# Log:

## October 26th:

* Prepared the repo with file structure, updating .gitignore (.1 hours)
* Ensured android studio works, emulator runs, and can deploy to device with example applications (1 hour)

## October 28th:

* Did android studios tutorial on how create an app and prepared environment to start work on the application (1.5 hours)
* Added a new activity for the coin flip
  + Design time (.2 hours)
  + Implementation time (1 hour)
* Designed a failed data structure to pass data (.5 hours)
* Added Skeleton for the cards ( 1 hour)
  + Card Model object
  + Card View
  + Card Enums for Suits

## October 29th:

* Previous activities end once new ones start to prevent user from going back (.5 hour)
* Buttons gray out after being clicked on coin toss (.5 hours)
* Added basic card functions, this includes values, constructors, and getters and setters (1 hour)
* Deck created and drawing card is implemented (.5 hours)
* Created skeleton for the hands, which include the underlying data structure, getters and setters, removal and adding cards. (1 hour)
* Created ability to draw four cards to a hand (.1 hour)
* Added views for all the created classes, and have them initialize based off the current state of the model (2 hours)

## October 30th:

* Created image buttons and displayed a test card (4 hours)
  + Many failed attempts were included in this
* Added images for all cards (.5 hours)
* Worked on xml for the human hand (.5 hours)
* Locked game orientation to portrait (.2 hours)
* Created help functions in view, allowing for all cards to be displayed to the image buttons ( 1 hour)

## October 31st:

* HandView is capable of drawing to all buttons, but not correct in all cases (1 hour)
* Improved initialization order to guarantee correct views are loaded before displaying (1 hour)
* Added PlayerMoves and PlayerID enum classes (.2 hours)
* Added skeleton for the player abstract class (1.5 hour)

## November 1st:

* Added computer hand to the XML file (.5 hour)
* Able to display the computer cards to the hand (.1 hour)
* Made the background table green (.1 hour)
* Round can initialize itself (.1 hour)
* Created PlayerView skeleton (.5 hour)
* Created basic player moves which always trails ( .5 hours)
* Converted the controller from using test variables to using the actual data vars (2 hours)
  + This needed more accessors and some code restructuring
* Cards start face-up (.1 hour)
* Fixed debug code to display cardBacks instead of testCards(.1 hours)

## November 2nd:

* Able to display N cards on the table section (1.5 hours)
* Change crop type of the cards so they display card with less whitespace (.2 hours)
* Added helper functions in controller to change color and set up pixels in density (.5 hours)
* Able to select and deselect cards (Graphically and in the model) (1.5 hours)
* Added vectors of ids for buttons declared programmatically so they can be accessed in loops (.2 hours)
* HandViews and hands can limit selection to one card at a time if needed (.2 hours)

## November 3rd:

* Able to set the submit and confirm buttons to correct state (1 hour)
* Able to correctly use string localization on the button (.2 hour)
* Able to trail cards graphically (3.5 hours)
* Updates cards in [graphic] hand after trailing (1 hour)

## November 4th

* Changes clickable of player hands and tables so the only clickable hands are the ones that are valid to be clicked at that point in time (1 hour)
* Computer trails correctly appear graphically (.1 hour)
* Program will now always correct unselect card after it is played (.5 hours)

## November 5th

* Hands and tables can now have their clickabilty toggled to match what is able to be clicked based on the current turn (.5 hours)
* Computer Trails update graphically (.2 hours)
* Human can now unselect cards they were tapped in their hand and reset the menu button (.5 hours)

## November 6th:

* Worked on manual and code formatting for mile stone submission (1 hour)
* Controller can now check for when the round ends (.2 hours)
* Internal logic updated so true means the same thing for all things in the main call stack (Contoller -> model) (.5 hours)
* Program is able to find all matching cards on the table, based on selected card (.5 hour)
* Table is able to clear all selected cards automatically, and forcibly select all cards which need to be captured (1 hour)
* Required cards can no longer be unselected (.2 hours)
* Table releases selected cards when the user is no longer required to capture them (.3 hours)

## November 7th:

* Fixed player view incorrectly initializing the pile observer (.5 hours to debug, .1 to fix)
* Rewrote Hand and HandView to use observers instead of being told when to update through polling / controller (3 hours)
* Hands properly display new cards in all cases when new cards are dealt from the deck (.5 hours)
* Created the Action Log which logs all moves that happen in the game, and displays thems in a separate activity, and as a button on the main screen (1 hour)
* Added more helper functions to Action Log to make it easier to integrate into the program (.2 hour)
* Added the correct hook-ups through-out the program to log what moves were taken (.5 hour)
* Controller can now removed buttons from the table once they are captured (.7 hours)
* Human is able to capture cards, with no rule validation (.5 hours)
* Made sure the removed card indices in Playermove is always sorted in descending order (.1 hour)
* Created the Human Pile in XML in preparation for captures being finished. (.5 hour)

## November 8th:

* PlayerMove can be told to mark itself invalid (.1 hour)
* Optimized the XML to make it more cleaner and so prevent it from having unintended complications whenever small changes are made (.5 hours)

