Comet Pinball

Supplementary Specification

Team Members:
Patrick Haring
Christian Bürgi

Client:
Jean-Pierre Caillot

Revision hash: a65a530

Commit time: 2013-01-21 00:23:30 +0100

https://github.com/boskoop/comet-pinball/

Contents

List of Figures

1 Introduction

In Unified Process methodology, the document *Supplementary Specification* contains all the non-functional requirements. (The functional requirements are specified with *Use Cases*.)

Some additional non-functional requirements are currently covered in the Vision Document.

2 Non-functional requirements

2.1 Usability

U1 Avoid colors associated with common forms of color blindness.

U2 The interface should be as intuitive as possible.

U3 The key mapping should be customizable.

2.2 Reliability

R1 When the application is closed unexpectedly the data integrity should not be harmed.

2.3 Performance

P1 The application should run on entry-level computer with over 30 frames per second.

2.4 Supportability

S1 The system has to run on Unix/GNU-Linux as well as on Microsoft Windows and Mac OS X.

3 Technical specifications

T1 Java 6 is used.

T2 The cross-platform game development library libgdx is used.

3.1 Libraries

The following libraries have been used in the project:

Name	Version	Link
Apache Commons Lang	3.1	http://commons.apache.org/lang/
Apache Commons IO	2.4	http://commons.apache.org/io/
JUnit	4.10	http://junit.sourceforge.net
Libgdx	0.9.7	http://libgdx.badlogicgames.com
Logback	1.0.9	http://logback.qos.ch
Mockito	1.9.5	http://code.google.com/p/mockito/
PicoContainer	2.13.6	http://picocontainer.codehaus.org
SLF4J	1.7.2	http://slf4j.org
Sysout-over-slf4j	1.0.2	http://projects.lidalia.org.uk/sysout-over-slf4j/
Universal Tween Engine	6.3.3	http://www.aurelienribon.com

4 Tools

Name	Version	Usage	Link
Apache Maven	3.0.x	Build automation	http://maven.apache.org
Artifactory	2.6.4	Repository manager	http://www.jfrog.com
Astah community	6.6.4	UML diagrams	http://www.astah.net
Eclipse	4.2	Java development	http://www.eclipse.org
Git	1.8	Revision control	http://git-scm.com
Github	-	Collaboration	http://www.github.com
Jenkins	1.499	Build Server	http://jenkins-ci.org
LaTeX	TeX Live 2012	Documentation	http://www.tug.org/texlive/
Redmine	2.1	Project management	http://www.redmine.org

5 Development environment

In order to create a stable build process without integration problems, we use a Jenkins build server. The build is done using Apache Maven. The source is managed using git version control system. Figure 1 explains the setup we used for the process.

The developer writes his code on his workstation and checks it into the local git repository. The code is committed and pushed into the remote Github repository, where it is made available to every developer in the team. The Jenkins build server pulls on his scheduled builds the current version of the codebase into his local repository and builds. Maven installs the artefact into its build-server-local repository and deploys it to the Artifactory. The developers update their local Maven repository by downloading the jars from Artifactory.

