Java Puzzle Ball
Nick Ristuccia

Lesson 3-1
Implementing Inheritance Puzzles
Developing from Specifications

• Labs 1 & 2 ask you to write classes based on design specifications.

• Specifications are often recorded in a design document.
  – The design document explains what the program must do. In other words, what features the program must implement.
  – The design document shouldn't tell you how to implement these features. It's up to programmers to decide on the best implementation solution.
    • Designers and Marketing probably don't understand code.
    • Programmers could record their implementation plans in a separate document.

• The order you implement features is based on a production schedule.
  – Let's go behind the scenes of Java Puzzle Ball again to examine this...
# Production Schedule

<table>
<thead>
<tr>
<th>Notable Build</th>
<th>Sprint</th>
<th>Sprint Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>August 16, 2013</td>
<td>First Runnable</td>
<td>Just get stuff working in JavaFX</td>
</tr>
<tr>
<td>August 22, 2013</td>
<td>First Playable</td>
<td>Allow bumpers to be designed by players</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The ball bounces</td>
</tr>
<tr>
<td>September 27, 2013</td>
<td>Alpha</td>
<td>Add more behaviors</td>
</tr>
<tr>
<td>October 16, 2013</td>
<td>Alpha</td>
<td>Most core features are implemented</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This includes Inheritance puzzles</td>
</tr>
<tr>
<td>November 21, 2013</td>
<td>Beta</td>
<td>Most features are implemented</td>
</tr>
<tr>
<td>December 31, 2013</td>
<td>Polish</td>
<td>Fix Bugs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Make things prettier</td>
</tr>
</tbody>
</table>
October 16, 2013

- We created additional game modes (Inheritance & Geometry Test).
  - Inheritance was planned, Geometry Test was incidental.

- There is a pop-up for choosing levels.
  - Because we didn't know how to unload/swap between levels.
  - You have to close the program to load a different level.
  - Levels are for testing features, and aren't quite puzzles for players.

- More notable features:
  - Level geometry
  - A GreenBumper and GreenWheel
  - Level-building instructions are read from a text file
November 21, 2013

• Over one month later:
  – We figured out how to unload levels!

• Use the Options button to choose levels.
  – It's a temporary solution until we learned to create menus.
  – Levels are actual puzzles instead of tech demos.

• More notable features:
  – Fancy new background art.
  – More levels.
  – Slots are labeled ABCD instead of NESW (People thought their solutions were wrong if the N slot didn’t face north.)
Both Designing and Programming are Iterative

• Periodically tested the application. Made changes based on the results.
  – See where users get confused.
  – See where performance is slow.

• Because design specifications change, your documents need to be somewhere that's accessible for viewing and editing by the team.

• Storing documents in version control is one option.
  – Oracle accommodates a version control solution for developers through Oracle Developer Cloud Service.

• Alternatively, store documents in a content management system.
  – Oracle accommodates this through OraDocs, or Oracle Content and Experience Cloud.
Why Label the Slots?

• There's a very specific reason why we needed to label the slots.
  – This will make sense with Inheritance Puzzles.

• Don't understand what inheritance is? That's ok. The game will teach you.
  – Students who have never programmed before are able to come up with a fairly close definition for inheritance.
Exercise 3

• Play **Inheritance Puzzles 1 through 3.**

• Consider the following:
  – What is inheritance?
  – Why are these considered "Inheritance" puzzles?

4 & 5 too, if you want to play more inheritance puzzles