CSc 133 Lecture Notes

2 – Introduction to Mobile App Development and CN1

Computer Science Department

California State University, Sacramento

Overview

- Why to Use a Mobile Programming Environment?
- Why to choose Codename One (CN1)?
- CN1 Features
- CN1 vs Java
- CN1 Installation
- CN1 Hello World App
- CN1 and Assignments
- Assignment#0
- CSC 133 resources
- CN1 Online resources

Why to Use a Mobile Programming Environment?

- Mobile computing is **ubiquitous** and allows:
  - Instant retrieval of information
  - Constant communication
  - Easy access to games, company products etc.
- Hence, there is an **ever growing need** for mobile app developers.
- Also, knowing how to program in this contemporary environment is **fun** and **cool**!

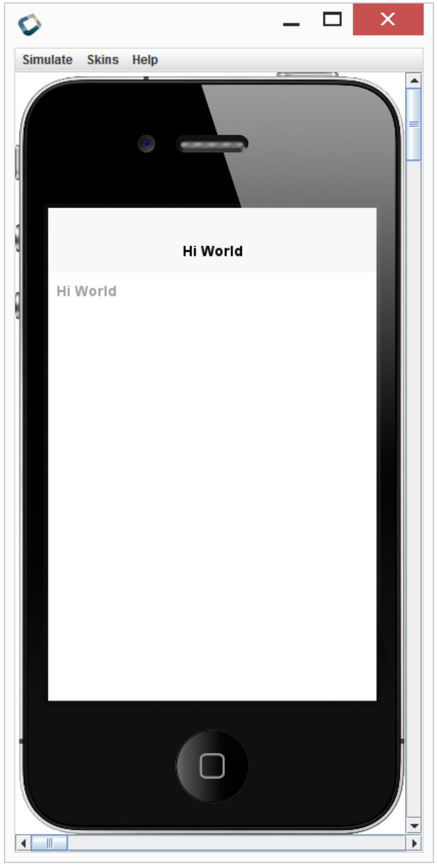
Why to Use a Mobile Programming Environment? (cont.)

- CSC 133 topics are widely applicable to a mobile programming environment.
- Hence, using this environment in the **lectures** and **assignments**, will help to:
    - Enhance learning by relating CSC 133 topics to their **contemporary use cases**
    - Provide a base for **further exploration** of this environment (apply it to other CSC topics or create your own brilliant app!)
    - Build a stronger **resume**

Why to choose Codename One (CN1)?

- There are various popular mobile programming environments:
  - Platform specific:
    - e.g. Android, iOS SDK
  - Cross-platform (write one program and run it on various platforms - iOS, Android, Windows, etc.):
    - e.g. Codename One (CN1), PhoneGap, Appcelerator, Xamarin
- We choose CN1, because it is:
  - Java-based
  - Cross-platform

CN1 Features



- Features we will use:
  - Free and open source
  - Have **simulator** environment

  (does not require you to have a mobile device)

- Features that we will **not** use:
  - Build and cloud **servers** (converts the CN1 code to a native app e.g. Android, iOS, Windows app)
  - **GUI builder** (provides drag and drop tools to automatically create GUI components)

CN1 vs Java

- CN1 API was initially limited to subset of Java 1.3 and then added support for subset of Java 5 and now supports some Java 8 language features.
- Does not support Java features that are not suitable for mobile devices e.g.:
  - Reflections
  - Desktop APIs such as java.net, java.io.File etc. (provides its own alternatives)
  - Swing library (provides Swing redesigned for mobile environment in its UI API/package)

CN1 Installation

- Install latest version of Java SE JDK (version 8):

http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

- CN1 can be installed to one of the following IDEs: Eclipse, NetBeans, or IntelliJ IDEA which run on various operating systems.
- For CSc-133, "Eclipse IDE for Java Developers - Neon version" is **required**:

http://www.eclipse.org/downloads/packages/eclipse-ide-java-developers/neonr

(Note: The Oxygen version works as well:

Get Eclipse OXYGEN

Install your favorite Eclipse packages.

- Windows is recommended.

CN1 Installation (cont.)

- Install CN1 plugin to Eclipse using instructions at:

http://www.codenameone.com/download.html

In Eclipse select “Help” → “Eclipse Marketplace” → search for “Codename One” and follow the installation process

- Alternative Eclipse installation steps:
    - Select “Help” → “Install New Software”.
    - Click the ”Add” button on the right side.
    - Name = ”Codename One” and location = https://codenameone.com/files/eclipse/site.xml .
    - Select the entries & follow the wizard to install

CN1 Installation (cont.)

