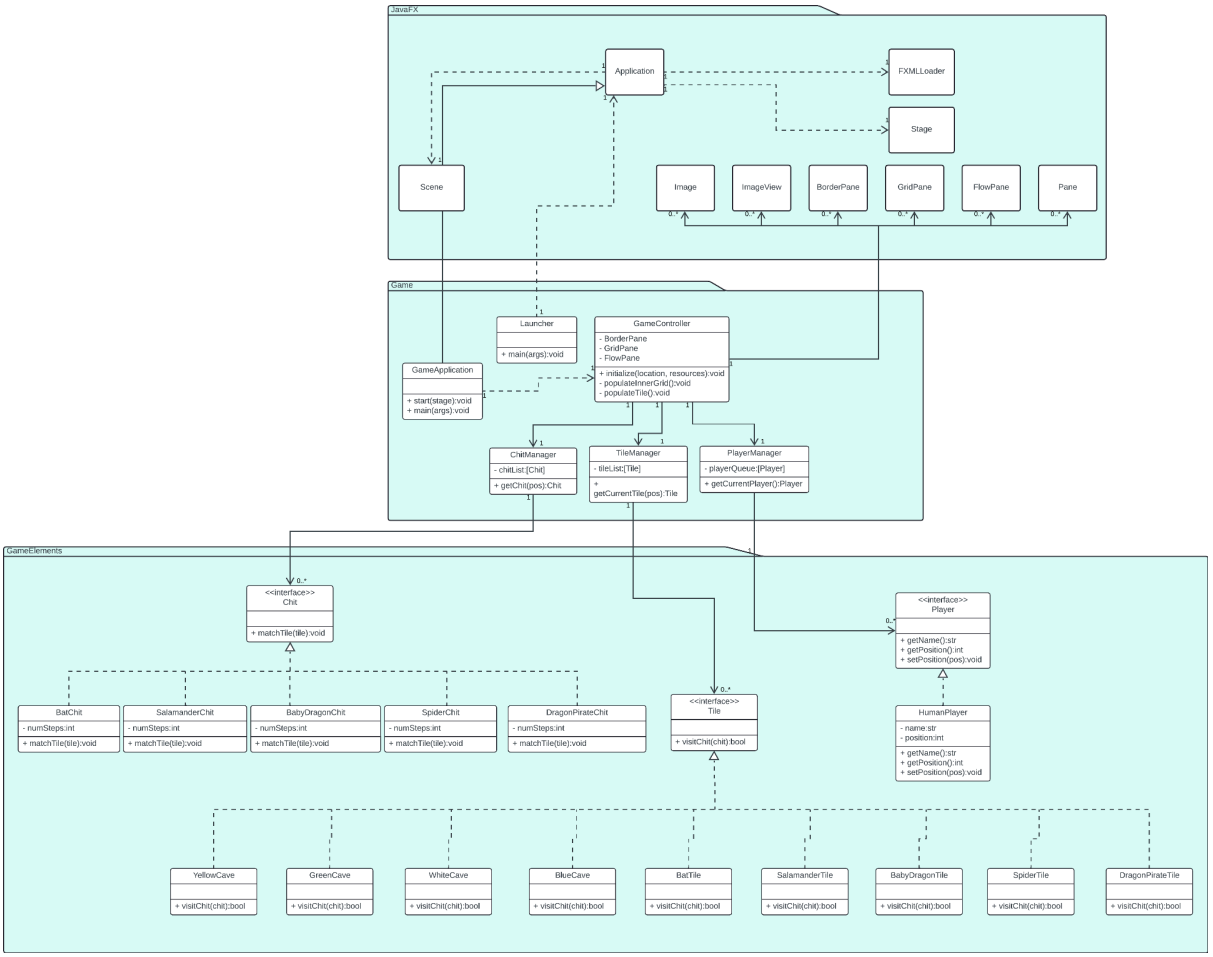
Class Diagram



Sequence Diagrams

Design Rationales

HumanPlayer

This class has been added instead of having a single Player class without extending from an interface so that the game is open to extensibility in the future. Different implementations of Player such as AIPlayer, RemotePlayer, and more may be added. This design also allows each type of player to have their own implementation of methods which are likely to be different. It is not appropriate to have different methods in the same class to handle each kind of player because the concept of these different kinds of players are better represented as a class.

BatChit

The BatChit, and every other dragon chit, have been implemented as concrete classes implementing the Chit interface instead of having a Chit class with an attribute which defines the animal. This is because each concrete dragon chit represents a specific type of animal, which promotes a clear separation of concerns. With a single Chit class and an attribute defining the animal, it would have to include conditional logic to handle the different behaviours of each animal chit. Despite each animal chit being functionally the same in the base game, they may be extended to have different effects in the future.

GameApplication - Scene

A scene is a JavaFX class used for displaying content (in this case, it will display the game board). The relationship between GameApplication and Scene is a composite relationship since the Scene object responsible for displaying the board is created within the context of GameApplication, its life cycle is dependent on it, and is directly managed by it; it is tightly coupled with the GameApplication object.

ChitManager - Chit

This is an aggregate relationship, because Chit implementations can be contained within ChitManager, but they do not require ChitManager to serve their function or exist.