GAME

Attributes:

- player_num: Holds the number of players selected through select_player_num(int) and initializes the game accordingly.
- players: Stores player states and facilitates turn processing.
- gameboard: Utilized for rendering all drawable entities.
- menu: Manages the display of menu options and obtains the number of players.
- chitcards: Determines which chit card is clicked and inputs the dragon token into the chit card method to execute a move.

Method:

- select_player_num(): Initialize the player count.
- initialize_game(): Generate tiles, caves, chit cards, and dragon tokens according to the number of players.
- game_loop(): Continuously execute the game and manage player input.
- process_play_turn(): Manage the player's turn, iterate through available options, and update the position of the dragon token.
- check_player_winning(): This method is executed in each loop to determine if any player has won the game.
- run(): Execute all the aforementioned methods when called by the driver class.

GameBoard

Attributes:

- drawables: Stores a list of drawable entities to be rendered in each game loop iteration.

Method:

- create_caves(player_num): Generates a list of caves based on the input player_num and returns the list to the game.
- create_dragon_tokens(player_num): Creates a list of dragon tokens based on the number of players and returns the list to the game.
- create_tiles(): Generates a list of tiles and returns it.
- create_chit_cards(): Creates a list of chit cards and returns it to the game class.
- randomise_chit_card_position(): Called by create_chit_cards() before creating the chit cards to randomize their positions. This ensures that every chit card is created with a randomized position.
- draw_gameboard(): Iterates through all drawable objects and renders them on the board.
- display_winning_message(): Show the winning message indicating the specified player.

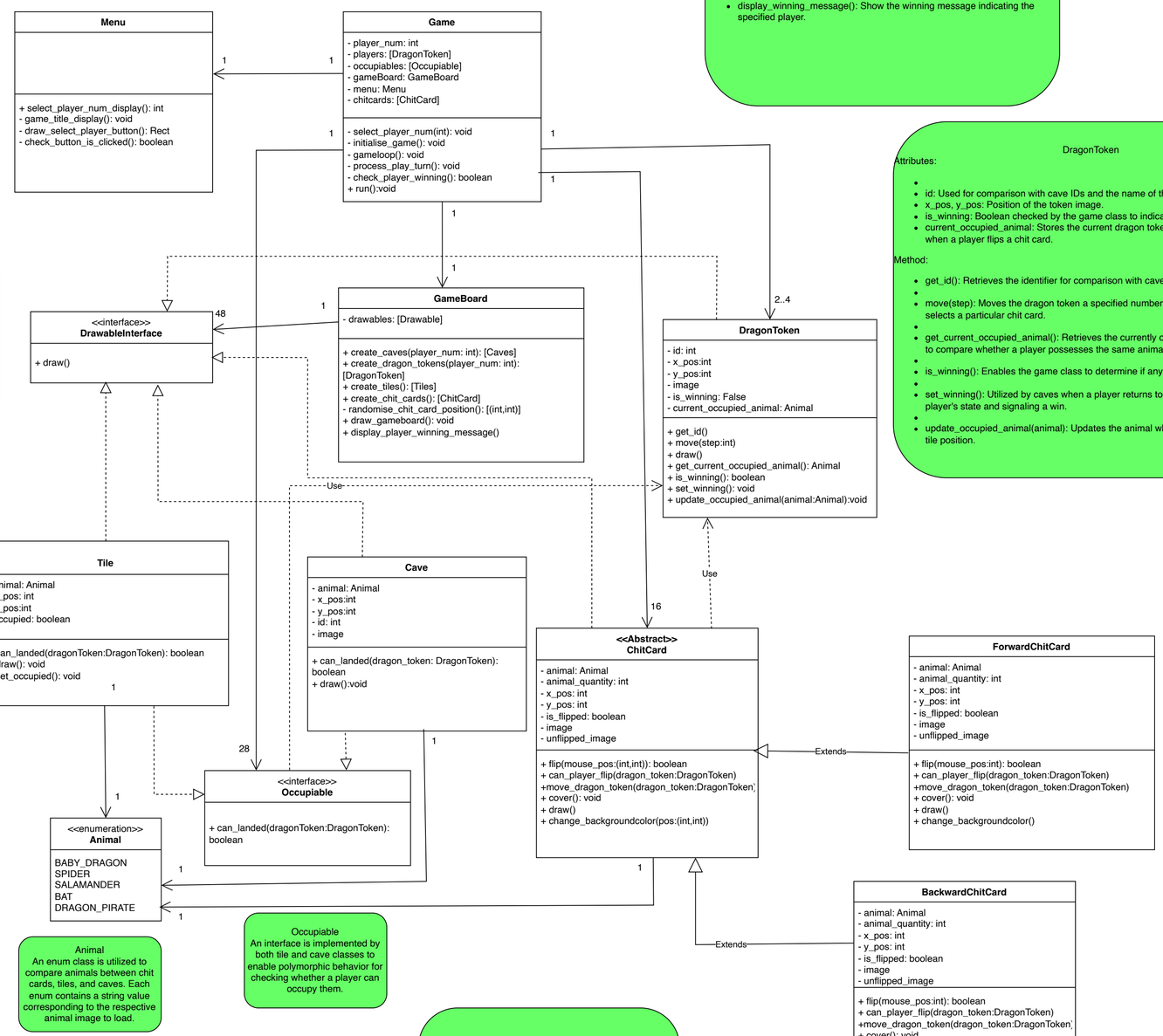
Menu

Method:
select_player_num_display(): Displays a menu allowing the player to select the number of players and returns the selected number to the game class.

game_title_display(): Renders the game title on the board.

draw_select_player_button(): Draws buttons for different numbers of players on the board and returns the button image rectangle.

check_button_is_clicked(): Called by select_player_num_display() to determine which button is clicked by the player.



