features, and implement them. Come up with a time management sheets and delegate tasks efficiently. Take coffee and code on.
Unfamiliar with certain api, such as Google sign in, Google Play Billing etc.	Read documentation and source Google for tutorial. Stackoverflow is more useful than asking friends for help.

Security Practises

As we have little knowledge on networking, it is not wise to create the game network architecture ourselves. Hence, we employed Google Play Game Services for both Real Time Multiplayer and Saved Game features. This allows us to push the security responsibilities to Google, who has far more resources and expertise to secure the connections. We do not collect any user information, nor payment details. In-app-purchase are done via Google Play Billing API. We only collect anonymous event logs for data analytics.

App optimization

To ensure user gets the best experience possible when using our app, we did the following refinements to fine tune the app:

1. Compressing images and audio files. This ensures that the apk size is kept to minimum. Nobody likes to download a large game. Also, a game that takes up lesser storage space has a lower chance of user uninstalling the app. Care has to be taken to ensure a fair tradeoff between quality and file size. PNG-8 is used as far as possible for images with little colors. Image dimensions are kept to where appropriate, as this also helps in app memory usage.
2. Creating graphics for the game. To ensure a consistent feel and look of the game, most graphics are created by ourselves. This also allows us to be more flexible in representing the in-game graphics.
3. Obfuscate code. At the very least, proguard is used to obfuscate the code and make it harder for one to read the decompiled source file. At the same time, proguard also reduces the apk file size. Care has to be taken to add proguard rules appropriately so as not to break the app.
4. Support for all screen sizes, including tablets and phones. To ensure that players with different screen dimensions have a pleasant experience using our app, we make sure that all screen sizes will look uniform in terms of proportions. To accomplish this, we make use of different dimens.xml for different screen sizes to represent UI dimensions.
5. Memory management. Memory leak are observed via LeakCanary library. We do not create unnecessary objects and ensure that all objects are properly garbage collected at the end of their lifecycle. We monitor the memory usage via Android Profiler in Android Studio, and make sure that the app does not crash due to OutOfMemoryError. Images sizes are downscaled to the appropriate sizes to reduce memory footprint.

Project Log

Date	Task	Venue	Time Spent	Done by
27 May 2018	Ideation, throw out ideas, eliminated and choose what we feel is interesting to us and doable.	NUS Computing	5hrs	Together
28 May 2018	Look through the Android playstore to come up with features for our game, how to make it interesting and unique so as to stand out from the market	NUS Computing	5hrs	Together
29 May 2018	Discussed what to be included as a minimum viable product and timeline to achieve the MVP. Discussed what additional features we can implement if we are able to finish ahead of time	Skype	3hrs	Together
30 May 2018	Decided on doing an Eraser Flipping game on Unity for mobile. Read up and watch videos / tutorials on Unity online	Home	3hrs	Gan Chin Yao Lim Wai Lun
1 June 2018	Continue learning about Unity online	Home	5hrs	Gan Chin Yao Lim Wai Lun
2 - 9 June 2018	Playing around the Unity software and learning how to use it from scratch. Try out some projects found online. Understood the basics of Unity. Created simple unity project to test run the platform.	Home	50hrs	Gan Chin Yao Lim Wai Lun
10 - 16 June 2018	Come up with initial UI design for the Eraser game in Photoshop	Home	10hrs	Gan Chin Yao
25 June 2018	Decided to ditch the Eraser game and go for another idea. Brainstorm what other ideas we should do	NUS Computing	3hrs	Together
26 June 2018	Ideation pivoted to creating a Word-based Multiplayer game on Android platform. Idea creation for how the game works.	NUS Computing	3hrs	Together

27 June 2018	Initial mockup UI design for the Word game is done on Photoshop	Home	5hrs	Gan Chin Yao
27 June 2018	Came up with the point system, award system, achievements, types of pack for the game	Home	5hrs	Lim Wai Lun
28 - 30 June 2018	Created the following xml file in Android Studio: Main Screen, Popup screen, 1 Player Screen. Coded the following in Java: RecyclerViewAdapter, ViewPager Fragments, TabLayout	Home	30hrs	Gan Chin Yao
28 – 30 June 2018	Coded the following in Java: Created a data structure to store our words, created custom in-game-keyboard layout, binding EditText to custom in-game-keyboard, writing logic for the point system	Home	30hrs	Lim Wai Lun
1 – 2 July 2018	Redesigned the UI of the app. Created mockup of skins in Photoshop. Compressed image resources and port them to the actual app. Created various assets for different screen sizes.	Home	20hrs	Gan Chin Yao
1 – 2 July 2018	Scrapping the internet for dictionary meaning that will be used in the app. Came up with packs ideas. Grouped words according to the 4 level difficulties: easy, medium, hard, insane.	Home	20hrs	Lim Wai Lun
5 – 8 July 2018	Created different resources for state_selected, state_pressed, and default for the following: 1playerbutton, multiplayerbutton, buybutton, settingsbutton, shopbutton, achievementbutton, helpbutton, in-game-keyboard, hintbutton, skipbutton, multiplayersigninbutton, multiplayerinvitefriendsbutton, multiplayerseeinvitationbutton. Wrote code to implement these different state selections, and persistent state selections for level difficulty and pack selected	Home	10hrs	Gan Chin Yao

5 – 8 July 2018	Integrated SoundPool and MediaPlayer into the app to play sound on click events. wrote the following Java codes: 1. Playing of sounds on button clicked 2. RecyclerView in ViewPager under category selection 3. SharedPreferences for default level difficulty, category packs	Home	10hrs	Lim Wai Lun
14 July 2018	Learning Google Play Game Services Real Time Multiplayer api, and testing them out. Design icons for spells and their clicked states:	Home	10hrs	Gan Chin Yao
14 July 2018	Wrote java code for game statistics, pause popup screen, end game popup screen, history horizontalScrollView	Home	10hrs	Lim Wai Lun
18 – 20 July 2018	Setting up Google Play Game Services for real-time multiplayer: authorizing app, sha1, linking app, submitting info for game details, adding tester account Multiplayer communications: sending messages, receiving messages, on player ready, on spell activation, on player scores update, on game ends, on player restarts Integrate game logic to multiplayer connections.	Home	30hrs	Gan Chin Yao
18 – 20 July 2018	Coded out logic for game settings: sound selector, name selector, Google link account. Learnt and coded out logic for in-app-purchase	Home	30hrs	Lim Wai Lun
21 – 22 July 2018	Created the following ui and graphics: multiplayer screen, multiplayer versus logo, multiplayer popup graphics, home screen logo Created selectors for all 9 spells. Settings screen: prepare image resources in photoshop, code xml ui, sharedpreference for sound checkbox, player name edittext, keyboard	Home	15hrs	Gan Chin Yao

	open/close listener, privacy policy text screen, faq text screen.			
21 – 22 July 2018	Wrote code for multiplayer ranking slider and algorithm to switch rankings, text, and text color. Fixed some bugs regarding NullPointerException and unexpected behaviours. Added animations for game start dialog, score and streak textview.	Home	15hrs	Lim Wai Lun
25 – 26 July 2018	Optimized App. Fine-tuned some parts. Fixed some bugs. Changed some UI proportions. Prepare for milestone 3 submissions, poster, video.	NUS Computing	13hrs	Together

Total time spent:

