/\*\*

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

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

\*/

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);

view.showStartView();

/\*\* First the view displays the user with a display to start the game. After the user enter the input then model takes care calling the act, scene, backdrop, screen and then the game element

Model takes the control and decides how the view needs to be displayed.

\*/

}

}