Date	Description	Classes	Methods
Aug 7, 2022	Implemented parts of the game model	**GameModel –** represents the game’s logic. **Player –** represents each player which includes information associated with each player. **Tile –** represents each tile players will use. **MainActivity** – implements the view of the game.	**initializeTiles** – initializes the “tiles” arrayList which represents the total number of tiles players will draw from. **initializePlayers** – initializes the “players” arrayList which represents the total number of players that will be playing the game. **initialDraw** – performs the initial draw of each player i.e inserting tiles in the players’ arrayList. **initialPlayer** – determines the initial player depending on the result of **initialDraw** i.e the player with the most playable tiles is the initial player. **getPlayerHighestCCount** – helper method to the **initialPlayer** method that determines the player with the highest number of tiles with the same colour. **getPlayerHighestSCount** - helper method to the **initialPlayer** method that determines the player with the highest number of tiles with the same shape. **draw –** draws tiles from the tiles arrayList into each player’s arrayList. **play –** plays a particular player’s move after determining whether the move is legal. **turn –** changes the current player after a player makes a valid move. **legal –** determines whether a player’s move is valid. **equivalent** – determines if a tile a player is about to play is equivalent in the sense of the game i.e the tile has the necessary characteristics, such as the sequence of equivalent characteristics (for instance colour), for a play to be valid. **nul –** determines if the 2D board has no tile in that position.

Aug 8, 2022	Fixed *part* of the bug which incorrectly determined the legality of a player's move.	GameModel – modified the game model to determine parts of the code responsible.	**play –** modified to remove the tile that a player has played. **nul -** included a condition to check if a position is within the bounds of the 2D board of the game. **duplicate –** implemented method to check if the current tile to be played is the same in both types of characteristics with the surrounding tiles.
Aug 15, 2022	Fixed *part* of the bug which incorrectly determined the legality of a player's move.	GameModel – modified the game model to determine parts of the code responsible.	**draw –** modified to add back a player's removed tile back into the main tiles arrayList. **play –** modified to added missing conditions to determine the legality of a player's move e.g prevent inserting a tile where there's already a tile present. **copy -** implemented method to deep copy the 2D board array so that a backup can be made if a player decided to not continue with their move i.e to draw mid play. **backup** & **recover** – backups and recovers various fields in case a player decided to not continue with their move. **withinBounds –** implemented helper function to determine whether position is within bounds of the 2D board
Aug 16, 2022	Fixed the bug which incorrectly determined the legality of a player's move.	GameModel – made major changes to the game model to fix the issue.	**duplicate –** completely changed the method to make a new implementation of the method to fix the bug. **next** – changed the name of **duplicate** to **next.** **adjcEquivalent** – implemented method to check if the current tile to be inserted is legal relative to the already existing sequence of tiles.
Sep 08, 2022	Added/ modified functionality to the game model and parts of the user interface	GameModel – added additional methods. **MainActivity** – added parts of the UI. ImageAdapter – adapter that connects the arraylist of a	**calculate –** implemented method to calculate a player's points. **paths –** helper method to determine whether the current tile of the **calculate** method has paths in a different direction to the direction in **calculate**. For instance, if in **calculate** we are moving in the

		player's tiles and the recyclerview.	horizontal or x direction than **paths** determines if the current tile has alternative paths in the vertical direction. **orientation** – implemented helper method to determine the orientation of a player's plays. In other words, whether they have placed their tiles horizontally or vertically. **nullTile** – implemented helper method to determine a tile of a player's play that is at the very end in the sequence. (Allows for counting from the null tile in the direction determined by **orientation**) **assignPoints** – uses the helper methods to assign points to the current player. **getBagCount** – implemented method to determine total number of tiles in the bag. **setupRecyclerView** – implemented method to connect the arraylist of a player's tiles and the recyclerview. **resetWidthExcept** – implemented helper method to enlarge a tile represented by an image when a player clicks on it. **setupBagCount** – implemented method to display the total number of tiles in the bag on the user interface.
Sep 09, 2022	Added/ modified functionality to the game model and parts of the user interface	GameModel – added/ modified methods. **MainActivity** – added parts of the UI. **StatusAdapter** – adapter that connects the gridview to each player's status such as scores.	**draw** – adapted the method so that it works with the UI. **play** – removed some redundant code. **setupPlayersStatus**– implemented method that sets up the **StatusAdapter**. **setupCurrentPlayer –** implemented method that sets up the current player so that it reflects on the UI when the current player changes. **setupGridLayout -** implemented method that sets up the UI board that players will interact with when placing their tiles. **updateTags** – implemented method that updates the tags embedded within

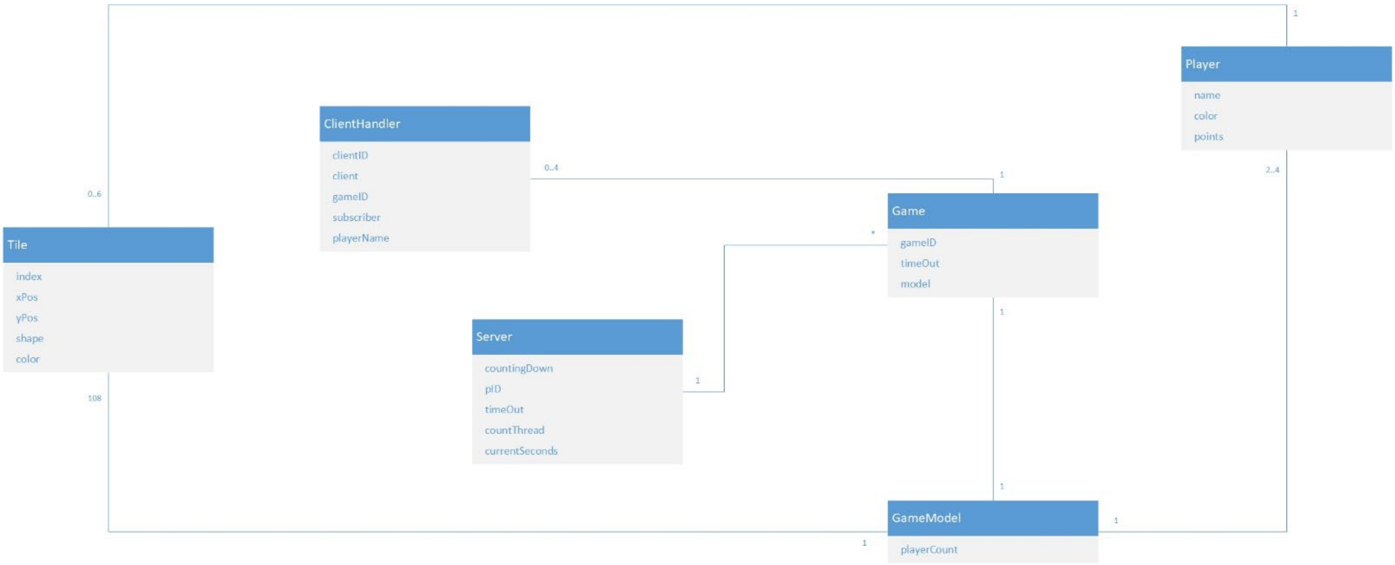
			imageviews that represents each player’s tile so that if a player for instance selects a tile to place on the board, the index of the remaining tiles is updated. **onTileClicked** – implemented method that is called whenever the user selects the board’s cell such as when placing their tile. **getDrawable** – implemented method that returns a drawable when given a string. **setupRecyclerView** – modified method to add functionality such as long pressing a tile activates the “multi select feature”. **onPlay –** implemented method that is called when a player presses the play button. **onDraw –** implemented method that is called when a player presses the draw button when drawing tiles from the bag. **resetMultiSelect** – implemented helper method that resets variables of the multi select feature. **vibrate –** implemented helper method for the multi select feature which vibrates a player’s device when the feature is activated. **updatePlayerTiles –** updates the UI when something about the tiles views changes such as when the current player changes which will change the images used to represent the tiles.
Sep 09, 2022	Added/modified functionality to the game model and parts of the user interface	GameModel – added/modified methods. **MainActivity** – added parts of the UI.	**allSidesNull** – implemented method to check if a tile placed by a player has any tiles on either side of it if there already are any tiles. **assignPoints** – completed the method to assign points to current player. **calculate** – reimplemented the method to fix previous bug. **isQwirkle** – implemented method to determine if there is a qwirkle.

