

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

EventBus: Handles registration of listeners, and notification of appropriate listeners on event fire

WinEventXxxx: Publisher and listeners for win event

DrawableByAsset: Indicates that the object is drawable by pygame using assets

DrawAssetInstruction: A data class for organising data required for drawing an asset

Patterns Used

Observer: WinEventPublisher, WinEventListener
• Why?: Don't have to check all starting tiles to see if win occurred. Allows for wins from other sources

Singleton: EventBus
• Why?: Should be one central event bus managing all events

Todo

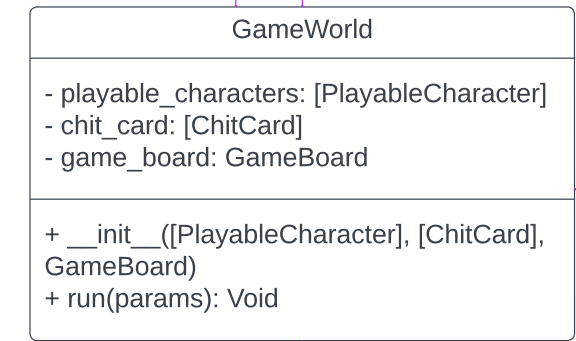
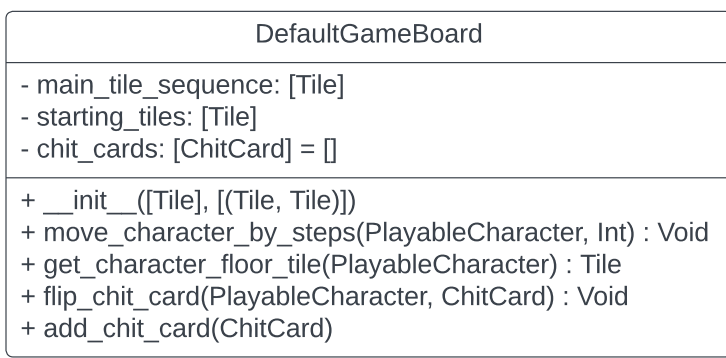
Cardinalities

Notes

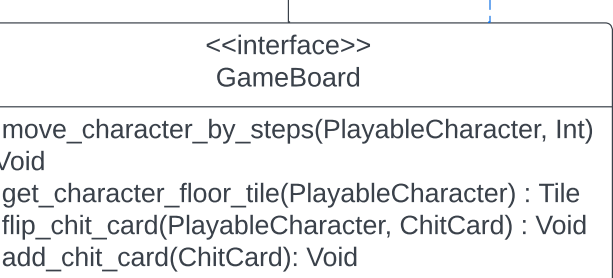
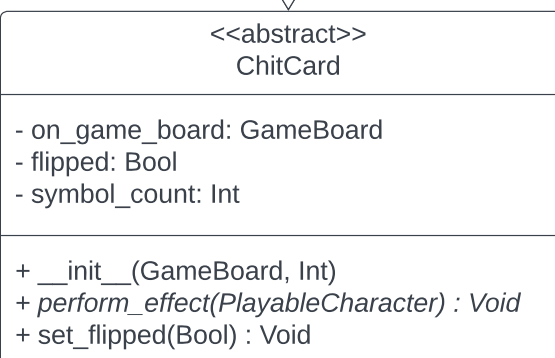
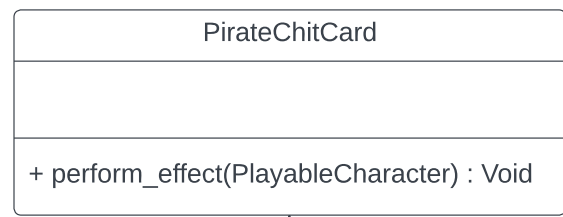
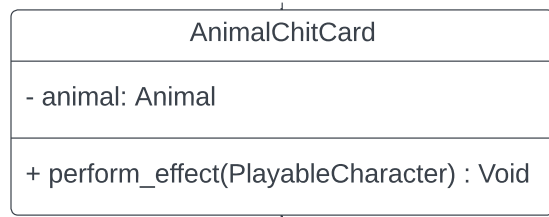
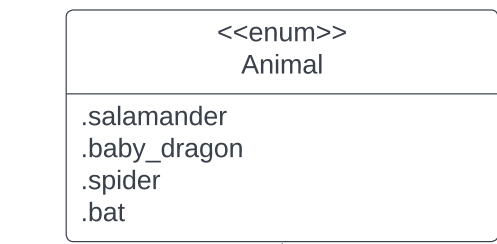
Upcasts are safe

