/\*\*

The Game Play Engine has an active MVC (model, view, controller) architecture.

The purpose of this file is to create the model, view and controller components and their interfaces.

- also starts, plays, ends the game?

\*/

package gamePlayEngine;

import gamePlayEngine.controller.GameController;

import gamePlayEngine.controller.MessageType;

import gamePlayEngine.model.gamemodel.GameModel;

import gamePlayEngine.model.gamemodel.GameModelBoundary;

import gamePlayEngine.view.GameView;

/\*\*

\* Core GamePlayEngine class - Creates Model, View and Controller and sets up interaction amongst them.

\*/

public class GamePlayEngine {

/\*\* Main entry point of the GamePlayEngine - Sets up the game and starts it up.

\* @param args

\* @throws Exception

\*/

public static void main(String[] args) throws Exception {

if(args.length != 1) {

System.out.println("Please specify the game xml file");

return;

}

// Create the model

GameModel gameModel = GameModel.load(args[0]);

// Create the View

GameView view = new GameView();

// Create the Controller

GameController gameController = new GameController(gameModel, view);

// Tell the view the controller that will respond to user input.

view.setController(gameController);

// not sure why the controller is being called to start the game

- would expect to see sending the command to the view and having the view handle it - it asks the controller to do X - start, play, end (i.e., view handles the interface to the whole MVC game engine)

// Start the game

gameController.start();

// Play the game

gameController.play(MessageType.Internal, null);

// End the game

gameController.end();

}

}