ASSIGNMNET 2

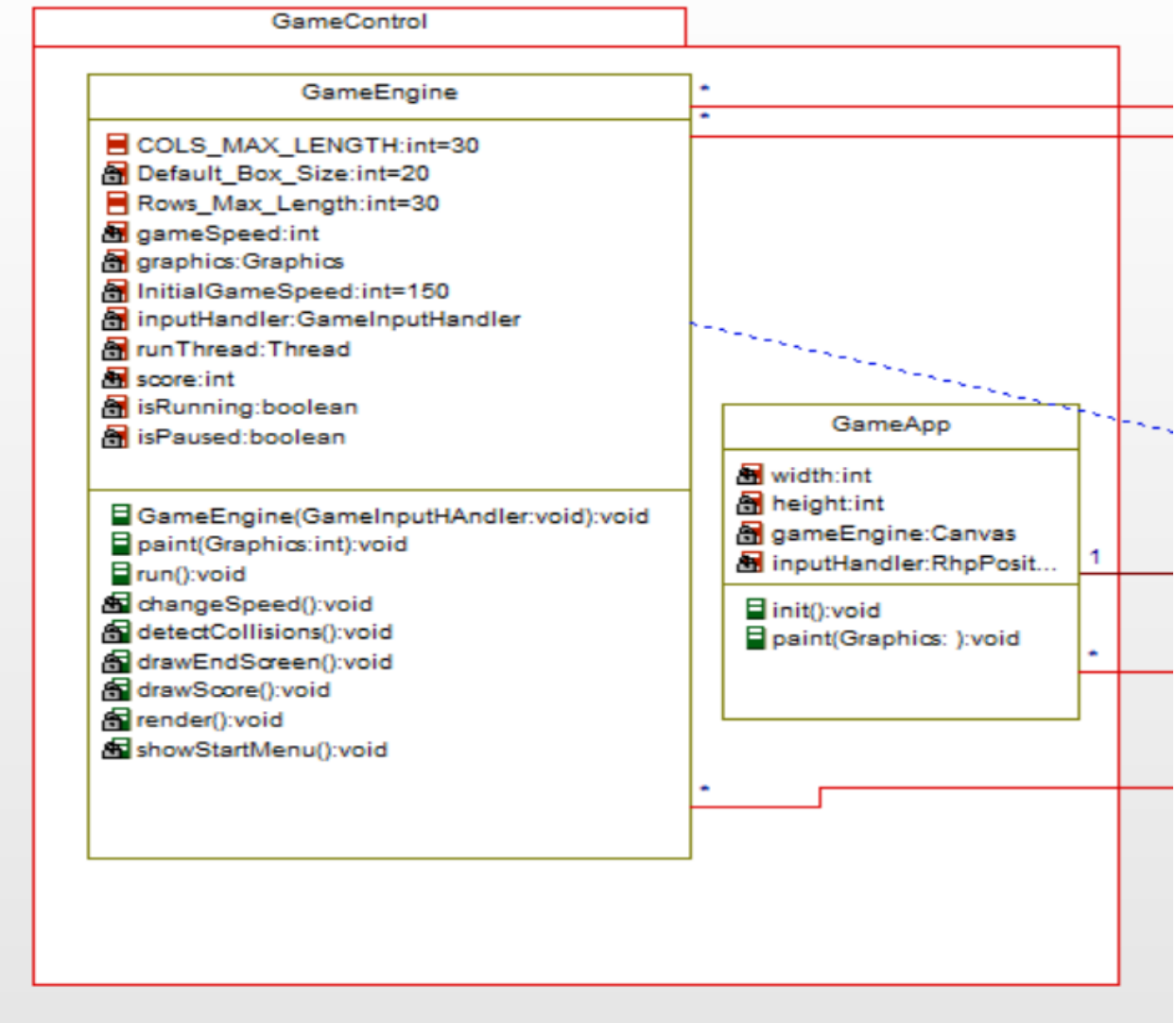
Task: To create class diagram for snake video game.