GameApplication.__init__: Configure and initialise playable characters, game board & its tiles & its chit cards.
GameApplication.run: Contains the main game loop. Handles policy of player turns, and drawing logic (based on drawable)

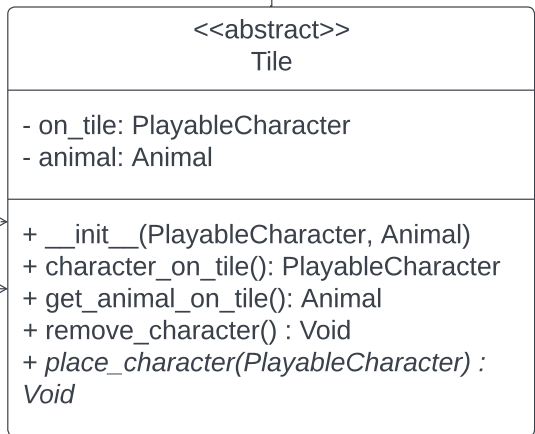
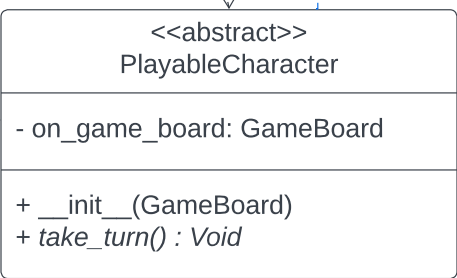
DefaultGameBoard.__init__: First parameter is set main tile sequence. Second parameter is [(starting tile, next tile)]. The second parameter (starting tiles) will be used to make insertions to main tile sequence property.
DefaultGameBoard.starting_tiles: [starting tiles]
DefaultGameBoard.move_character_by_steps: 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



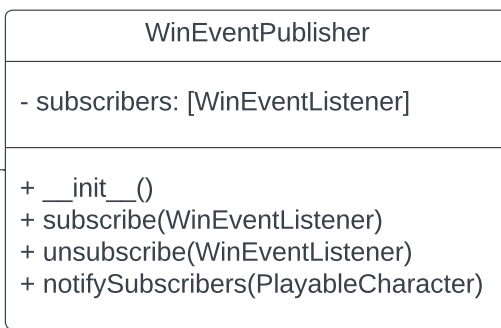
WinEventListener.onPlayerWin: On a player win, do something



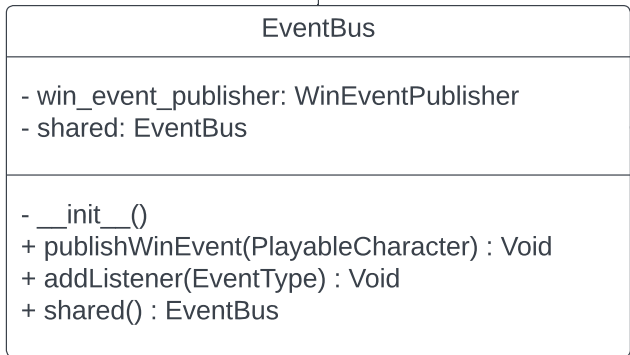
ChitCard.__init__: Create a chit card with the # number of symbols
ChitCard.perform_effect: Perform the chit card's effect on the dragon.
Run by flipChitCard
ChitCard.set_flipped: Set flipped



