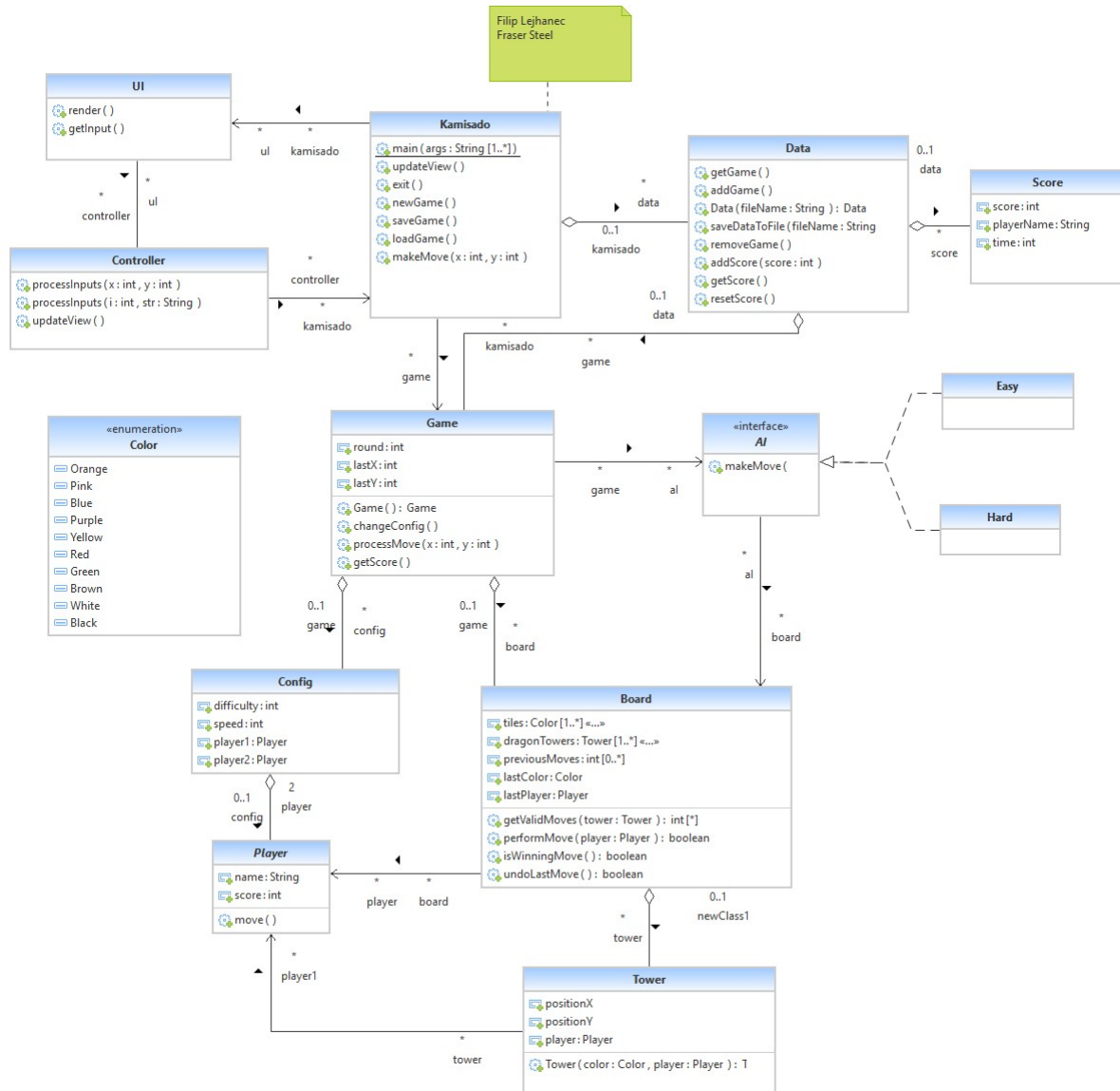


Kamisado Design



Rationale

We tried to implement the MVC model in order to decouple the design. The UI draws the interface and gets the user input, which is then processed by the controller. The controller makes the appropriate calls on the Kamisado model. Kamisado notifies the UI when a change has been made. Permanent data is handled by the Data class. Kamisado class controls the model. The Game class handles the inputs passed on to it by the Kamisado class and according to the configuration can call AI to make a next move. The Strategy model is used to implement the different difficulties of AI. The Board class is responsible for evaluating the validity of any inputs given to it before performing them. References to the Player class are used to validate whether the appropriate player is making a move. The Board contains all of its Towers and basically is a representation of the state of the game. AI uses a reference to the Board to evaluate the next move. Color enumeration is used to better abstract the color values throughout the system.