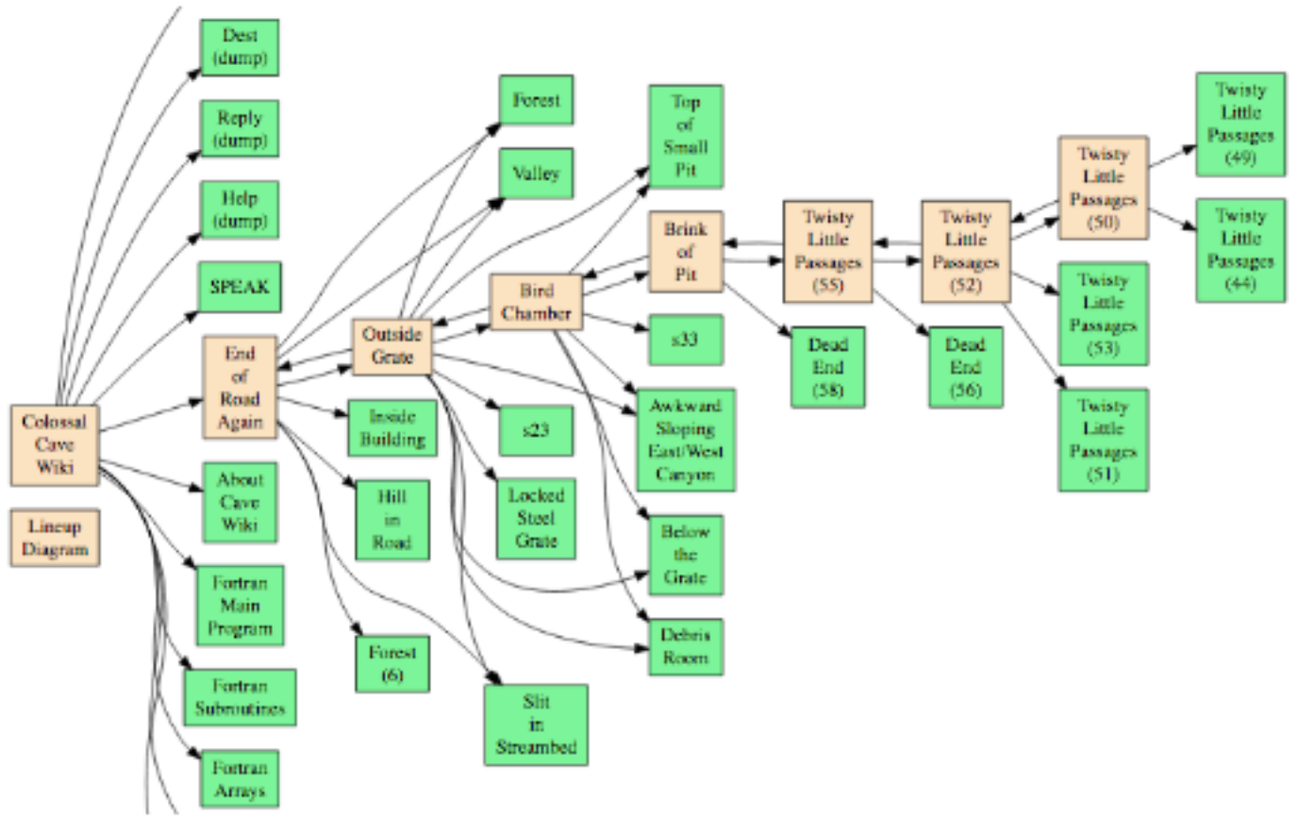
I translated the original Colossal Cave Adventure to wiki as an experiment to view the game as the database the author saw.

