14. Individual Project 4, How to use Git 1
Pair Programming

Minhaeng Lee
Pair Programming

Pair programming (sometimes referred to as peer programming) is an agile software development technique in which two programmers work as a pair together on one workstation. One, the driver, writes code while the other, the observer, pointer or navigator, reviews each line of code as it is typed in. The two programmers switch roles frequently.

-wikipedia
Pair Programming

• **20** min each person (because we have only 120 min)

• If you
  – Have enough progress
  – or
  – Hate work with other

• Then you
  – Can work alone

• Otherwise
  – Do Pair programming
Individual Project : Tetris

• **Basic**
  – Get Block (and other) images
  – Generation 2D Map
  – Key input
  – **Thread based Timing Control**
    • Frame control
  – Game, Score, Next zone

• **Advanced**
  – Continue from previous
  – Item
  – Auto Play
Individual Goal : Tetris

• Oct 11
• Oct 18
• Oct 25
  – Mid check point
• Nov 1
• Nov 8
• Nov 15
  – HTML page drawing,
  – Final
Individual Project : Sneak Game

• **Basic**
  – Get Sneak Parts images
  – Generate 2D Map
  – Key input
  – **Thread based Time Control**
    • Frame control
  – Game, Score zone

• **Advanced**
  – Continue from previous
  – Item
  – Auto Play
Individual Goal : Sneak Game

• Done
  – Moving,

• Oct 11

• Oct 18

• Oct 25
  – Mid check point

• Nov 1

• Nov 8

• Nov 15
  – HTML page drawing,
  – Final
Individual Project: Music Player

• Basic
  – Get required button images (play, stop ... etc.)
  – CoverFlow (template provided)
    • ExampleGUI – CoverFlowDemo.java
  – File Scanning
  – File load/save
  – Play List Management
  – Music Play/Stop
    – exampleGUI - MP3Player.java
    – exampleGUI - WAVPlayer.java

• Advanced
  – Music Equalize
Individual Goal : Music Player

• Oct 11
  – Music Play Test
  – Check MP3Player.java, WAVPlayer.java

• Oct 18

• Oct 25
  – Mid check point

• Nov 1

• Nov 8

• Nov 15
  – HTML page drawing,
  – Final
Individual Project : Calendar

• **Basic**
  – Date control
    – `exampleSwing - DataExample.java`
  – JComponents
  – **Layout using MigLayout** (Example Provided)
    • ExampleGUI – `DashboardDemo.java`, `SwingDemo.java`
  – Event Add/Delete/Edit

• **Advanced**
  – Sync to the web
Individual Goal : Calendar

• Oct 11
  – Prepare : Date library
  – Goal : ?

• Oct 18

• Oct 25
  – Mid check point

• Nov 1

• Nov 8

• Nov 15
  – HTML page drawing,
  – Final
Individual Project : Flash cards

• Basic
  – Layout
  – File management (read/write)
  – Something need?

• Advanced
  – Fancy GUI
  – Online data management
Individual Goal : Flash Cards

• Oct 11
• Oct 18
• Oct 25
  – Mid check point
• Nov 1
• Nov 8
• Nov 15
  – HTML page drawing,
  – Final
Homework

• Record Your Progress during week
  – What Problem you have
  – What have you done
  – No progress, nothing to learn

• Read Layout Source
  – ExampleGUI – DashboardDemo.java
References

• Swing
  – http://docs.oracle.com/javase/tutorial/uiswing/

• Mp3 Player

• Double Buffering
  – http://msonic.tistory.com/1
  – https://docs.oracle.com/javase/tutorial/extra/fullscreen/doublebuf.html
GIT Toolkit Install

• Help — Eclipse Marketplace
Git Toolkit Install

- type “git” go
- install
  - Egit-Git Team Provider 4.1.0
Team Menu added
After toolkit installation.. you can see
• Make Git Repository
  • https://github.com/
  • Sign up
Create new git repository

Create a new repository
A repository contains all the files for your project, including the revision history.

Owner
mhlee1215

Repository name
HelloGit

Great repository names are short and memorable. Need inspiration? How about nefarious-octo-waffle.

Description (optional)
test

Public
Anyone can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None
Add a license: None

Create repository
Git Repository Detail

Quick setup — if you’ve done this kind of thing before

Set up in Desktop or HTTPS SSH https://github.com/mhlee1215/HelloGit.git

We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
  echo "# HelloGit" >> README.md
git init
  git add README.md
git commit -m "first commit"
git remote add origin https://github.com/mhlee1215/HelloGit.git
git push -u origin master
```

...or push an existing repository from the command line

```
  git remote add origin https://github.com/mhlee1215/HelloGit.git
git push -u origin master
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code
Clone repository for remote connection
Remote/local connection
Connect project to local repository
Share Project

Select the repository plug-in that will be used to share the selected project.

Select a repository type:

- CVS
- Git
- SVN
Select proper local repository
Do First Commit!
Do First Commit!
Push to remote server

Repository https://github.com/mhlee1215/HelloGit.git
Commit Result
Git System