

MappyLand: Fast, Accurate Mapping for Console Games

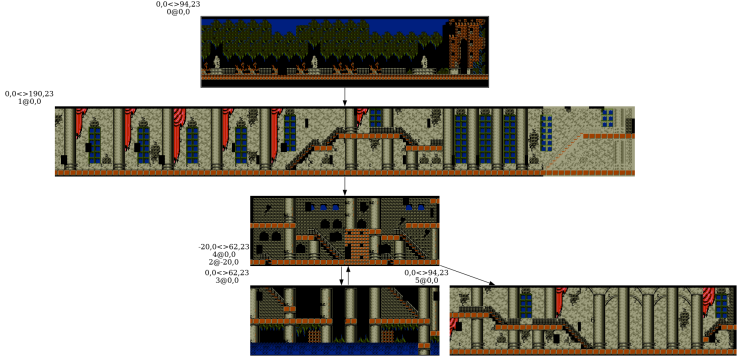
Joseph C. Osborn, Adam Summerville, Nathan Dailey,
Soksamnang Lim

September 28, 2021

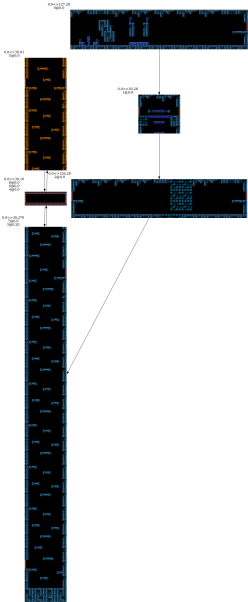
Overview

- ▶ Action videogames have common structures
 - ▶ Character moving in space
 - ▶ Between discrete rooms
 - ▶ Among other characters
- ▶ MappyLand interprets live game play to discover these structures
 - ▶ With an overhead of one millisecond per frame
- ▶ New opportunities for game-aware AI

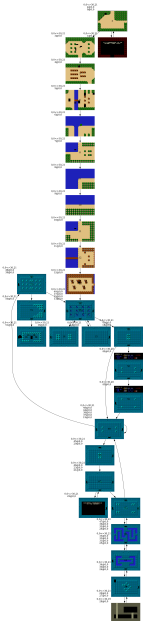
Example



Example



Example



This Talk

- ▶ What Mappy
- ▶ How Mappy
- ▶ Why Mappy

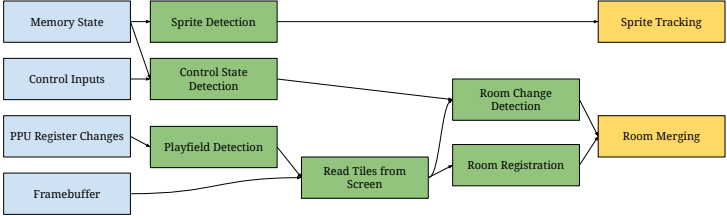
A Few Years Back

- ▶ We showed Mappy's predecessor at PCG'17
 - ▶ Batch-mode only
 - ▶ Not fast
 - ▶ Room merging by visual similarity
- ▶ Then we finished grad school and got jobs

Mappy → MappyLand

- ▶ Last year we Rewrote It In Rust
- ▶ On-line
 - ▶ Recorded or live play traces
 - ▶ (So it could fit into any AI pipeline)
- ▶ 1000x faster, vastly less RAM usage
- ▶ Merges rooms by tile similarity measure

Pipeline



What do we learn?

- ▶ What are the contents of rooms?
- ▶ Which tiles can turn into which other tiles?
- ▶ Which objects start where in the rooms?
- ▶ Which rooms are connected to which other rooms?
- ▶ What parts of the world have not yet been explored?

Limitations

- ▶ NES only (for now)
 - ▶ Working on CV models
- ▶ 2D, flat maps only
 - ▶ No layers, no parallax
 - ▶ Game Boy OK, Super NES not so much

Instrumentation

- ▶ Running an emulator means we can...
 - ▶ Read scroll register changes
 - ▶ Read sprite locations from OAM
 - ▶ Speculatively execute different input sequences

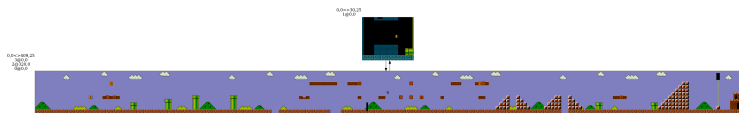
Implementation

- ▶ The screen is a grid of Tile IDs
 - ▶ Plus an "unobserved" tile ID 0
- ▶ Maps are grids of Tile Changes
 - ▶ "Observed a change from ID A to ID B at (x,y)"
 - ▶ Maps start out with $0 \rightarrow 0$
- ▶ We remember how often a Tile ID transitions to another
 - ▶ We remember how often a Tile Change is swapped with a successor Tile Change
- ▶ Compact, memory-efficient representation

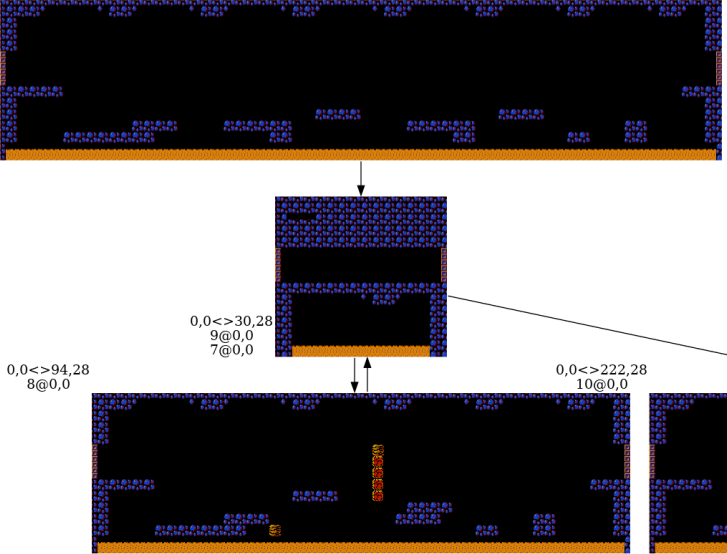
Implementation

- ▶ The tricky new part: Room merging
- ▶ We often revisit the same room in a game
 - ▶ In the same playthrough or in multiple plays
 - ▶ Maybe we just see pieces of a room
- ▶ Must align and match the pieces
 - ▶ Currently use a soft template matching algorithm
- ▶ Some essential ambiguities in non-Euclidean spaces
 - ▶ Same room, or confusing maze?

Example



Example



Good for AI players

- ▶ Feature extraction
 - ▶ Tiles over pixels
 - ▶ Sprites
- ▶ Combine with e.g. CHARDA for richer data
- ▶ High-level planning (maps!)
- ▶ Measure for novelty

Good for AI research

- ▶ More levels in corpora
- ▶ Pull more data from play traces
 - ▶ Combine data from several traces
- ▶ Automated exploration

Good for human players

- ▶ E.g. speed-runners, randomizers
- ▶ Add mapping feature to old games
- ▶ Accessibility (Aytemiz et al.)