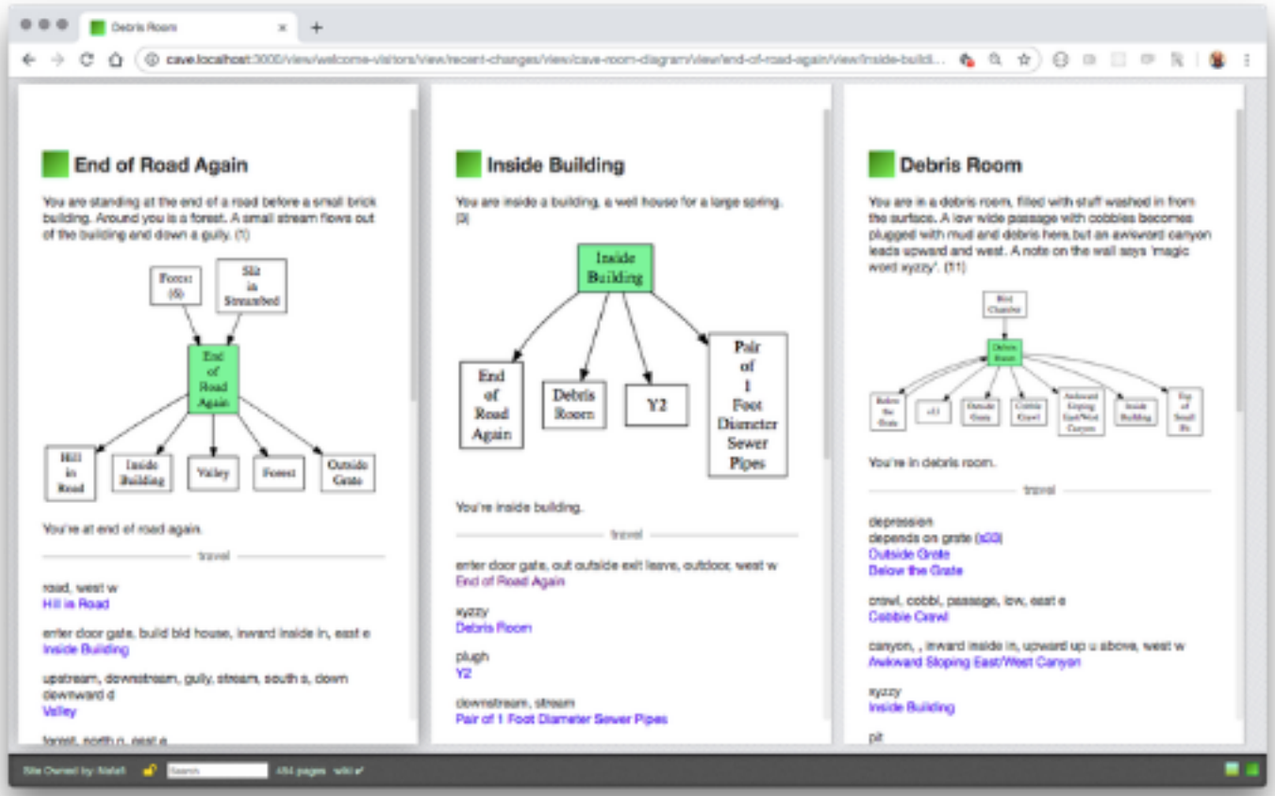
Now as wiki improves I return to it and other sites to feel my improvements working.

Here is one post I made last week with the then newest code. Now I am letting the cave wiki suggest further improvements