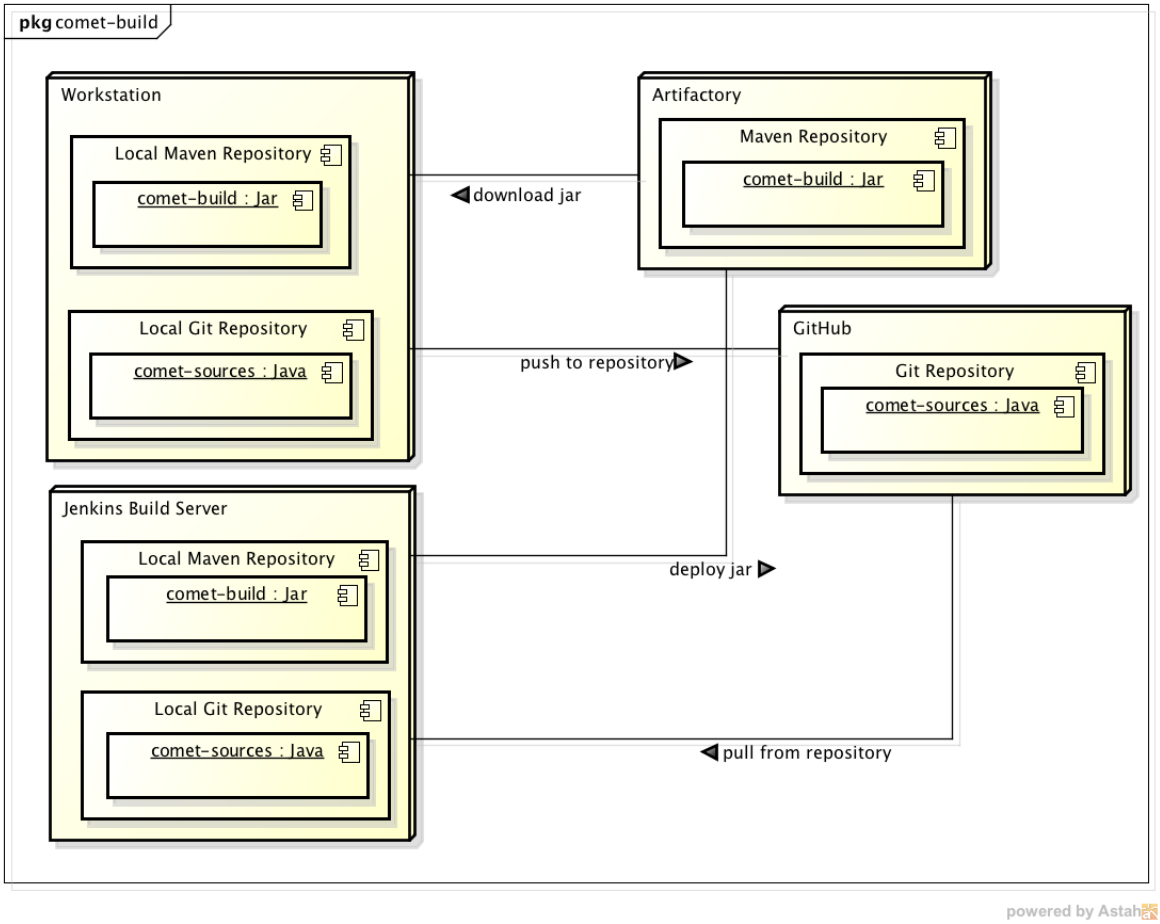

Figure 1: Comet build environment

5.1 Github configuration

The source repository is hosted at the following location: https://github.com/boskoop/comet-pinball/, a standard git installation is required to access the source.

5.2 Maven configuration

The developers need the following Maven configuration file in order to connect their Maven installation to the Artifactory server.

Listing 1: settings.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0 http://maven.
      apache.org/xsd/settings-1.0.0.xsd">
```

```
<pluginGroups>
</pluginGroups>
<proxies>
</proxies>
<servers>
  <server>
    <username><!-- username --></username>
    <password><!-- password --></password>
    <id>comet-artifactory</id>
  </server>
</servers>
<mirrors>
  <mirror>
    <mirrorOf>*</mirrorOf>
    <name>repo</name>
    <url>http://ci.m02.ch/artifactory/repo</url>
    <id>repo</id>
  </mirror>
</mirrors>
<profiles>
  <profile>
    <repositories>
      <repository>
        <snapshots>
          <enabled>false</enabled>
        </snapshots>
        <id>central</id>
        <name>libs-release</name>
        <url>http://ci.m02.ch/artifactory/libs-release</url>
      </repository>
      <repository>
        <snapshots />
        <id>snapshots</id>
        <name>libs-snapshot</name>
        <url>http://ci.m02.ch/artifactory/libs-snapshot</url>
      </repository>
    </repositories>
    <pluginRepositories>
      <pluginRepository>
        <snapshots>
          <enabled>false</enabled>
        </snapshots>
        <id>central</id>
        <name>plugins-release</name>
        <url>http://ci.m02.ch/artifactory/plugins-release</url>
      </pluginRepository>
      <pluginRepository>
        <snapshots />
        <id>snapshots</id>
        <name>plugins-snapshot</name>
        <url>http://ci.m02.ch/artifactory/plugins-snapshot</url>
      </pluginRepository>
```

```
        </pluginRepositories>
        <id>artifactory</id>
      </profile>
    </profiles>
    <activeProfiles>
      <activeProfile>artifactory</activeProfile>
    </activeProfiles>
</settings>
```