Entities & Responsibilities

GameWorld: Manages the game instance & the interface between it and the players (e.g processing for events) by running the main game loop

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

DrawProperties: Data class for organising the drawing properties required for drawing any element

PlayableCharacter: Represents the playable character a player interacts with

GameBoard: Represents the game board. It initialises the game board & runs the interactions with the game board by

ChitCard: Represents the chit cards and their effects

the players (e.g performing movement, flipping chit card)

EventBus: Handles registration of listeners, and notification of appropriate listeners on event fire. [Class is for organising all EventListeners into one place]

WinEventXxxx....: Publisher and listeners for win event

MoveActionXxxx....: Publisher and listeners for a move action (for characters) that is fired

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

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

ModularClickableSprite: Allows classes to be represented as a sprite that is clickable on a screen.

PlayableCharacterVariant: Represents a variant for the playable character entity

PygameScreenController: Contains useful methods for interacting with pygame's (the game's) screen

main.py: Game config before start, and serves as entry point to game

Patterns Used

Observer: WinEventPublisher, WinEventListener

Why?: Don't have to check all starting tiles to see if win occured. Allows for wins from other sources

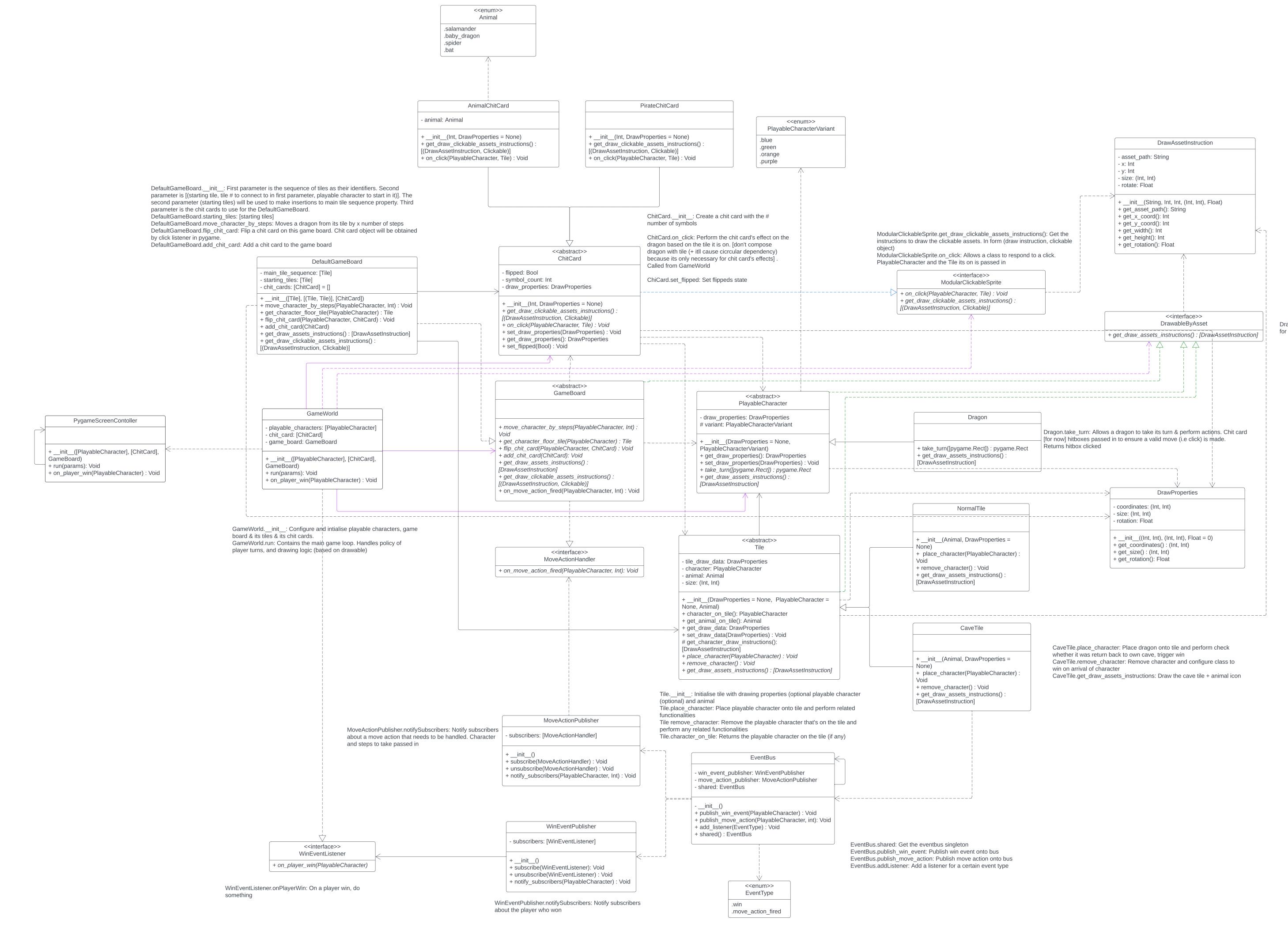
Singleton: EventBus, PygameScreenController

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

Cardinalities

<u>Later Refactorings Required</u>

Tile now can retrieve & store animal (optional?). Animals are crucial to the movement system of the game.



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