Learning Games by Demonstration

Rahul Banerjee
Brandon Holt
CSE 573 – Au 2012
Motivation

- Build simple 2D games without writing a single line of code.
- Programming by demonstration
  - Users demonstrate how objects behave
  - System infers the game logic
    - Use knowledge of how 2D games work
    - Emphasize more general solutions
  - User refines behaviors with more examples
Related Work

  - SMARTedit (repetitive text-editing tasks)

- **Spreadsheet Data Manipulation Using Examples.** S. Gulwani, W. R. Harris, R. Singh. *(CACM 2012, POPL 2011, PLDI 2011)*
  - (excel macros)

- **Synthesizing Geometry Construction.** S. Gulwani, V. A. Korthikanti, A. Tiwari. *(PLDI 2011)*
  - (drawing tool)

- (and many more…)
Learning Games

Implicit Game Loop:

```python
while (True):
    T = detectTriggers(gameState)
    for (trigger, objs) in T:
        action = actionsTable[trigger]
        action(objs)
```

actionsTable:

<table>
<thead>
<tr>
<th>Trigger</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>keypress&lt;UP_KEY&gt;</td>
<td>move(paddle, 0, 1)</td>
</tr>
<tr>
<td>init&lt;Ball&gt;</td>
<td>repeat(move(ball, 1, 1))</td>
</tr>
<tr>
<td>collision&lt;Ball, Paddle&gt;</td>
<td>bounce(ball, X)</td>
</tr>
<tr>
<td>collision&lt;Ball, Wall&gt;</td>
<td>bounce(ball, collision.direction)</td>
</tr>
</tbody>
</table>

GameObject

- Wall
- Ball
- Paddle
- ...

Trigger

- init<obj>
- keypress<key>
- collision<obj1,obj2>
- ...

Action

- replace
- bounce
- stop
- repeat
- delete
- ...
Implementation/Evaluation

- Simple prototype
  - Written in Python, ugly Qt GUI
  - Objects/movements constrained to tiles
  - Learning by “demand-paging”

- Evaluation
  - Metric: number of demonstrations needed to learn game
  - Plan to evaluate on Pong, Pacman, and some other game
Demonstration
(no pun intended)
Future Work

- World builder
  - Actually make your own game
  - Allow object spawning as Action
- Continuous space
  - Smoothing
- Timeline scrubbing
  - Rewind and demonstrate what “should have happened”
- Physics
  - Enable “platformer” with gravity
  - Pick most “physically accurate”