- Eclipse and CN1 are installed at:
  - ECS Open Labs [RVR 2011, SCL 1234, SCL 1208 (24 hour lab)]
  - ECS Teaching Labs [ARC 1014/1015 (classroom instruction only labs)]
  - CSC Labs [RVR 1013/2005/2009/2013/5029]
  - ECS Windows Terminal Server (Hydra)

CN1 Hello World App

- Steps for Eclipse:
  - File → New → Project → Codename One Project
  - Give a project name “HiWorldPrj” and **uncheck “Java 8 project”**. Hit “Next”.
  - Give a main class name “Starter”, package name “com.mycompany.hi”, and select a “native” theme, and **“Hello World(Bare Bones)” template (for manual GUI building)**. Hit “Finish”.
- It generates and builds the project. You can view your main class under the package explorer:

HiWorldPrj → src → com.myCompany.hi → HiWorld.java

CN1 Hello World App (cont.)

- Run the app on the simulator in Eclipse by right clicking the last entry of the project under the package explorer:

  HiWorldPrj → Simulator_HiWorldPrj.launch

  - Select “Run As” to run and “Debug As” to debug your app.

- You can also run it directly from the command-line. Get into the HiWorldPrj directory and (in Windows) type:

  java -cp dist\HiWorldPrj.jar;JavaSE.jar
  com.codename1.impl.javase.Simulator
  com.mycompany.hi.HiWorld **(all in one line, but put spaces between sub-lines)**

CN1 Hello World App (cont.)

- Unix-like operating systems (such as Linux and Mac OS X) use “forward-slash” and “colon” (instead of “back-slash” and “semicolon”:

```
java -cp dist/HiWorldPrj.jar:JavaSE.jar
com.codename1.impl.javase.Simulator
com.mycompany.hi.HiWorld
```
(all in one line, but put spaces between sub-lines)

- You can switch through different skins in the simulator. For games, better to use a tablet skin (e.g., iPad, Download it via “Skins” → “More” -> “iPad III iOS 7”)

Troubleshooting Problems

- If dist\HiWorldPrj.jar is not generated:

set **JAVA_HOME** environment variable to JDK directory

In Windows: goto “Control Panel -> System -> Advance System Settings -> Environment Variables” and add JAVA_HOME as *C:\Program Files\Java\jdk1.8.0_102* (to “System Variables”)

- If the command line complains that:

‘java’ is not recognized … : add *C:\Program Files\Java\jdk1.8.0_102\bin* to **PATH**

JavaSE.jar cannot be found … : (first make sure you are in the project directory that has JavaSE.jar) add current directory (indicated by a single period “.”) to **CLASSPATH**

(see Appendices.pdf for tips)

CN1 and Assignments

- For each assignment create a different CN1 project.

- You **must** create all assignments in the same way as HiWorldPrj example:
  - uncheck “Java 8 project”, select “native” theme, and **“Hello World (Bare Bones)”** template.
  - change the project, main class, and package names…

CN1 and Assignments (cont.)

- For instance for Assignment#1:
  - Project Name: A1Prj
  - Main Class Name: Starter (keep the same for all assignments)
  - Package: com.mycompany.a1
- Main class has the following structure:

```
public class Starter {
    …
    public void init(…)  {…}  //init() is used when the App is launched (NOT resumed !).
    public void start()  {…}  //start() is used when application come on foreground (After
    init, or after being minimized)
    public void stop()  {…}  //stop() is used when application goes to background (ie
    minimized)
    public void destroy()  {}// destroy() is used when the OS kill the process
}
```

CN1 and Assignments (cont.)

- Solve the assignment by modifying **start()** in Starter.java (**do NOT delete other methods**) and adding/modifying other necessary source files.
- **Make sure** dist\A1Prj.jar is up to date (if not, in Eclipse, right click on dist directory and say “Refresh”)
- **Make sure** your submission works from command-line. Go into the A1Prj directory and type:

  ```
      java -cp dist\A1Prj.jar;JavaSE.jar
  com.codename1.impl.javase.Simulator
  com.mycompany.a1.Starter
  ```

  (all in one line, but put spaces between sub-lines)

CN1 and Assignments (cont.)