Tool Used: IBM Rational Modeler

Description:

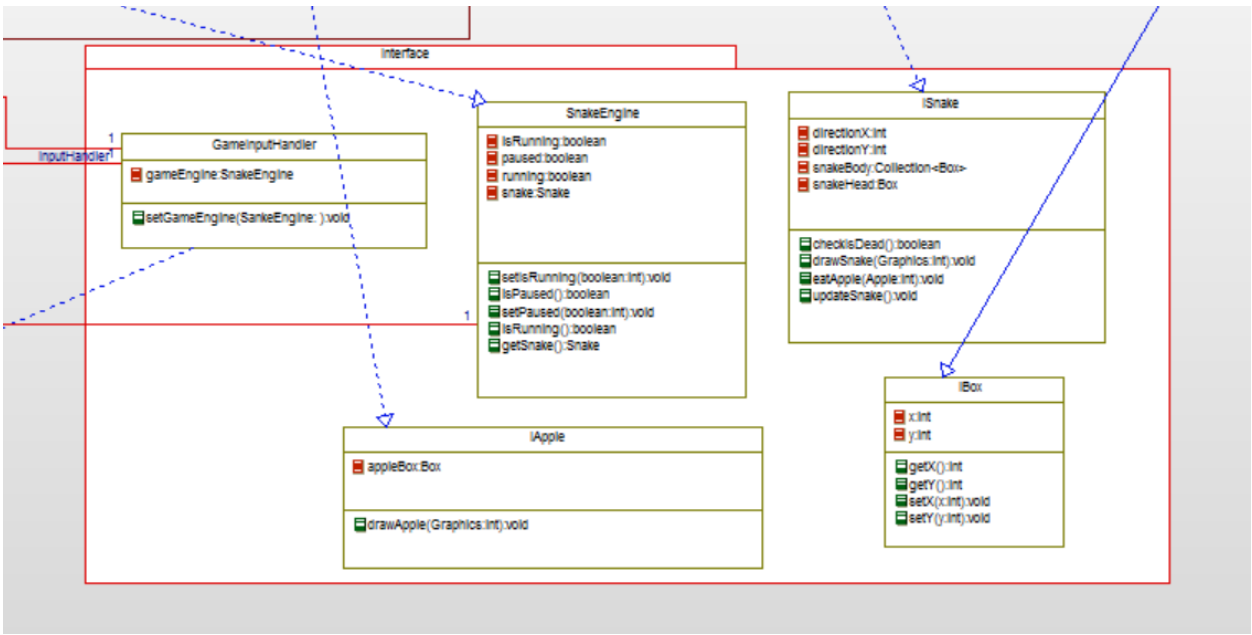
Packages included in the Class Diagram.

1. **GameControl**: It includes the canvas and graphical interfaces for the application.
    I. **GameEngine**: All the game functionalities like state of game i.e., pause, running are defined in this class. The default size of the screen, game speed and scoring functionalities are defined.
    II. **GameApp**: This class is used to define the width, height, and input Handler of the application.

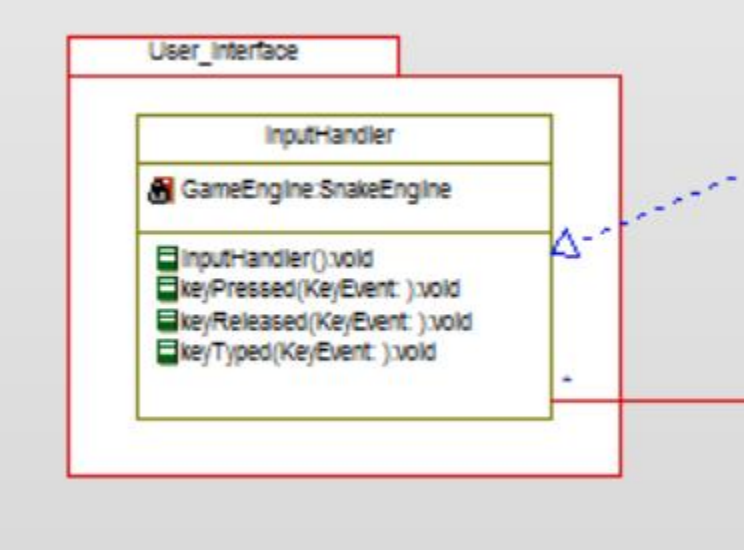


2. **Models**: It includes the classes used for modelling the application.
    I. **Apple**: It deals with the canvas.
    II. **Box**: It is used to define the box size and implements the class IBox.
    III. **Snake**: It includes the snake position, count.

