**State of game when it was forked from master** [**ygoduelistharry**](https://github.com/ygoduelistharry)**/**[**7-wonders-duel**](https://github.com/ygoduelistharry/7-wonders-duel)

* Included age card layout of game
* Included list of all playable cards with type and cost
* Included .py file which contains a basic version of the game and can be run in command line with player input switching between 2 players
* When playing the game the layout of the current age is displayed with available cards colour coded and not available cards hidden
* A picture containing timeline

  Description automatically generated
* Players can construct cards which are added to their board, discard cards for coins, or quit the game
* Cards which are constructed are moved to the specific player board
* When the last card of the last age is selected, the game ends but no winner is selected or displayed
* Includes 6 Class functions which make up the main part of the game
* Class Game:
  + Initially defines a single instance of the game and requests input from player
  + Main game loop - Function to select card on board and perform the appropriate action
    - Get player, opponent and age variables
    - Prompts the player to select a card
    - Prints whether card has already been chosen, whether it is covered, whether resources are missing, or whether the input is not a valid action
    - Checks whether no cards are left -> progress age
    - Otherwise updates board with input from player and changes player turn
    - Ends the game when the last age is over
    - Requests new input after input was accepted
    - Created empty functions for whether a card is constructable, and which moves are valid
    - Displays the game state i.e. current player turn, current state and cards on the board, the city/board of each player
* Class Card:
  + Define a single card. Attributes match the .csv headers
  + Sets the variables for the card which are filled from the CSV file later on
* Class CardSlot:
  + Define a card slot on board to represent selectability, visibility, etc.
  + Display the Card back to the player as either Hidden or the card based on game state and whether the card is covered or not
* Class Player:
  + Define a class for play to track tableau cards, money, etc.
  + Creates and sets initial variable for players such as coins, victory\_points, clay, etc.
  + Creates a function to construct cards (empty so far)
* Class StateVariables:
  + Randomly selected the first player if none specified
  + Starts at age 0 and military track 0
  + Changes current player turn
  + Changes the age when all cards are used up
* Class Age:
  + Reads the age layout and card list CSV’s
  + Takes dataframe of all cards and creates list of card objects representing the board for a given age
  + Updates a slot when a card is selected
  + Updates the whole board when an age is over
  + Prints visual representation of cards remaining on the board for this age

**Additional features needed**

* Select card differently?
* card\_constructable function to check whether card is constructable given state and cost
* construct\_card function to pay resources, add card to board, gain benefit
* update function to update players clay, wood, etc. when card is added
* valid\_moves function returns list of valid moves for current player
* Create military board
* Create science board
* Account for military victory
* Account for scientific victory
* Create Wonders cards -> draft, usage, limitations
* If go again wonders is chosen -> turn to same player
* When game end -> Count victory points and display winner
* Create interface for AI -> no need for player input
* Create a playable interface

**Rules to keep in mind**

* Buying a resource costs 2 coins plus the number of resources your opponent has of that type -> you only get that resource for 1 turn, there is no limit in amount of resources to buy, thus separate action for buying needed (not nested in constructing a card)
  + If you have a yellow training card with a resource and 1 coin next to it, then buying that resource only costs 1 coin no matter how many resources of that type you opponent produces (see below)
  + 
  + If multiple resources are listed as options, then the player can choose which of the resources to produce at each turn
* Constructing a card which has a symbol as prerequisite and you own the card with the respective symbol allows you to build that card for free -> (no material/coin costs)
* Military points -> for each military point move the conflict one step in the direction of the opponent (if a dotted line is crossed -> apply the effect now (lose coins) and remove the military token)
  + If the conflict pawn is moved all the way to the opponents side -> you immediately win
  + Player with weakest military chooses who begins the new age (if its in the centre, the player who played a card last will play first)
* Science -> any time you build a science structure that provides you with an identical pair of symbols, choose one of the progress tokens from the gameboard to keep (unique benefits)
  + If you have ownership of six different science symbols, you immediately win the game
* Wonders (8 wonders randomly selected) -> at the beginning of the game 4 wonders are placed in the middle, a random player starts and chooses 1 wonder, the next player chooses 2 and the first player chooses 1 again, for the next four player 2 starts in the same fashion
  + Only 7 wonders can be build -> the eighth wonder is discarded
  + To build a card you can place it under the wonder and pay the cost of the wonder instead of the cost of the card
* Victory points
  + Victory points awarded by Guild cards for each e.g. red card are awarded and counted up at the end of the game -> count towards all red cards that the player has at the end of the game not at the current time when it is played (one victory point for each red card in the city with the most red cards)

**Additional features added**

* Replaced pandas dataframe with numpy arrays
* Configured card\_constructable function to only allow constructing cards when enough money or resources are available
* Configured construct\_card to decrease coins when card is constructed
* Configured interface using s + card number to display an image of the card
* Configured interface to display whole rows with “s” -> Switch between them with W and S
* Added card effects for all cards except Yellow, Green, Purple
* Added indicator if card is selectable in GUI