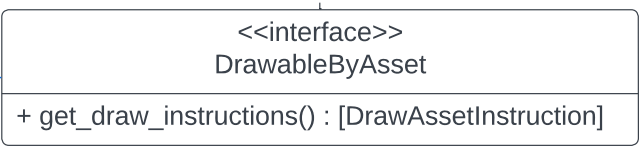
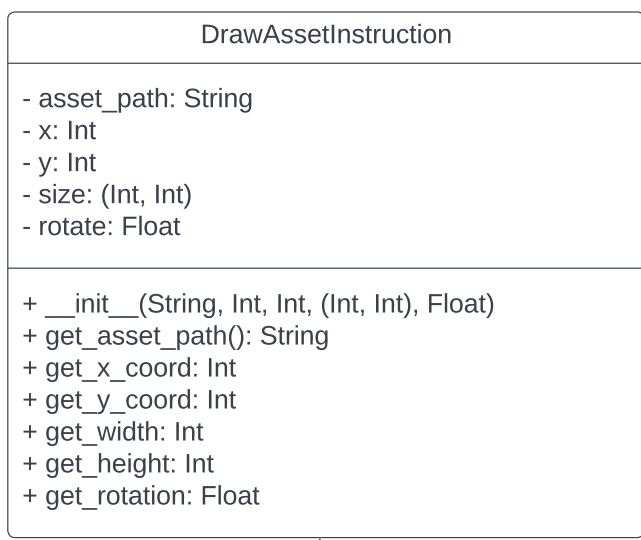
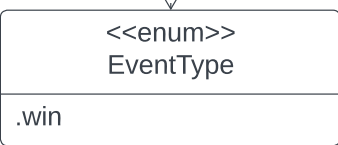
Tile.__init__: Initialise tile with playable character (dragon) and animal
Tile.place_character: Place dragon onto tile and perform related functionalities
Tile.remove_character: Remove the dragon that's on the tile (if any)
Tile.character_on_tile: Returns the dragon on the tile (if any)



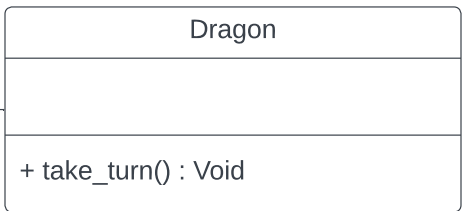
WinEventPublisher.notifySubscribers: Notify subscribers about the player who won



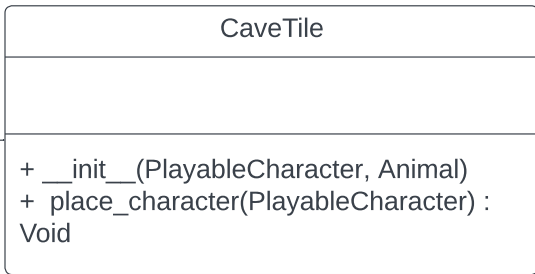
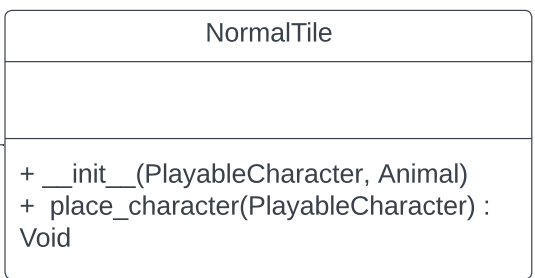
EventBus.shared: Get the eventbus singleton
EventBus.publishWinEvent: Publish win event onto bus
EventBus.addListener: Add a listener for a certain event type



DrawableByAsset.get_draw_instructions: Gets the drawing instructions for drawing an object onto the pygame screen using assets



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



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