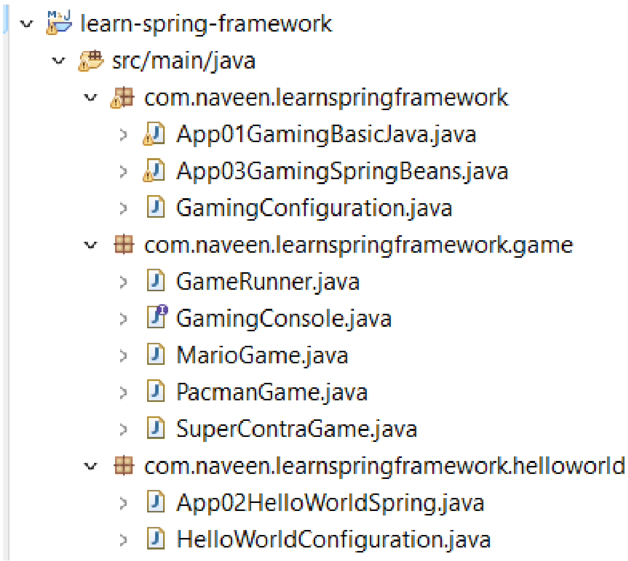
USING SPRING FRAMEWORK TO MANAGE BEANS FOR JAVA GAMING APP

PROJECT STRUCTURE:



```
learn-spring-framework
  src/main/java
    com.naveen.learnspringframework
      App01GamingBasicJava.java
      App03GamingSpringBeans.java
      GamingConfiguration.java
    com.naveen.learnspringframework.game
      GameRunner.java
      GamingConsole.java
      MarioGame.java
      PacmanGame.java
      SuperContraGame.java
    com.naveen.learnspringframework.helloworld
      App02HelloWorldSpring.java
      HelloWorldConfiguration.java
```

Already we have the Gaming class “App01GamingBasicJava”, but now we are going to make that using spring container.

1. Launch a **Spring Context**.
2. **Configure** the things that we want spring to manage.

GamingConfiguration.java

```java
package com.naveen.learnspringframework;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

import com.naveen.learnspringframework.game.GameRunner;
import com.naveen.learnspringframework.game.GamingConsole;
import com.naveen.learnspringframework.game.PacmanGame;

@Configuration
public class GamingConfiguration {
    @Bean
    public GamingConsole game() {
        var game = new PacmanGame();
        return game;
    }

    @Bean
    public GameRunner gameRunner(GamingConsole game) {
        var gameRunner = new GameRunner(game);
        return gameRunner;
    }
}
```

How do we pass game in GameRunner?

1. Directly **call** the game method, like

```
var gameRunner = new GameRunner(game());
```

2. To pass game as the **parameter**.

```
public GameRunner gameRunner(GamingConsole game) {
        var gameRunner = new GameRunner(game);
        return gameRunner;
}
```

In the above example, we followed this approach. What would Spring do is, **it will take a bean named "game", it will wire it to the gameRunner method.** It will pass game() bean as an argument to the method gameRunner(). When the gameRunner bean is created, the parameter "GamingConsole game" would be **get arguments from the game() bean**.

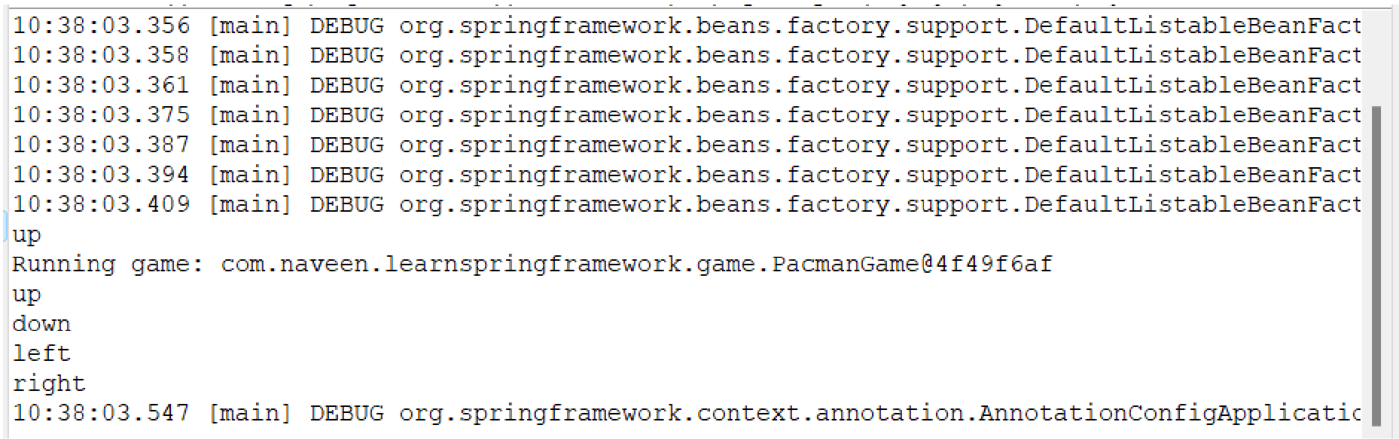
So, what we are doing here is, we are **creating a PacmanGame** and **wiring** it into the **gameRunner**.

App03GamingSpringBeans.java

```
package com.naveen.learnspringframework;

import
org.springframework.context.annotation.AnnotationConfigApplicationContext;

import com.naveen.learnspringframework.game.GameRunner;
import com.naveen.learnspringframework.game.GamingConsole;
import com.naveen.learnspringframework.game.MarioGame;
import com.naveen.learnspringframework.game.PacmanGame;
import com.naveen.learnspringframework.game.SuperContraGame;

public class App03GamingSpringBeans {

    public static void main(String[] args) {

        try(var context =
                new AnnotationConfigApplicationContext
                    (GamingConfiguration.class)){
            context.getBean(GamingConsole.class).up();

            context.getBean(GameRunner.class).run();
        }
    }
}
```

Here we are get the gameRunner bean and run it. We have launching up the game and gameRunner as Spring beans. So**, game and gameRunner are spring beans** and we are **picking the beans from our Spring context** and running them.

OUTPUT:



```
10:38:03.356 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFact
10:38:03.358 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFact
10:38:03.361 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFact
10:38:03.375 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFact
10:38:03.387 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFact
10:38:03.394 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFact
10:38:03.409 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFact
up
Running game: com.naveen.learnspringframework.game.PacmanGame@4f49f6af
up
down
left
right
10:38:03.547 [main] DEBUG org.springframework.context.annotation.AnnotationConfigApplicatic
```