

Quick and Dirty: 2 Lightweight AI Architectures

Kevin Dill
Rez Graham

The Mars Game

- A team of rovers crash-landed on Mars...
- Level-based puzzle game (think Cut the Rope or Where's My Water)
- Each level is "solved" with math or programming
- "Indy-sized" project



AI Design

- We don't need much AI
- We do need a way to trigger events
 - Level success / failure
 - Sound effects & dialog
 - UI manipulation
 - Special effects
 - ...
- Rapid development & iteration is essential!
 - Short time scale
 - Experimental game design

Aside: Vocabulary

- Level, Scenario, Task, Mission => all the same thing

Anatomy of a Trigger

- Triggers contain:
 - A single Boolean **trigger condition**
 - Can be multiple **clauses** and-ed or or-ed together
 - A list of **actions**
- When the Boolean clause becomes true, the trigger fires
- When the trigger fires, we execute each action (in order)
- Event based
 - (unlike most architectures)
 - no polling or update cycle
 - (mostly)

A Simple Trigger

Play a hint after 20 seconds, but only once

playHint_1_8_moveCamera:

triggerCondition:

- and:
 - delay:
 - 20
 - doOnce:

actions:

- playSound:
 - ALV036_Rover

Gotchas, Bells, & Whistles

- Modular design
 - Deep dive tomorrow!
- Clauses are event handlers
- Event persistence
 - Tunable timing threshold (default: 667 ms)
 - Resets if the trigger fires or the scenario resets (succeeds or fails)
- Trigger groups
 - Say you have 3 different reasons that you failed... and one you succeeded
 - ~~Carefully craft the trigger condition~~ **<= too brittle!!**
 - Give each a (fixed) priority, take the highest
- Global triggers (e.g. fail on collision, fail on program completion)
- Level initialization => just a bunch of actions

A Complex Trigger

```
# Trigger success when the program  
# finishes if they drew the correct  
# triangle and are at the correct  
# position.
```

```
succeedOnTriangle_4_5:
```

```
  group: successOrFailure
```

```
  priority: 1
```

```
  triggerCondition:
```

- and:
 - onBlocklyStopped:
 - Perry
 - onBlocklyPolygon:
 - Perry
 - [[0, 0], [0, 2], [-2, 0]]
 - isAtPosition:
 - Perry
 - [-27, 19]

```
  actions:
```

- buildBaseComponent:
 - solarPanel2
- delay:
 - 4
- playSound:
 - musicSuccessShort
- clearBlockly:
- scenarioSuccess:

Parting Thoughts

- Trigger systems are really useful & not hard to build
 - But there are a few tricky bits – hopefully this helped
- Modular AI is awesome!
 - Just do it!
 - Come to the talk tomorrow to find out how
- All of the code & assets are open source
 - Apache 2 license
 - On Github: <https://github.com/virtual-world-framework/mars-game>
 - (or just Google “github mars game”)
 - We would love to find a partner looking to take the game farther!

Quick and Dirty: 2 Lightweight AI Architectures

Kevin Dill
Rez Graham

The Mars Game

- Research project
 - Hypothesis: a "true" game will be more engaging, and also more effective, than "traditional" online learning software
 - The game needed to be both educational & fun
- Small team, short time span
 - ~1.5 years, ~4 people
 - Built from scratch, in JavaScript

