EECS1720 - A1 - Card Game

Game Name: *Battle of Fate*

This project is an original **Card Battle Game** for **2** players(Human vs Human or Human vs AI).

Each player in the match has an initial 20 **life points(HP)**. Players **summon** cards to the **battlefield** to attack and defend after **drawing** cards from the **card pool** into their **handcards**. When the HP of one Player ≤ 0, then got defeated.

Each player has 3 **action points(AP)** in the first turn and thereafter each turn has 2 **initial action points**. **Drawing** card from Card pool, **summon** and **attack** commands each cost 1 **action point**. However, if the opponent has a summoned card on the **battlefield**, attacking the player directly will cost 2 **action points**.

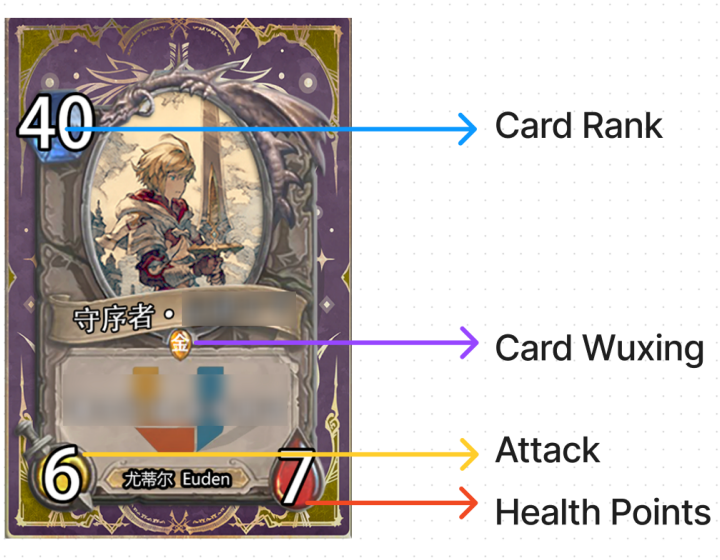
Both players determine who acts first on every turns by comparing the size of their **dice rolls**, and if the dice roll a six, an action point bonus is added to the turn.

Here is the UML of Player and Dice Class:

battleFate :: Player	**battleFate :: Dice**
//fields + playerNumber: int + playerName: String + isHuman: boolean + healthPoints: int + actionPoints: int + isFailed: boolean + inAction: boolean	//fields + ofPlayer: int + maxValue: int + pointsValue: int

// constructors
+ Player(int playerNumber)
 	
// methods
+ inName(): setter playerName
+ activeNow(): setter inAction
+ getATK(): setter healthPoints
+ actionBonus(): setter APs
+ actionSpend(): setter APs
 	
+ isAction(): boolean
+ checkAI(): boolean
+ getHP(): int
+ getAP(): int
+ checkFailed(): boolean
	// constructors
	+ Dice(int ofPlayer)
	
	//methods
	+ roll(): setter pointsVaule
	
	+ getValue(): int
	+ checkSix(): boolean

Here is the Card Class Design:

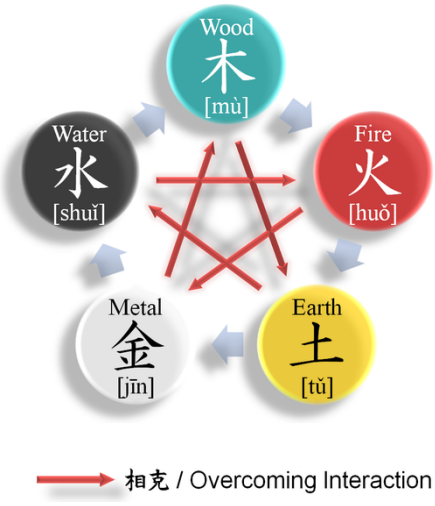


Card Rank: The **rarer** rank, the more **powerful** but the **rarer** in the card pool.

Attack(ATK) and **Health Points(HP)**: Spend **1 AP**(Action Point) to attack a card on the opponent's battlefield. When the card's $HP \leq 0$, it goes to the **graveyard**.

Probability of a card appearing in the card pool $= \frac{\text{card rank}}{20}$.

The remaining $HP_{\text{attacked card}} = HP - ATK_{\text{Attacking card's}}$



Card Elements: A **balancing** mechanism that allows lower ranked cards to be traded directly with higher ranked cards (i.e. both cards go to the **graveyard**) through Wuxing in the traditional Chinese culture.

Here is the UML of Card Class:

battleFate :: **Card**
//fields + cardRank: int + cardName: String + cardElement: String + cardATK: int + cardHP: int + inCardpool: boolean + inHand: boolean + inBattlefield: boolean + isAlive: boolean + ofPlayer: int + isAvailable: boolean + isSelected: boolean

// constructors
+ Card(int cardRank, String cardName, String cardElement, cardATK, cardHP)
 	
// methods
+ beATK(): setter new HP when got Attacked
+ byPlayer(): setter ofPlayer and inCardpool to inHand
+ toBFD(): setter inHand to inBattlefield
+ toGrave(): change inBattlefield to isAlive
+ getActive(): setter isAvailable when Player is active
+ getSelected(): setter isSelected when be selected
 	
+ checkinPool(): boolean
+ checkinHand(): boolean
+ checkAvailable(): boolean
+ checkSelected(): boolean
+ getHP(): int
+ getATK(): int
+ getElement(): String

And some example **Cards data**:

Card Rank	5	6	4	3	2
Card Element	Earth	Fire	Wood	Metal	Water
Card ATK	2	6	6	9	10
Card HP	5	4	7	7	9

Here is the Battlefied UI prototyping sketches: