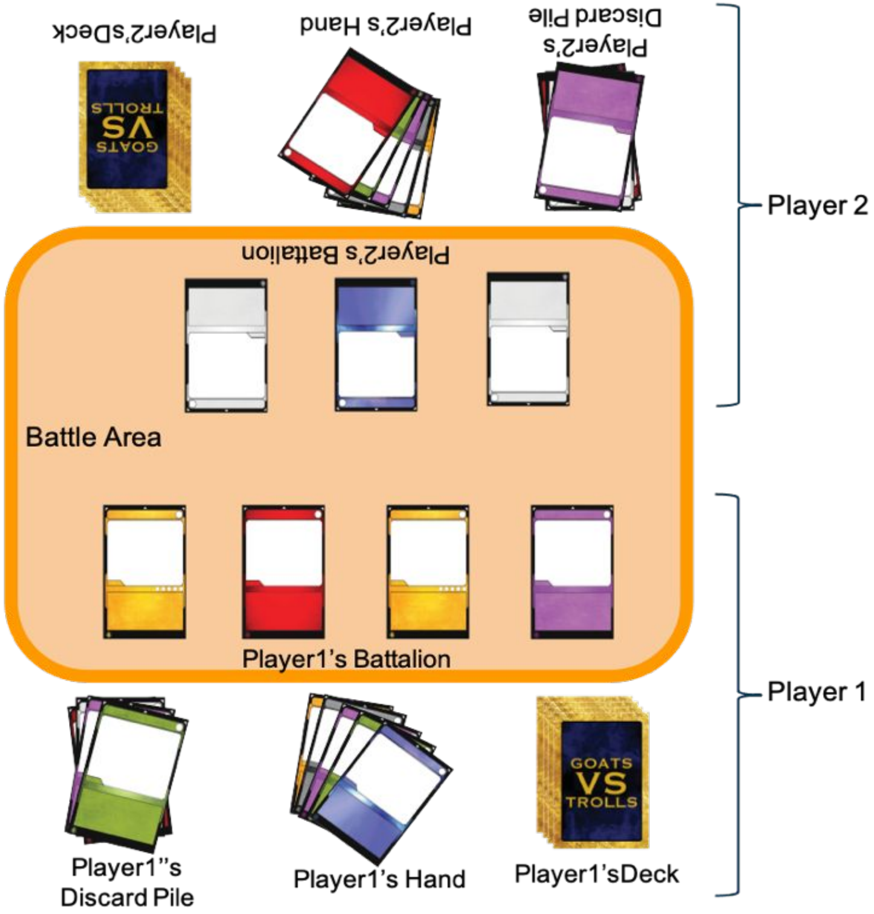

A Game of Goats vs. Trolls

Each player in the game starts with a deck of cards and a hand. In their turn, a player chooses cards from the hand and places them in the battalion. A player’s attack power is computed at the end of the player’s turn as the sum of the battalion’s power. The damage of the player is deducted from cards in the opponent's battalion. If a card's health is reduced to 0, it is removed from the battalion and placed into the discard pile. Players take turns until one of them is defeated.

Problem 1

Perform a problem analysis using the following description for a player. Each player has the following attributes:

- A name.
- A score, which starts at 100. If the score is reduced to 0, the player loses.
- Resource points. These start at 0 but an additional resource point is added at the start of every turn, up to a maximum of 10. Players spend resource points to play cards against their opponent. Resource points are replenished at the start of each turn.
- A deck of GvT cards from which cards are drawn and placed into the hand.
- A hand of cards from which the player chooses cards to play. When the player is created, 5 cards are immediately drawn from the shuffled deck into the hand. An additional card is drawn at the start of each of the player's turns.
- Cards that are currently in play. These are referred to as the player's battalion.
- Cards that have been discarded (after being defeated in battle).

```
Words:

      deck, hand, card, battalion, attack power,
      battalion power, turn, damage, health, discard,
      name, score, resource points, player

class Player:

      name (string)

      score (int)

      resource points (int)

      deck (list)

      hand (list)

      battalion (list)

      discard(list)
```

```
class Player:
      __slots__ = ["__name" , "__score" ,
      "__resource_points" , "__deck" , "__hand" ,
      "__battalion" , "__discarded"]
    def __init__(self, name, deck):
        self.__name = name
        self.__score = 20
        self.__resource_points = 0
        self.__deck = deck
        self.__hand = []
        self.__battalion = []
        self.__discarded = []
    def get_name(self):
        return self.__name
    def get_score(self):
        return self.__score
    def get_resource_points(self):
        return self.__resource_points
    def get_deck(self):
        return self.__deck
```

Problem 2

- Begin implementing your **player class.**
- For now, just focus on the ***fields*** and the ***constructor***.
- Be sure to use proper encapsulation, and note that many of the fields do not need to be initialized using constructor parameters.

Problem 3

Add a function that returns a detailed string representation that matches the example below. The values for the deck and discarded cards are the numbers of cards in each.

```
Player: Buttercup
Score: 16
Resource Points: 5/10
Deck: 24
Discarded: 8
Battalion: [TT 01 06 01][TT 02 06
03][TR 05 06 07]
Hand: [TF 02 02 02][TT 03 07 01]
```

```
def __repr__(self):

      return "\nPlayer: " + str(self.__name) +
      "\nScore: " + str(self.__score) + "\nResource
      Points: " + str(self.__score) + "\nDeck: " +
      str(self.__deck) + "\nDiscarded: " +
      str(self.__discarded) + "\nBattalion: " +
      str(self.__battalion) + "\nHand: " +
      str(self.__hand) + "\n"
```

Problem 4

Add a method named `start_turn` to the `Player` class which is used at the start of the player's turn:

- A resource point is added to the player's resource points at the start of their turn, up to a maximum of 10 points.
- Any resource points spent in the previous turn are also refunded.
- A card is drawn from the player's deck and placed into their hand.

```
def start_turn(self):
      self.__resource_points =
      min(self.__resource_points + 1, 10)
      card = self.__deck.draw()
      self.__hand.append(card)
```