|  |  |  |  |
| --- | --- | --- | --- |
| **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 – implementes the view of the game. | **initializeTiles** – initializes the “tiles” arrayList which represents the total number f 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 bugs caused by 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 bugs caused by 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 bugs caused by which incorrectly determined the legality of a player’s move. | GameModel – modified the game model to determine parts of the code responsible. | **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. |