Animal «enum»

.salamander
.baby_dragon
.spider
.bat

AnimalChitCard

- animal: Animal

+ perform_effect(PlayableCharacter) : Void

PirateChitCard

+ perform_effect(PlayableCharacter) : Void

DefaultGameBoard.__init__: First parameter is set main tile sequence. Second parameter is
[(starting tile, next tile)]. The second parameter (starting tiles) will be used to make insertions to
main tile sequence property.
DefaultGameBoard.starting_tiles: [starting tiles]
DefaultGameBoard.move_character_by_steps: Moves a dragon from its tile by x number of
steps
DefaultGameBoard.flip_chit_card: Flip a chit card on this game board. Chit card object will be
obtained by click listener in pygame.
DefaultGameBoard.add_chit_card: Add a chit card to the game board

DefaultGameBoard

- main_tile_sequence: [Tile]
- starting_tiles: [Tile]
- chit_cards: [ChitCard] = []

+ __init__([Tile], [(Tile, Tile)])
+ move_character_by_steps(PlayableCharacter, Int) : Void
+ get_character_floor_tile(PlayableCharacter) : Tile
+ flip_chit_card(PlayableCharacter, ChitCard) : Void
+ add_chit_card(ChitCard)

«abstract» ChitCard

- on_game_board: GameBoard
- flipped: Bool
- symbol_count: Int

+ __init__(GameBoard, Int)
+ perform_effect(PlayableCharacter) : Void
+ set_flipped(Bool) : Void

ChitCard.__init__: Create a chit card with the # number of symbols
ChitCard.perform_effect: Perform the chit card's effect on the dragon.
Run by flipChitCard
ChitCard.set_flipped: Set flipped

Entities & Responsibilities

GameWorld: Manages game initialisation & admin tasks (e.g game
config, game setup, main game loop, player turn.)

Tile: Represents the tiles which the dragons will interact with (stand on)

PlayableEntity: Represents the playable entity a player interacts with

GameBoard: Runs the interactions with the game board by the players
(e.g performing movement, flipping chit card)

