Peter Maitland, Christopher Pauly, Matus Mikus, Petr Sramek

JOOSE lab group ??????

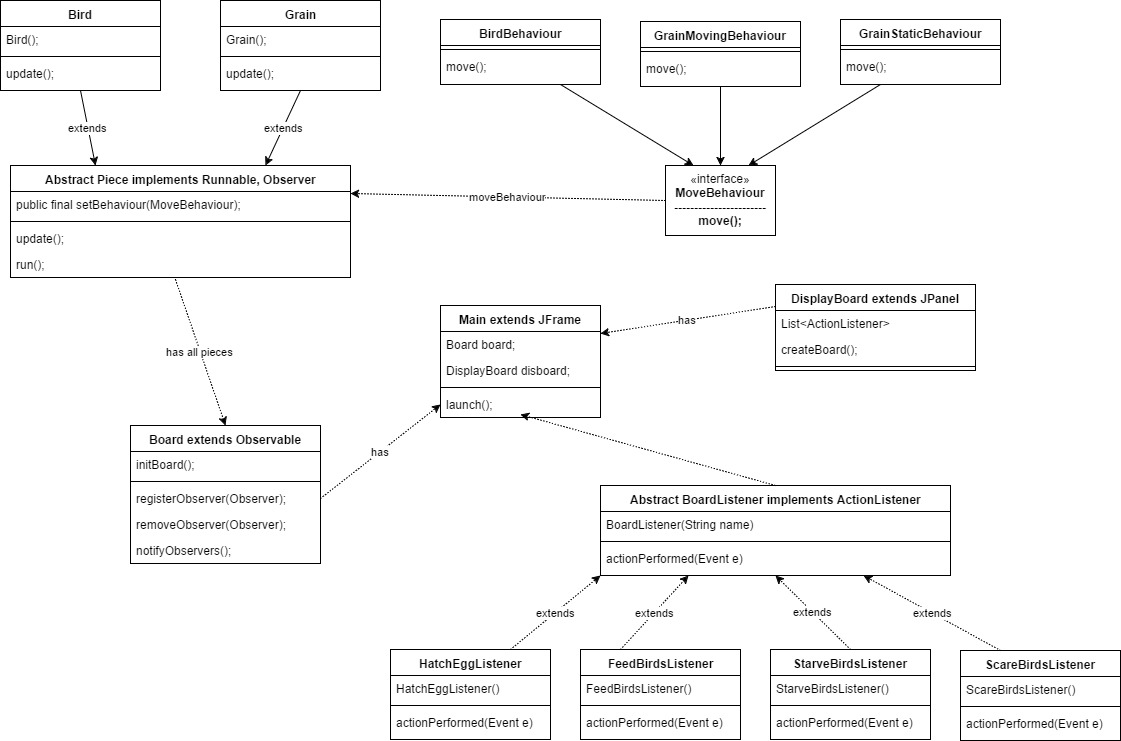
Task 1:

The current design is not appropriate. Apart from using a lot of inheritance which is bad for scalability as changes are not as easily propagated, Board also simultaneously extends Observable and implements Observer which is not good practice. The behaviour of Pieces is implemented entirely inside Board which increases complexity of Board and classes inheriting from Board. This means decreased cohesion.

The current design also makes Board responsible for the graphics while acting as a model to hold the board and all the pieces. This is not good practice either as these concerns should be separated.

Task 2:

We have decided to use Strategy design pattern to decrease the usage of inheritance. This allows the behaviour of a piece to be implemented inside a MoveBehaviour interface, and each Piece accepts a behaviour and moves accordingly. This means that adding new behaviour to a Bird or Grain requires only to add a new Behaviour class implementing MoveBehaviour. We have also used Observer/Observable pattern where each Piece observes the Board to know where the other Pieces are. In addition to that, we have decided to change the design in a way that would separate concerns thus we used MVC pattern.



Task 5:

We have decided to extend the behaviour of the simulation to multiply Birds when they feed on Grain. The Strategy design pattern allowed for easy extension since it only involved implementing new MoveBehaviour for Bird. Our design allowed us to add a new feature easily, but we would like to point out that when running GrainMovingBehaviour a race condition occurs demonstrated by NullPointerException. This happens because two Birds are accessing the same Grain and one of them removes it.