- Again, for Unix-like operating systems (such as Linux and Mac OS X) use “forward-slash” and “colon”:

```
java -cp dist/A1Prj.jar:JavaSE.jar
com.codename1.impl.javase.Simulator
com.mycompany.a1.Starter
```

(all in one line, but put spaces between sub-lines)

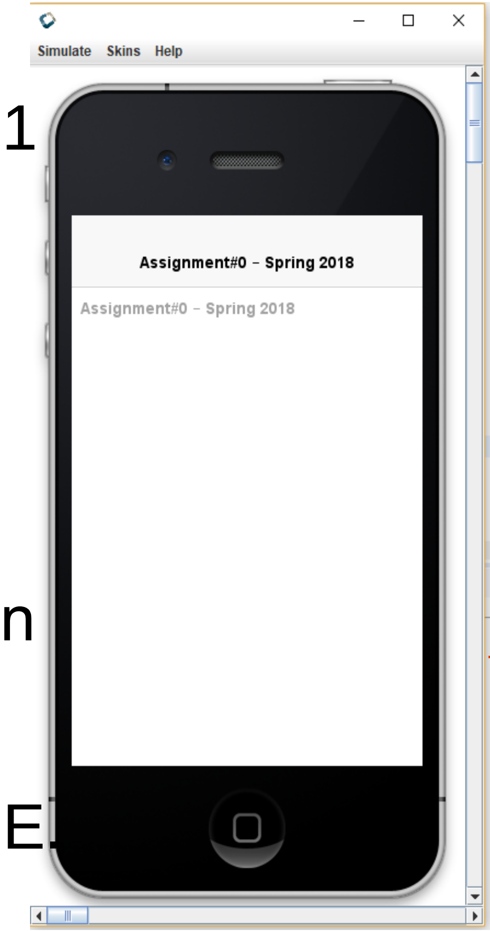
- Deliverables:

Zip A1Prj.jar (under *dist* dir) and entire *src* dir to a file called YourLastName-YourFirstName-a1.zip & submit this zip file to Canvas.

Assignment#0

- Find a lab computer that has CN1 or install CN1 to your computer.
- Following the instruction in the previous slides, generate an empty project called A0Prj.
- Modify Starter.java by replacing the texts “Hi World” with “Assignment#0 – Spring 2018”. Run the simulator.
- Experiments with debugging options of your IDE.
- Verify that your submission also works from the command line.

Do submit A0 via Canvas for Grading (its purpose is to make sure you have access to CN1 and ready to solve real assignments)

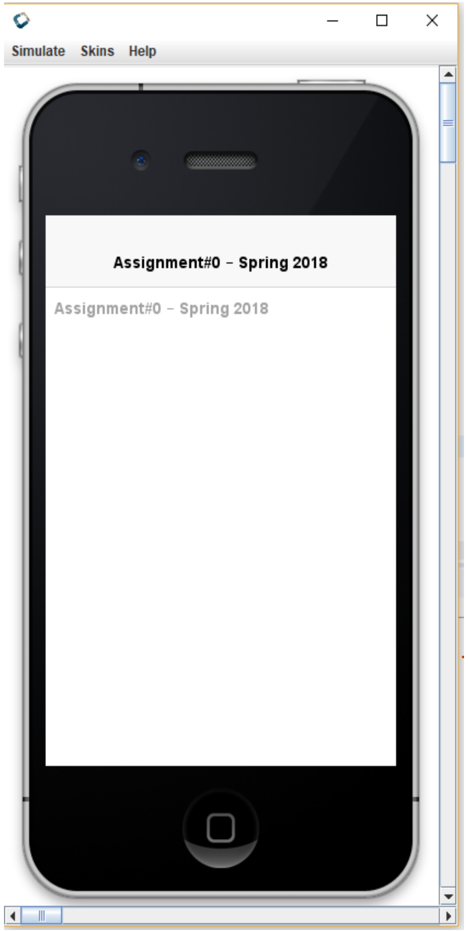


Assignment#0 – Spring 2018

Assignment#0 – Spring 2018

Demonstration

- Install CN1 Plugin.
- Create a CN1 Project and run a simulation.
- Run the CN1 app from command line.
- Debugging CN1 application.



Assignment#0 – Spring 2018

Assignment#0 – Spring 2018

Other CN1 Online resources

- Developers guide:

CN1 Developer Guide - Revision 3.6 (pdf is available at Canvas)

- Video tutorials can be found at:

http://www.codenameone.com/how-do-i.html

(note: mostly give examples that use the GUI builder which we will not utilize)

- You can view JavaDocs of APIs:

https://www.codenameone.com/javadoc/index.html