DrawableInterface
An interface is implemented by all entities that need to be rendered on the board to enable polymorphic behavior for the gameboard.

DragonToken

Attributes:

-
- id: Used for comparison with cave IDs and the name of the image.
- x_pos, y_pos: Position of the token image.
- is_winning: Boolean checked by the game class to indicate if the game should stop.
- current_occupied_animal: Stores the current dragon token position animal to compare when a player flips a chit card.

Method:

- get_id(): Retrieves the identifier for comparison with cave IDs.
-
- move(step): Moves the dragon token a specified number of steps after a player selects a particular chit card.
-
- get_current_occupied_animal(): Retrieves the currently occupied animal when needed to compare whether a player possesses the same animal.
-
- is_winning(): Enables the game class to determine if any player has won the game.
-
- set_winning(): Utilized by caves when a player returns to their cave, updating the player's state and signaling a win.
-
- update_occupied_animal(animal): Updates the animal when a player changes their tile position.

ChitCard

Attributes:

- animal: The animal represented by this chit card.
- animal_quantity: The number of animals on this chit card, determining the number of steps a player needs to take.
- is_flipped: A boolean indicating whether the chit card is flipped, used by the game board to draw either the flipped or unflipped image.
- image: The image displayed when the chit card is flipped.
- unflipped_image: The image displayed when the chit card is covered.

Method:

- flip(): Determines if the clicked mouse position is within the chit card's area and returns true if the chit card is successfully flipped.
- can_player_flip(dragon_token): Checks whether the player can still flip a chit card after flipping this particular chit card.
- cover(): Utilized by the game class at the end of a player's turn to cover all chit cards.
- move_dragon_token(dragon_token): Adjusts the position of the dragon token based on the quantity of animals depicted on the chit card.
- draw(): Used by the gameboard class to render the gameboard on the screen.
- change_backgroundcolor(pos): Alters the background color when the player's mouse is positioned over this chit card.

Tile

Attribute:

- animal: The animal represented by this tile.
- x_pos, y_pos: Position of the tile
- occupied: A boolean indicating whether a dragon has occupied it.

Method:

- can_landed(dragonToken): Returns false if the tile is occupied by another dragon token; otherwise, returns true.
- draw(): Called by the gameboard to render the tile.
- set_occupied(): When a dragon token moves toward this tile, this function is invoked to set the "occupied" boolean to true.

ForwardChitCard

Attribute:

- references to chitcard class

Method:

- flip(): Determines if the clicked mouse position is within the chit card's area and returns true if the chit card is successfully flipped.
- can_player_flip(dragon_token): Returns true if the player selects this chit card with a matched animal and false if it does not match.
- move_dragon_token(dragon_token): Advances the player's position based on the quantity of animals depicted on the chit card. This function is only called if the player matches the animal type.e.
- cover(): Utilized by the game class at the end of a player's turn to cover all chit cards.
- draw(): Used by the gameboard class to render the gameboard on the screen.
- change_backgroundcolor(pos): Alters the background color when the player's mouse is positioned over this chit card.

Animal
An enum class is utilized to compare animals between chit cards, tiles, and caves. Each enum contains a string value corresponding to the respective animal image to load.

Occupiable
An interface is implemented by both tile and cave classes to enable polymorphic behavior for checking whether a player can occupy them.

Cave

Attribute:

- animal: The animal represented by this cave. Players need to flip a chit card matching this animal to exit the cave.
- x_pos, y_pos: Position of the cave image.
- id: An attribute used to compare with dragon tokens when they attempt to enter.
- image: The image displayed on the board.

Method:

- can_landed(dragonToken): Returns false if the dragonToken does not match the initial dragon token that emerged. If the dragon token matches, it calls the dragon token's set_winning() function.
- draw(): Invoked by the gameboard to display the cave.

BackwardChitCard

Attribute:

- references to chitcard class

Method:

- flip(): Determines if the clicked mouse position is within the chit card's area and returns true if the chit card is successfully flipped.
- can_player_flip(dragon_token): Returns true if the player selects this chit card. Players can still flip cards if they select a piratedragon card.
- move_dragon_token(dragon_token): Moves the dragon token backward a number of steps based on the quantity of animals depicted on the chit card.
- cover(): Utilized by the game class at the end of a player's turn to cover all chit cards.
- draw(): Used by the gameboard class to render the gameboard on the screen.
- change_backgroundcolor(pos): Alters the background color when the player's mouse is positioned over this chit card.