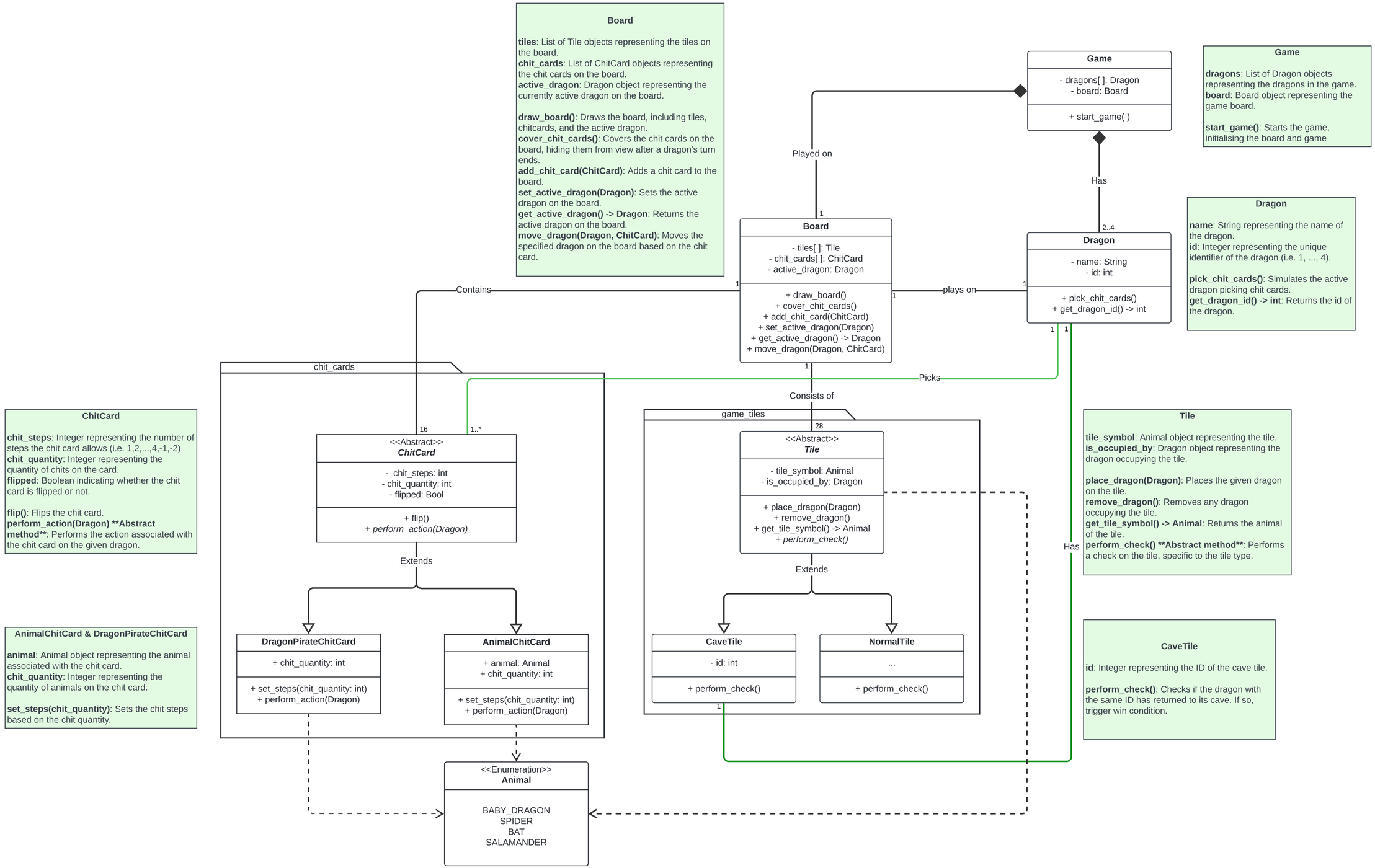
Board
tiles: List of Tile objects representing the tiles on the board.
chit_cards: List of ChitCard objects representing the chit cards on the board.
active_dragon: Dragon object representing the currently active dragon on the board.
draw_board(): Draws the board, including tiles, chitcards, and the active dragon.
cover_chit_cards(): Covers the chit cards on the board, hiding them from view after a dragon's turn ends.
add_chit_card(ChitCard): Adds a chit card to the board.
set_active_dragon(Dragon): Sets the active dragon on the board.
get_active_dragon() -> Dragon: Returns the active dragon on the board.
move_dragon(Dragon, ChitCard): Moves the specified dragon on the board based on the chit card.
Game
- dragons[ ]: Dragon
- board: Board
+ start_game( )
Game
dragons: List of Dragon objects representing the dragons in the game.
board: Board object representing the game board.
start_game(): Starts the game, initialising the board and game
Played on
Has
1
2..4
Board
- tiles[ ]: Tile
- chit_cards[ ]: ChitCard
- active_dragon: Dragon
+ draw_board()
+ cover_chit_cards()
+ add_chit_card(ChitCard)
+ set_active_dragon(Dragon)
+ get_active_dragon() -> Dragon
+ move_dragon(Dragon, ChitCard)
Dragon
- name: String
- id: int
+ pick_chit_cards()
+ get_dragon_id() -> int
Dragon
name: String representing the name of the dragon.
id: Integer representing the unique identifier of the dragon (i.e. 1, ..., 4).
pick_chit_cards(): Simulates the active dragon picking chit cards.
get_dragon_id() -> int: Returns the id of the dragon.
Contains
plays on
Picks
Consists of
chit_cards
game_tiles
16
1..*
28
<<Abstract>>
ChitCard
- chit_steps: int
- chit_quantity: int
- flipped: Bool
+ flip()
+ perform_action(Dragon)
<<Abstract>>
Tile
- tile_symbol: Animal
- is_occupied_by: Dragon
+ place_dragon(Dragon)
+ remove_dragon()
+ get_tile_symbol() -> Animal
+ perform_check()
Has
Tile
tile_symbol: Animal object representing the tile.
is_occupied_by: Dragon object representing the dragon occupying the tile.
place_dragon(Dragon): Places the given dragon on the tile.
remove_dragon(): Removes any dragon occupying the tile.
get_tile_symbol() -> Animal: Returns the animal of the tile.
perform_check() **Abstract method**: Performs a check on the tile, specific to the tile type.
ChitCard
chit_steps: Integer representing the number of steps the chit card allows (i.e. 1,2,...,4,-1,-2)
chit_quantity: Integer representing the quantity of chits on the card.
flipped: Boolean indicating whether the chit card is flipped or not.
flip(): Flips the chit card.
perform_action(Dragon) **Abstract method**: Performs the action associated with the chit card on the given dragon.
Extends
Extends
DragonPirateChitCard
+ chit_quantity: int
+ set_steps(chit_quantity: int)
+ perform_action(Dragon)
AnimalChitCard
+ animal: Animal
+ chit_quantity: int
+ set_steps(chit_quantity: int)
+ perform_action(Dragon)
CaveTile
- id: int
+ perform_check()
NormalTile
...
+ perform_check()
1
AnimalChitCard & DragonPirateChitCard
animal: Animal object representing the animal associated with the chit card.
chit_quantity: Integer representing the quantity of animals on the chit card.
set_steps(chit_quantity): Sets the chit steps based on the chit quantity.
CaveTile
id: Integer representing the ID of the cave tile.
perform_check(): Checks if the dragon with the same ID has returned to its cave. If so, trigger win condition.
<<Enumeration>>
Animal
BABY_DRAGON
SPIDER
BAT
SALAMANDER