Assignment 2

- Actions
  - Icons
  - Menu
- Internationalization (I18N)
- Progress
- Remember size & position of GUI
CVS

- Don’t att `dist` folder (or bin) to the CVS
javax.swing.Action

- Interface, contains
  - Accelerator, Mnemonic, Name, Icon, Description (short+long), Enabled?, command

javax.swing.AbstractAction
  - default implementations for the Action interface

- new JButton(new ExitAction());
- new JButton(new ExitAction(initValue));
Action – example 1

class MyAction extends AbstractAction {
    ImageIcon icon = new ImageIcon //TODO
    MyAction(String name) {
        putValue(Action.NAME, name);
        putValue(Action.SMALL_ICON, icon);
    }
}

Action playAction = new MyAction(“ply”);
JButton b = new JButton(playAction);
playAction.setEnabled(false);
Action – example

Action playAction = new AbstractAction("Play",
new ImageIcon("play.gif")) {
    public void actionPerformed(AE e) {
        move("left");
    }
};
JButton b = new JButton(playAction);
JMenuItem menuItem = commandMenu.add(playAction);
playAction.setEnabled(false);
Action

- The previous example is not good enough; we also need:
  - mnemonic, tooltip description, …
  - I18N
  - …
I18N

- See build.xml, target "run"

File: player/ui/lang.properties:
ui.ok = Ok
ui.cancel = Cancel

File: player/ui/lang_sv.properties:
ui.ok = Okej
ui.cancel = Avbryt

ResourceBundle rb =
  ResourceBundle.getBundle("player.ui.lang");
String okString = rb.getString("ui.ok");
JButton okButton = new JButton(okString);
java.util.ResourceBundle

- Be specific and clear! Like this;

  ```
  ui.menu.exit.name = Exit
  ui.menu.exit.mne = x
  ```

- Not like this:
  ```
  exit=Exit
  ```
Icons

URL u = Player.class.getResource("exit.gif");
JButton exitButton =
    new JButton(new ImageIcon(u));

- At compile time, exit.gif must be located next to Player.java
  - Ant will copy the icons to the build folder
- Icons also need to be in CVS
  - Location, next to source code
- Binary files need extra flag
  - cvs add –kb binary.gif