13. Individual Project 3,
Double Buffering,
Pair Programming

Minhaeng Lee
Double Buffering

- For game developer

- **Blinking, when we are drawing animated object.**

  - **Reason**
    - Drawing to screen is relatively slow
    - Might be slower than frame-rate-per-second (FPS)
      - Think when we have 100 different objects in game screen

- Instead of Drawing directly,
- **Draw memory image first then copy the image to actual screen**
Double Buffering

1. Draw
   - graphics
   - Image
     - Back Buffer
   - Screen
     - Primary Surface

2. Blt (copy)
   - Image
     - Back Buffer
   - Screen
     - Primary Surface

DoubleBuffering.java
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.
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

• **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
  – 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)

- **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