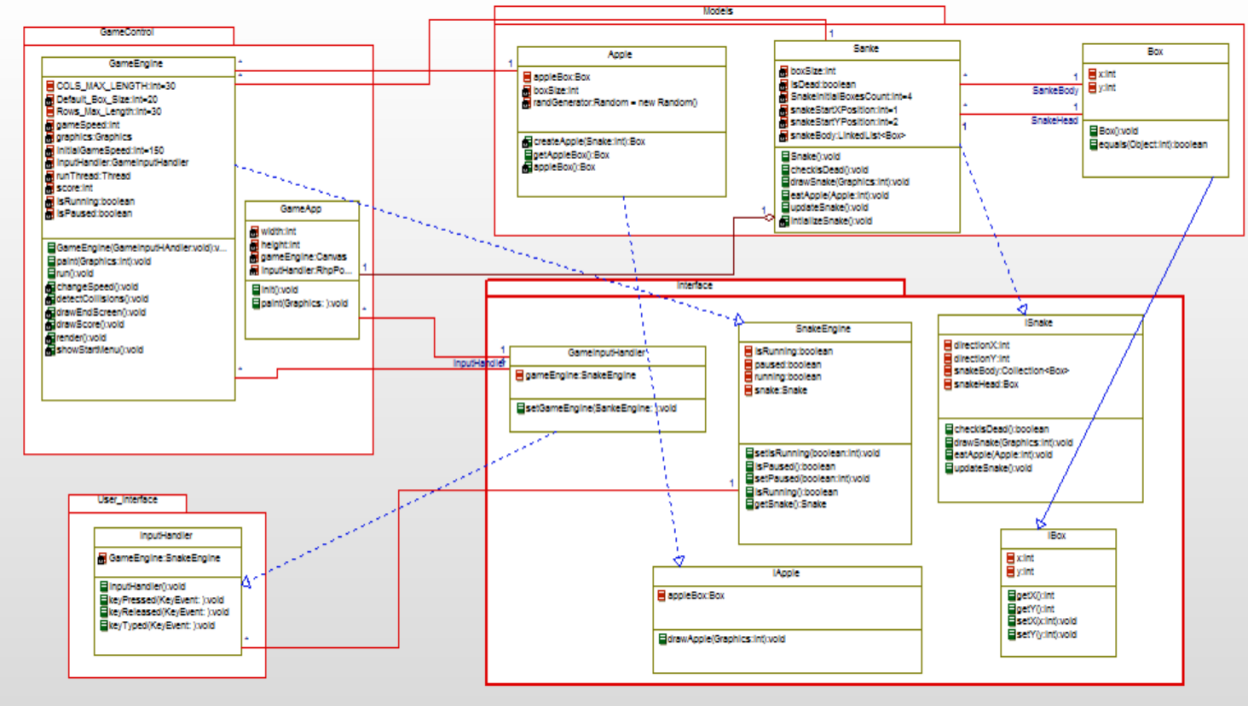
3. **Interface**: It includes the classes that are used as interfaces.
    I. **GameInputHandler**: It extends the GameEngine class.
    II. **IApple**: Interface for the canvas.
    III. **IBox**: Interface of the Box.
    IV. **ISnake**: Interface of the direction of snake.
    V. **SnakeEngine**: Interface for snake state engine.



4. **User Interface(UI):**
    I. InputHandler: Used to handle all the key inputs from user.



Complete Class Diagram:



Relationships Included:

1. Association: (GameApp -> GameInputHandler)
2. Association: (GameEngine -> Snake)
3. Association: (GameEngine -> Apple)
4. Association: (GameEngine -> GameInputHandler)
5. Association: (Snake -> Box)
6. Association: (Apple ->Box)
7. Realization: (IBox -> Box)
8. Realization: (ISnake -> Snake)
9. Realization: (IApple -> Apple)
10. Aggregation: (GameApp -> Snake)
11. Association: (InputHandler -> SnakeEngine)
12. Realization: (GameInputHandler -> InputHandler)
13. Association: (GameEngine -> GameInputHandler)
14. Realization: (SnakeEngine -> GameEngine)