In this scenario, the player initiates the game, triggering the game class to invoke the menu class to present a display featuring a list of buttons—such as "2 players," "3 players," and "4 players"—for the player to choose from. Once the player selects a number of players, the game calls upon the gameboard class to generate 24 tiles with predetermined positions and images. The gameboard class then returns a list of these tiles to the game class, which sets them as attributes.

Following the creation of tiles, the game class instructs the gameboard to generate 16 chitcards. The gameboard first randomizes positions for these chitcards before creating them with the designated images. After all chitcards are created, the game class prompts the gameboard to generate a list of caves based on the previously chosen number of players from the menu. Similar to tiles and chitcards, caves are created with fixed positions. Subsequently, the game class calls upon the gameboard to create a list of dragontokens corresponding to the number of players.

Throughout the process of creating tiles, chitcards, caves, and dragontokens, the gameboard stores these entities for a drawable interface list, enabling the board to be drawn later. Additionally, the gameboard returns these entities to the game class for game control purposes.

Player

: Game

menu: Menu

gameboard: Gameboard

run the game

display a menu for selecting player number

The number of players

create tiles

create 24 tile with fixed position and image

tile: Tile

a list of 24 tile

create 16 chit cards

generate a list of randomised chit card position

a list of chit card position

create 12 forward chit card with allocated position and image

chitcard: ForwardChitCard

create 4 backward chit card(pirate dragon) with allocated position and image

chitcard: BackwardChitCard

a list of 16 chit cards

create cave with the number of player

create cave based on the input player number with fixed position and image

cave: Cave

a list of caves

create dragon token with the number of player

create dragon token based on the input player number with fixed position and image

dragontoken: DragonToken

a list of dragon tokens

Loop   [for each drawable item]

Iterate through the list of drawable entities we have and render them.

draw

draw

draw

draw

draw