			undoTiles – implemented method to undo a player’s move when drawing when already placed a tile(s) on the UI. **isBonus** – implemented method to determine the last player to assign bonus points. **updatePlayerScore** – implemented method to update player’s score in adapter array. **setOnDraw** - fixed a bug that brough issues regarding syncing the UI with the model.
Sep 30, 2022	Reimplemented using fragments. One fragment responsible for the game and another for messaging among players.	**GameFragment –** Responsible for the gaming functionality. **MessagesFragment** – Responsible for the messaging functionality. **PagerAdapter** – An adapter for the ViewPager2 to connect the **GameFragment** and **MessagesFragment**. **PlayerMessage** – Represents each message sent by a player. **PlayerMessageAdapter** – An adapter for the RecyclerView in the **MessagesFragment** that list all messages sent by players. **StartActivity –** An activity shown pre-game to capture and send the number of players specified by user to the **MainActivity**.	**setupListeners** – Implemented method to attach listener for the **EditText** to enable or disable a button dependeing on whether a user a entered a value. The method also attaches lister for the send button. **setupRecycleView** – implemented method to setup messages **RecyclerView.**
Oct 01, 2022	Made some minor improvements, moved hardcoded strings to the string resource and fixed some issues due to serialization which caused the app to crash.		

Oct 01, 2022	Added activity to congratulate winner.	**EndActivity** – shows congratulatory message to winning player **GameFragment** – added method. **GameModel** – added method.	**gameFinished –** starts **EndActivity**. **getWinner –** gets player with the most points.
Oct 22, 2022	Implemented network component of the app. **NB:** uses PubSubBroker pattern.	**Message –** message objects defined on both the server and client and are executed when read accordingly. **Begin** – message subscribed by all clients. Published when the game is ready to start with available players. **Name –** message subscribed by all clients. Published when a player connects to server to give them their display name. IMessage - message subscribed by all clients. Published when a player sends a post a message. Waiting - message subscribed by all clients. Published when players must wait for game to start. Stop - message subscribed by all clients. Published when something about one of the clients causes them to no longer participate in the game. Played - message subscribed by all clients. Published when the current player has confirmed a move. Drawn - message subscribed by all clients. Published when the current players draw tiles from the bag. Client: ServerHandler – handles server related messages.	**Game.add() –** adds client to list and sets common subscription messages such as **Begin**. **Game.**begin() – publishes **Begin** message when the game is ready. **Game.**remove() – removes a client when for instance **Stop** message is received. ***.**runLater – used by message's apply method to run code when the fragments are in view by starting a thread defined in each fragment. **GameFragment.*focusOnView –** scrolls in view of the tiles placed by a player. **GameModel. updatePlaterTiles –** called in the **Message** objects to update tiles of other players when one of the players replenish/draws their tiles. **GameModel.**isTurn() – determines whether it's the clients turn to play.

		Server: ClientHandler – handles client related messages. Game – represents a game among each group of players.	
Oct 26, 2022	Implemented undo functionality, some animations, notification service for player messages	**Notication –** used to show a notification when a player sends a message.	**GameFragment. setOnUndo–** undoes a player's tile placement. **GameFragment.qwirkleAnimate –** animates the board background and text to show when a player performs a qwirkle. **GameFragment.easeInTilePlacement –** eases in the tile opacity after the player makes a playing move.
Oct 31, 2022	Included two fragments in BeginActivity - GameHistoryFragment and BeginFragment. Added additional animations. Implemented in-game walkthroughs. Made the size of UI objects – tiles, scoreboard etc - to be dependent on the screen size.	**GameHistoryFragment –** displays a player's previously played games. BeginFragment – works as the previous BeginActivity. XMLHandler – include methods write and load a player's previously played games in xml format. Game – stores information about a player's previously played game. GameHistoryAdapter – an adapter for the recyclerview used by GameHistoryFragment. Helper – has functionality used in multiple places within the application such as animations and sound. Helper.AnimateTilePlacement – responsible for animating the placement of tiles on the board.	
Nov 02, 2022	Reimplemented both server and client for game re-join feature i.e., a player can re-join a game if it exists, and they have not explicitly forfeited	**Server:** **GamesHandler –** took everything from the Server class to GamesHandler. The server only accepts connections; each time the client app is opened, it is checked whether they have	

		implicitly forfeited the game such as when losing internet connectivity. The GamesHandler is responsible for only when the client joins for the first time. CountdownThread – counts down from a set timeout and at the end starts the game unless interrupted such as when a player leaves a game in the middle of the countdown.	


1
Player
name
color
points
ClientHandler
clientID
client
gameID
subscriber
playerName
0..4
1
2..4
0..6
Game
gameID
timeOut
model
*
Tile
index
xPos
yPos
shape
color
1
Server
countingDown
pID
timeOut
countThread
currentSeconds
1
108
1
GameModel
playerCount
1
1