Final Project Initial Presentation

by Brien Colwell
interactive poetry in Pad++

And searched, I searched,

That's waiting where the rainbow ends,

I went to find the pot of gold,

- The way the view moves enhances the story – e.g. indirect when searching
attempt to move a square in Piccolo (UMD)

```java
Pnode node = ... 
node.set ... initial position, initial scale

node.animateToPositionScaleRotation(200, 100, 1, _PI )
node.animateToPositionScaleRotation(300, 300, 2, _PI )
node.animateToPositionScaleRotation(0, 300, 1, _2_PI )
```
attempt to move a square in Piccolo

- What happens in between key points is unexpected
when it gets more artistic

( lots of framing code )

for u in 0..1

node.setX( x0 + dx * u + r * cos( _2_PI * u ) )
node.setY( y0 + dy * u + r * sin( _2_PI * u ) )

- not immediately clear this is it
- not immediately clear what changing a parameter – e.g. r – will look like
- long development cycle
- investigate interactive ways to create paths – position, size, rotation
- investigate ways to visualize the “in between”
- apply results to Piccolo ... to move that square
background: Maya (Alias)

- (motion path) Movement of an object over time shown
background: space scale diagram (Bederson and Furnas)

- Gives a 1D+Scale view of the movement
background: interactive is good

- Real time systems – Artistic Resizing (Dragicevic et al), Artistic Multi projection Rendering (Agrawala et al)

- Media authoring – Alan Kay's Squeak:
  create a car, see how it moves, graphically change steering (control not shown), immediately see how the car responds
background: where the camera fits in

- Camera can be treated the same as scene elements w.r.t. movement, rotation, scale

- for example, In Piccolo, PCamera is a PNode
approach: key frame interpolation

- Key frames are in the scene
- Manipulate key frames
- Immediately see how the in between changes
visual encodings

- How to visually encode speed, rotation, scale, and position between key frames
paths

- Represent paths as parametric curves + operations
- Use pacing functions to get speed effects (Hudson and Stasko)
- Interactively create a path, export it to XML, a custom PActivity reads the XML and moves something along the path