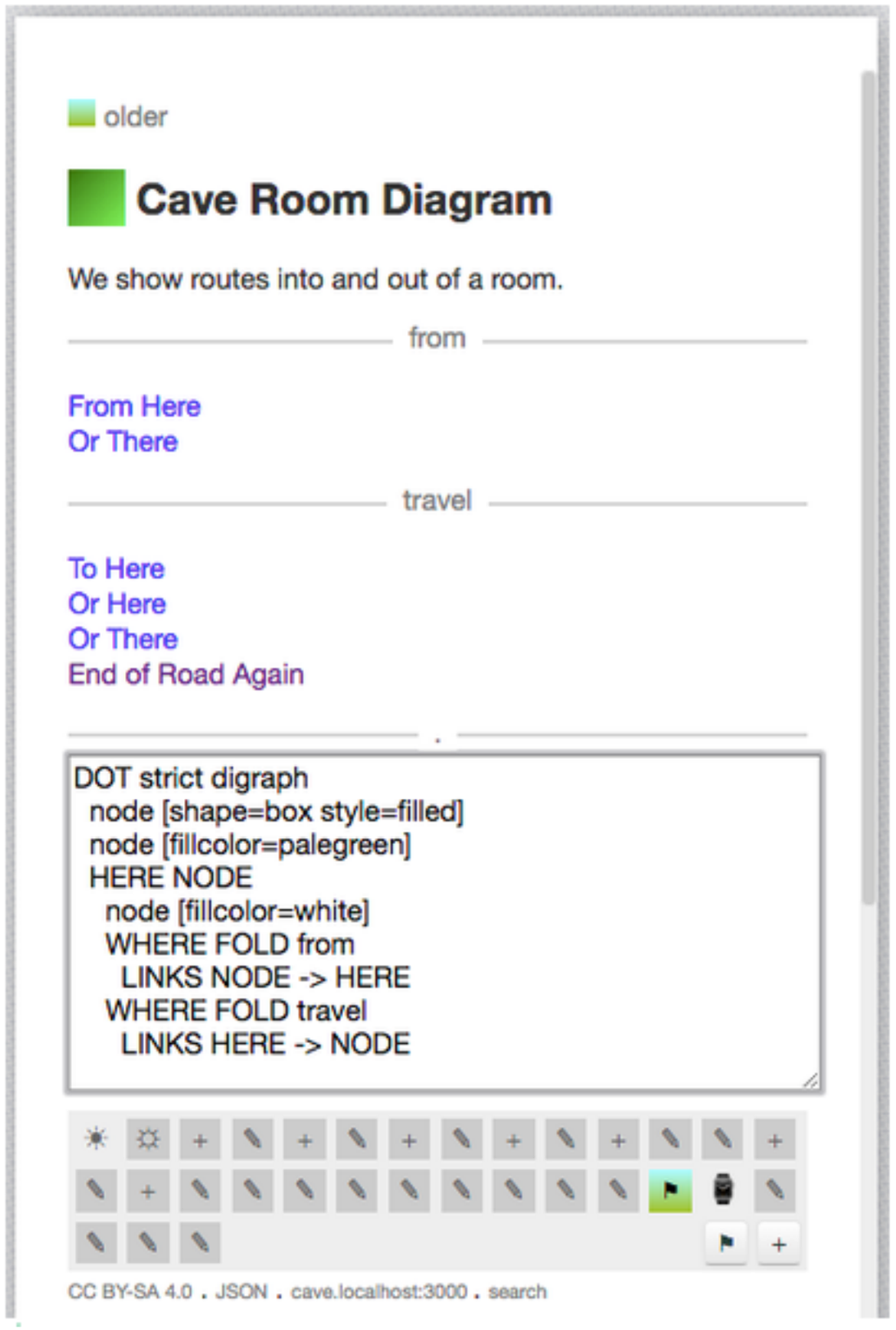
Five years ago I converted the Colossal Cave Adventure to wiki. Now I can draw my path through those twisty little passages.



New SELECT FOLD keyword enables drawing each Colossal Cave room with ins and outs and clicks to move so there is no game left, just browsing the database of places.



Here is the algorithmic markup written one place with test data and reused throughout the cave. Notice that the test data includes a link that jumps into the game.



For the code-curious here is the SELECT FOLD option.

```
166
167             if (ir.match(/^WHERE/)) {
168               let tree = nest()
169               var want = context.want
170               if (m = ir.match(/\/.*?\//)) {
171                 let regex = new RegExp(m[0].slice(1,-1))
172                 want = want.filter(item => (item.text||'').match(regex))
173               } else if (m = ir.match(/ FOLD ([a-z_-]+)/)) {
174                 var within = false
175                 want = want.filter((item) => {
176                   if (item.type == 'pagefold') {
177                     within = item.text == m[1]
178                   }
179                   return within
180                 })
181               } else if (m = ir.match(/[a-z_]+/)) {
182                 let attr = m[0]
183                 want = want.filter(item => item[attr])
184               } else trouble("can't do where", ir)
185               deeper.push({tree, context:Object.assign({},context,{want})})
186             } else
187
```

WardCunningham/cave-wiki

more-about-algorithmic-markup