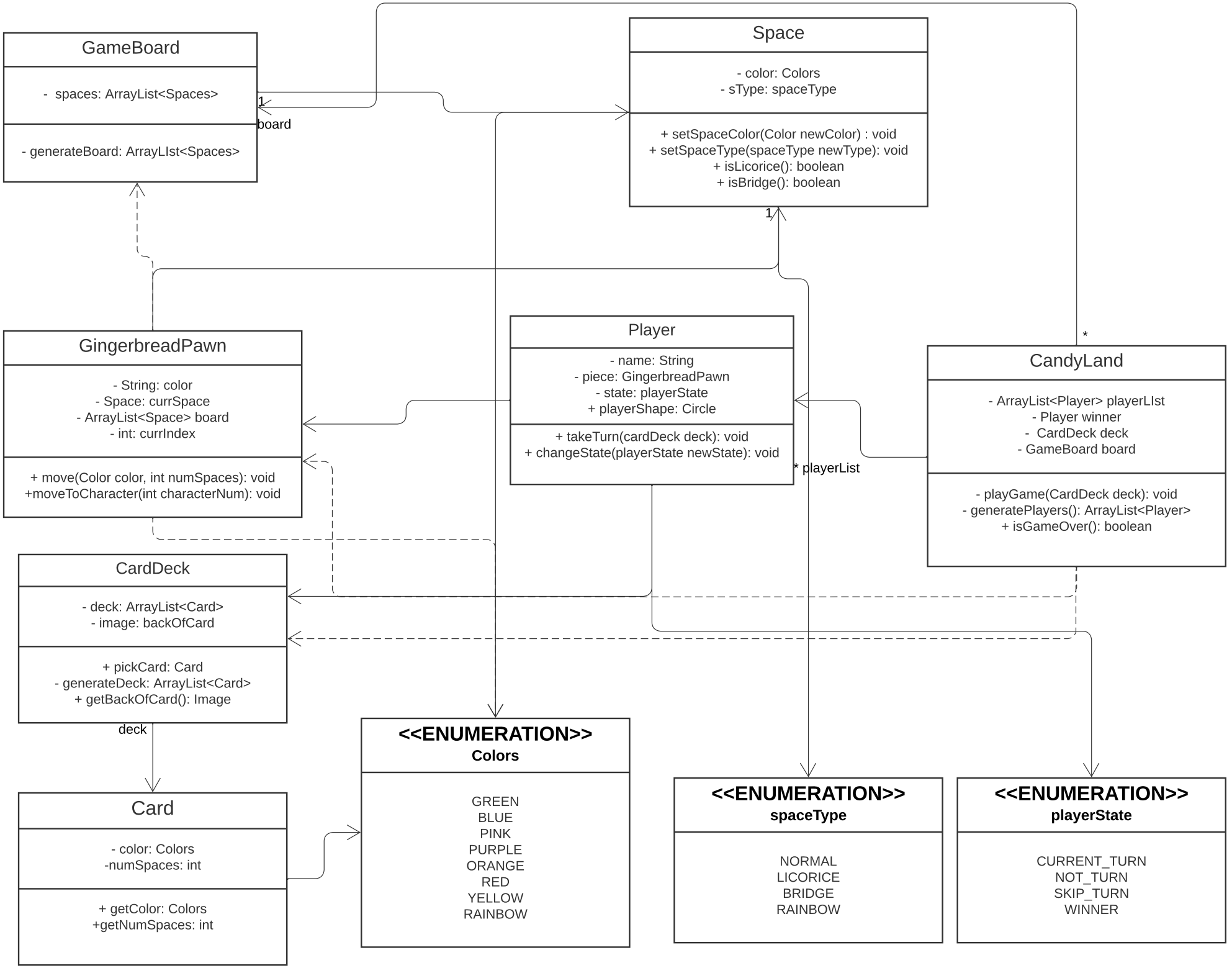
GameBoard
- spaces: ArrayList<Spaces>
- generateBoard: ArrayLIst<Spaces>
1
board
Space
- color: Colors
- sType: spaceType
+ setSpaceColor(Color newColor) : void
+ setSpaceType(spaceType newType): void
+ isLicorice(): boolean
+ isBridge(): boolean
1
*
GingerbreadPawn
- String: color
- Space: currSpace
- ArrayList<Space> board
- int: currIndex
+ move(Color color, int numSpaces): void
+moveToCharacter(int characterNum): void
Player
- name: String
- piece: GingerbreadPawn
- state: playerState
+ playerShape: Circle
+ takeTurn(cardDeck deck): void
+ changeState(playerState newState): void
* playerList
CandyLand
- ArrayList<Player> playerLIst
- Player winner
- CardDeck deck
- GameBoard board
- playGame(CardDeck deck): void
- generatePlayers(): ArrayList<Player>
+ isGameOver(): boolean
CardDeck
- deck: ArrayList<Card>
- image: backOfCard
+ pickCard: Card
- generateDeck: ArrayList<Card>
+ getBackOfCard(): Image
deck
Card
- color: Colors
-numSpaces: int
+ getColor: Colors
+getNumSpaces: int
<<ENUMERATION>>
Colors
GREEN
BLUE
PINK
PURPLE
ORANGE
RED
YELLOW
RAINBOW
<<ENUMERATION>>
spaceType
NORMAL
LICORICE
BRIDGE
RAINBOW
<<ENUMERATION>>
playerState
CURRENT_TURN
NOT_TURN
SKIP_TURN
WINNER