<<enum>>
Animal

.salamander
.baby_dragon
.spider
.bat

AnimalChitCard

- animal: Animal

+ perform_effect(Dragon) : Void

PirateChitCard

+ perform_effect(Dragon) : Void

ChitCard.perform_effect: Perform the chit card's effect on the dragon.
Run by flipChitCard
ChiCard.set_flipped: Set flipped

<>
ChitCard

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

+ perform_effect(Dragon) : Void
+ set_flipped(Bool) : Void

Entities & Responsibilities

GameApplication: 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)

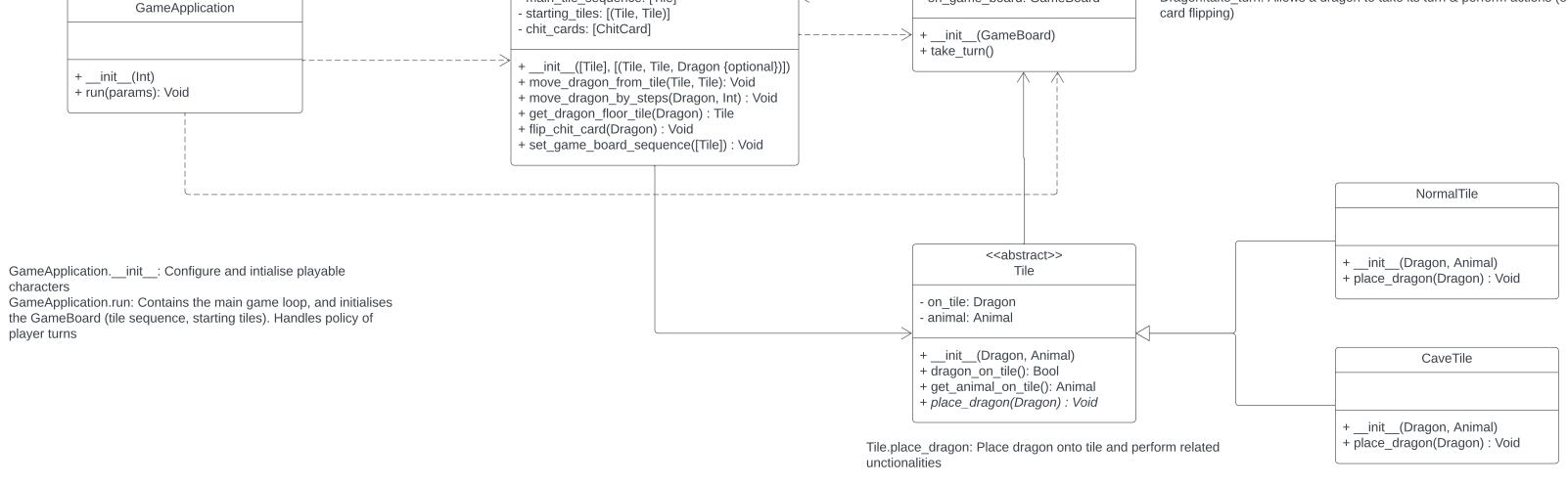Dragon: Represents the playable entity a player interacts with

GameBoard: Runs the game board that users can interact with

ChitCard: Represents the chit cards and their effects

TODO

Builder pattern for configuring game board?

Dragon should be an interface instead (e.g Playable)

GameBoard.__init__: Configure main tile sequence. The second parameter (starting tiles) will be used to make insertions to main tile sequence
GameBoard.move_dragon_from_tile: Moves a dragon from source to destination tile
GameBoard.move_dragon_by_steps: Moves a dragon from its tile by x number of steps
GameBoard.set_game_board_sequence: Set tile sequence for game board
GameBoard.flip_chit_card: Flip a chit card on this game board
GameBoard.starting_tiles: [(starting tile, next tile)]

GameApplication

+ __init__(Int)
+ run(params): Void

GameBoard

- main_tile_sequence: [Tile]
- starting_tiles: [(Tile, Tile)]
- chit_cards: [ChitCard]

+ __init__([Tile], [(Tile, Tile, Dragon {optional})])
+ move_dragon_from_tile(Tile, Tile) : Void
+ move_dragon_by_steps(Dragon, Int) : Void
+ get_dragon_floor_tile(Dragon) : Tile
+ flip_chit_card(Dragon) : Void
+ set_game_board_sequence([Tile]) : Void

Dragon

- on_game_board: GameBoard

+ __init__(GameBoard)
+ take_turn()

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

GameApplication.__init__: Configure and intialise playable characters
GameApplication.run: Contains the main game loop, and initialises the GameBoard (tile sequence, starting tiles). Handles policy of player turns

---

NormalTile

+ __init__(Dragon, Animal)
+ place_dragon(Dragon) : Void

<>
Tile

- on_tile: Dragon
- animal: Animal

+ __init__(Dragon, Animal)
+ dragon_on_tile(): Bool
+ get_animal_on_tile(): Animal
+ place_dragon(Dragon) : Void

CaveTile

+ __init__(Dragon, Animal)
+ place_dragon(Dragon) : Void

Tile.place_dragon: Place dragon onto tile and perform related unctionalities

Tile.dragon_on_tile: Returns whether the dragon is on the tile

---

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

Tile.dragon_on_tile: Returns whether the dragon is on the tile