Gan Chin Yao: 220 hrs

Lim Wai Lun: 220 hrs

Miscellaneous

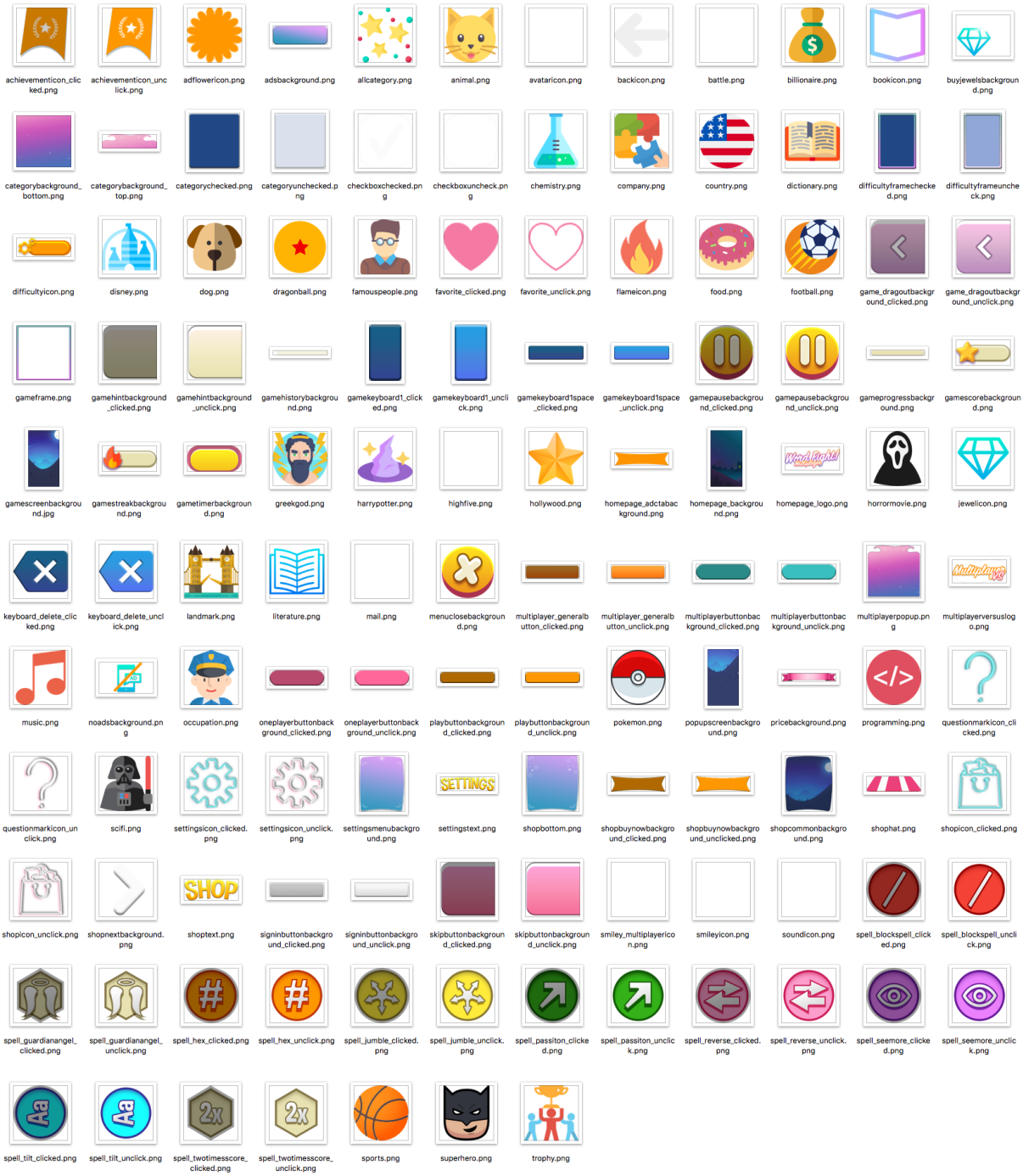
As of Milestone 3:

Total lines of JAVA code written (include comments, exclude libraries used): 6098

Total lines of XML code written: 5102

Totals lines of JAVA + XML: 11200

Graphics created:



<End>