Introduction to Frameworks

In this recitation you will create a TicTacToe plug-in for a simple framework that facilitates the implementation of 2D grid games. To make grid game development straightforward, the game plug-ins must only implement the game's logic. Everything else (plug-in registration, player management, GUI implementation, etc.) is done by the framework.

The framework's core implementation is located in the `edu.cmu.cs.cs214.rec09.framework.core` package and provides you with a `Player` class, a `GamePlugin` interface, and a `GameFramework` interface defining the methods `Plugin`s can call on the framework. Grid game plug-ins must implement the `GamePlugin` interface in order to be registered with the framework. The `GamePlugin` interface contains several lifecycle methods that are called at various times throughout the period of a game (See Figure 1). It also has a few getter methods that the framework will call to obtain the name of the plug-in game, the width/height of the plug-in game's board, etc.

Two example plug-ins, "Rock-Paper-Scissors" (a classic grid-based game, seriously) and "Memory", are already implemented for you in the `edu.cmu.cs.cs214.rec09.plugin` package. You can see the behavior by running `Main.java` from Eclipse or `gradle run` from the command line.

Concepts

- Discuss similarities and differences shared between grid-based games. What common functionalities does the framework provide, and what decisions are left up to the plug-ins?
- Pair programming is a software development technique where two teammates collaboratively write software. At any point in time, one partner, the "driver", is using the computer, focusing on syntax and physically writing the code. The other partner, the "navigator", will guide the driver, focusing on higher level concepts. The two programmers switch roles frequently.

Instructions

- Read the given framework implementation to familiarize yourself with what it provides you.
- Read our sample plug-ins to see how you can implement your own plug-in.
- Read the documentation for the `GamePlugin` interface; in addition to Javadoc comments, we have provided a lifecycle diagram for framework plugins and several interaction diagrams in this document. For each lifecycle method, be sure you understand when and why it is called.
- Working with a partner using pair programming, implement a simple TicTacToe plugin in the `edu.cmu.cs.cs214.rec09.plugin` package. If you wish, you can reuse the implementation of TicTacToe from Recitation 08 that we provide in the `edu.cmu.cs.cs214.rec09.tictactoe` package. To register your plugin with the framework, add the fully-qualified class name of your plugin to the `edu.cmu.cs.cs214.rec09.framework.core.GamePlugin` file in the `src/main/resources/META-INF/services/` directory; the `java.util.ServiceLoader` will then use Java reflection to instantiate your plug-in and register it with the framework.

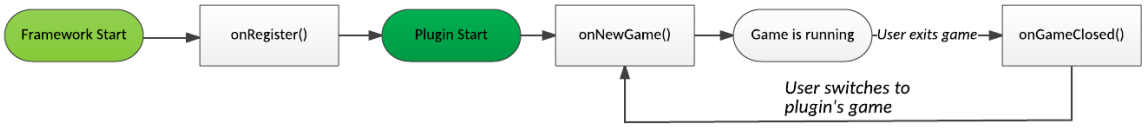
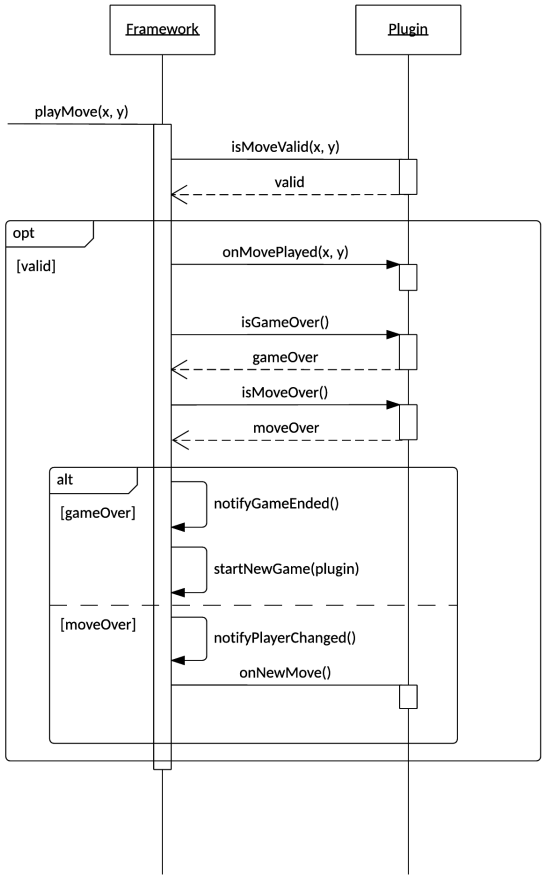
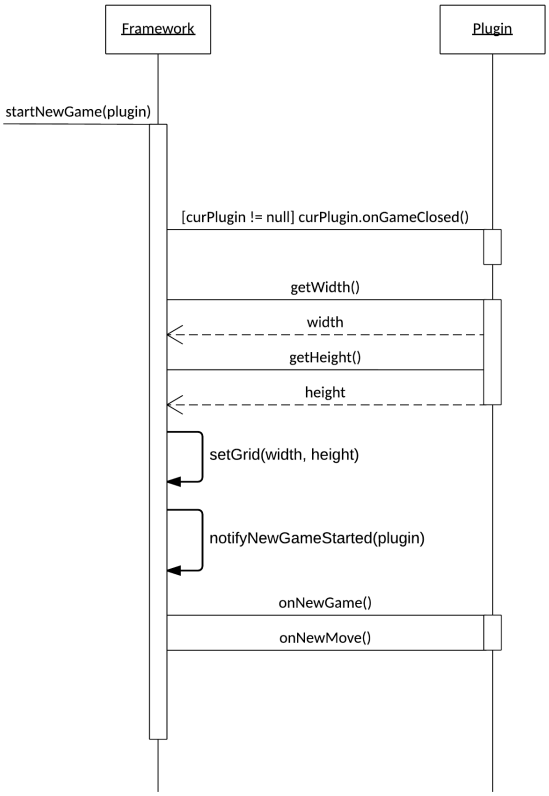


Figure 1: Framework Lifecycle



(a) Playing a Move



(b) Starting a New Game

Figure 2: Framework Sequence Diagrams