ChitCard: Represents the chit cards and their effects

TODO

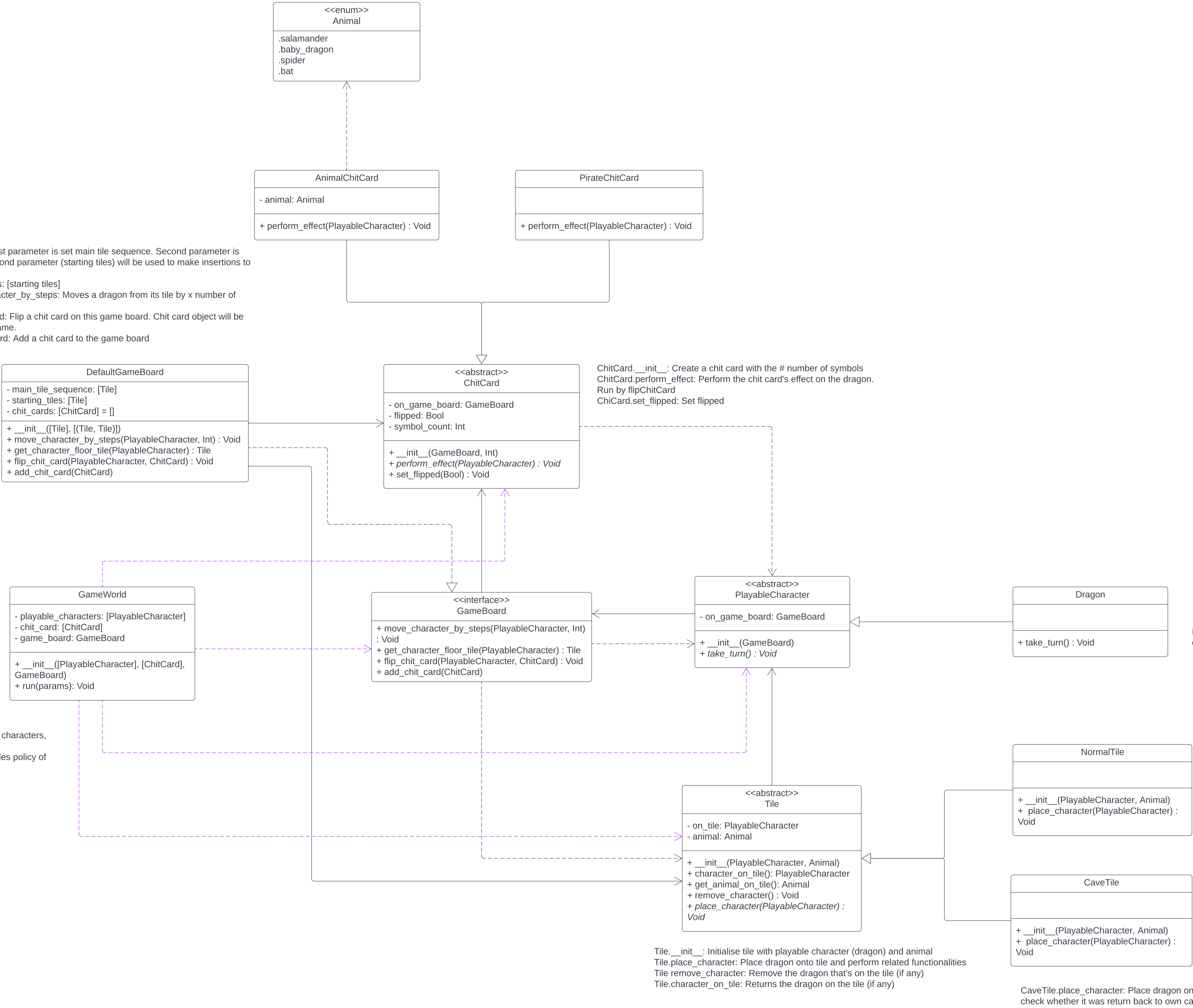Builder pattern for configuring game board? (e.g number of chit cards,
number of players?)

Cardinalities

NOTES
Upcasts are safe

GameWorld

- playable_characters: [PlayableCharacter]
- chit_card: [ChitCard]
- game_board: GameBoard

+ __init__([PlayableCharacter], [ChitCard], GameBoard)
+ run(params): Void

«interface» GameBoard

+ move_character_by_steps(PlayableCharacter, Int) : Void
+ get_character_floor_tile(PlayableCharacter) : Tile
+ flip_chit_card(PlayableCharacter, ChitCard) : Void
+ add_chit_card(ChitCard)

«abstract» PlayableCharacter

- on_game_board: GameBoard

+ __init__(GameBoard)
+ take_turn() : Void

Dragon

+ take_turn() : Void

Dragon.take_turn: Allows a dragon to take its turn & perform actions (e.g chit
card flipping)

GameApplication.__init__: Configure and intialise playable characters,
game board & its tiles & its chit cards.
GameApplication.run: Contains the main game loop. Handles policy of
player turns

NormalTile

+ __init__(PlayableCharacter, Animal)
+ place_character(PlayableCharacter) : Void

«abstract» Tile

- on_tile: PlayableCharacter
- animal: Animal

+ __init__(PlayableCharacter, Animal)
+ character_on_tile(): PlayableCharacter
+ get_animal_on_tile(): Animal
+ remove_character() : Void
+ place_character(PlayableCharacter) : Void

CaveTile

+ __init__(PlayableCharacter, Animal)
+ place_character(PlayableCharacter) : Void

Tile.__init__: Initialise tile with playable character (dragon) and animal
Tile.place_character: Place dragon onto tile and perform related functionalities
Tile remove_character: Remove the dragon that's on the tile (if any)
Tile.character_on_tile: Returns the dragon on the tile (if any)

CaveTile.place_character: Place dragon onto tile and perform
check whether it was return back to own cave, trigger win