Piano Hero

A Learning Game for my Final Project

Solo Team: Daniel Mendelsohn
Teaching kids an instrument is hard
Games help kids learn

Math Blaster

Mavis Beacon Teaches Typing
Music games are commercially successful
Moving away from a “toy” controller
A block diagram for a commercial product
MIDI is a sensible lightweight protocol

<table>
<thead>
<tr>
<th>Command</th>
<th>Meaning</th>
<th>#Parameters</th>
<th>Param1</th>
<th>Param2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x80</td>
<td>Note Off</td>
<td>2</td>
<td>key</td>
<td>velocity</td>
</tr>
<tr>
<td>0x90</td>
<td>Note On</td>
<td>2</td>
<td>key</td>
<td>velocity</td>
</tr>
<tr>
<td>0xA0</td>
<td>Aftertouch</td>
<td>2</td>
<td>key</td>
<td>touch</td>
</tr>
<tr>
<td>0xB0</td>
<td>Continuous controller</td>
<td>2</td>
<td>controller#</td>
<td>value</td>
</tr>
<tr>
<td>0xC0</td>
<td>Patch change</td>
<td>2</td>
<td>instrument#</td>
<td>value</td>
</tr>
<tr>
<td>0xD0</td>
<td>Channel Pressure</td>
<td>1</td>
<td>pressure</td>
<td></td>
</tr>
<tr>
<td>0xE0</td>
<td>Pitch bend</td>
<td>2</td>
<td>lsb(7bits)</td>
<td>msb(7bits)</td>
</tr>
<tr>
<td>0xF0</td>
<td>(non-musical commands)</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
A block diagram for a commercial product
The game logic block reads data controls game state

• Data stored in a lightweight format in RAM
• Internal “atomic” clock for smallest possible time subdivision (~50ms)
• Passes relevant information to specialized blocks
A block diagram for a commercial product.
Action interpreter interfaces between game logic and parsed user input

• Gets input from game logic – what notes should be played and when
• Gets input from MIDI interpreter – what notes user is playing and when
• Determines if user action matches the correct action based on internal game data
• Lots of little design decisions w/ regards to input
A block diagram for a commercial product

Serial MIDI input from keyboard → MIDI Interpretation Module → Keys pressed → Action Interpretation Module

Input events → Nearby notes

RGB output to VGA module → Display Module → Upcoming Notes → Game Logic Module
A display module drives the VGA output
Insight: reusable sub-modules

- Example: “NxM” buffer
- A NxM buffer takes in blocks of size M as input.
- It can hold N such blocks
- An input signal triggers data clearing
- An output signal indicates data overflow
Timeline

• Week of 11/18: Build a working minimum viable product for each block
• Week of 11/25: Debug and adjust the individual blocks
• Week of 12/2: Integration and user testing
• If there is time: Fun augmentations. For example, a “learning” mode in which you can play a tune and thus “create your own level”