## November 9th:

* User interface now properly updates when a move is rejected by the model (1 hour)
* Added radio buttons to the XML and hooked them up so the human can choose which move they want to make (1.2 hours)
* Fixed a bug in human logic where it would drop cards that it selected in data structure copying (.2 hours)
* Added accessors to get the piles from the views to allow them to be drawn (.1 hour)
* Captured cards are now added to the human pile graphically (.75 hours)
* Added a RoundView class with the round to maintain properly Model View Controller semantics (1.5 hours)
* Added a Computer Pile and the deck to the XML in preparation for them being added (.2 hours)

## November 10th:

* The deck is displayed on the main game screen (.5 hour)
* Tracked down a null pointer exception which only happened when the Computer goes first (.5 hours to find, .1 to fix)
* At end of rounds, cards go to the pile of who captured last (not graphically) (.5 hours)
* Cards now render graphically for both players (.5 hours)
* Radio buttons now disable trailing as an option when a player must capture cards on the table (.2 hours)
* Worked on how to display a Build on the table (1.5 hours prototyping different designs)
* Builds can now be created in the model (1 hour)

## November 11th:

* Builds can draw to the screen (.2 hours)
* Updated many functions in Controllers and views which were needlessly specific in the datatypes they accepted which prevented the Builds from displaying polymorphically. (2 hours)
* Cards in a captured build get added to the pile, however build graphic remains on the table (1.2 hours)
* Builds format correctly in the log (.1 hour)
* Builds act like normal image buttons through program logic, but not through user interaction ( 1 hour)

## November 12th:

* Builds can be selected or deselected like normal cards (.2 hours)
* Able to get to the load file screen (1 hour)
  + This includes design time for the screen
* Failed to implement file loading (3 hours)
  + This was do to a failure in understanding documentation and how android deals with file permissions
* Required cards unselect when not on the capture actions (.2 hours)
* Computer is now able (and required) to capture identical symbols on the table(.5 hours)
* Started to implement reserved values, which are used to track how many and what values are being used in a build (1 hour)

## November 13th:

* Prevent cards being selected in a duplicated manner for captures (this is due to a required capture stacking on top of a use selected capture) (.5 hours)
* Load file is able to find and display all save files in the directory (1 hour)
* Added the turn menu, which is able to quit the game (1 hour)
* Computer is now able to offer help to the player (1.5 hours)
* Ask for Help button is only visible while the human is making their move (.5 hour)
* Able to show the current player graphically (.2 hour)
* Added a switch which allows the turn menu to be accessed when the computer is the current player (.2 hour)
* PlayerSaveData class was added to store all player save in a single object (.25 hours)
* Added the framework to write data to a save file (.75 hours)
* Increased minimum api version to 26 from 21 (.1 hour)
* Fully able to write (what data is available) to a save file (1.5 hours)

November 14th:

* All new activities are now locked to portrait as the others are (.1 hours)
* Fixed bug where the application would close before finished writing data – by not closing the app. (.5 hour)
* Able to open a file and read in all its data (.5 hour)
* Able to remove blank lines (.1 hours)
* Able to clear headers from lines (.2 hours)

## November 15th:

* Human is able to capture sets (.5 hours)
* Fixed starting logic for the computer playing first on a save (.2 hour)
* Able to fully load in a file minus for build owners (1 hour)
* Able to now move to the Game Screen after reading in a file (.2 hour)
* Added more getter and setters to serializer to clean up code (.1 hour)
* Core Gameloop can now tell itself if its loading in a file or starting a new game (.5 hours)
* Able to create a card from a string to load aide in Serialization (.5 hour)
* Able to load in a deck from a string (.2 hour)
* Able to load in a save file in the round, which passes the data to the rest of the model (.5 hour)
* All other model files can parse their save file data (1 hour)
* Restructured model so builds and multi builds inherit from an abstract class BuildType (.5 hour)
* Fixed logic error in the remove blank line function which accidentally marked lines containing only whitespaces as non-blank (.5 hour)
* Overloaded toString in multi-builds (.1 hour)
* Added a new constructor in Hand for the table save which can parse builds (1 hour)
* Went over multiple functions to use build types instead of builds so both can be used (.5 hours)
* Added draw call on load for piles for save games (.1 hours)
* Multi builds display to the screen correctly (.3 hours)
* Added helper counting functions in preparation for scoring (.5 hours)
* Added round over logic to detect when round is over in Controller(.1 hour)
* Added skeleton to the tournament class for scoring and round continuation (.5 hours)
* Designed and added skeleton for the Score screen (1 hour)

## November 16th:

* Piles display on score screen (.5 hours)
* Added Build Strings to build buttons for readability (.1 hour)
* Computer move checkbox starts checked (.1 hour)
* Added move reason to the computer moves (.5 hours)
* Ai can capture sets up to size 3 (1.5 hour)
* Updated parsed lines in file loading to allow valid blank lines – such as empty piles (.75 hours)
* Table no longer makes all builds a multi build on loading in a file (.5 hour)

## November 17th:

* Fixed a bug which caused a desync between the controller, view, and model when a build was extended (4.5 hours to debug and fix)
  + The above fix also had a rewrite of the controller button logic to be more consistent, which was another (1 hour)
* Fixed cards being falsely added to a pile when multi builds were made (.5 hour)
* Fixed computer buttons being clickable in save files (.1 hour)