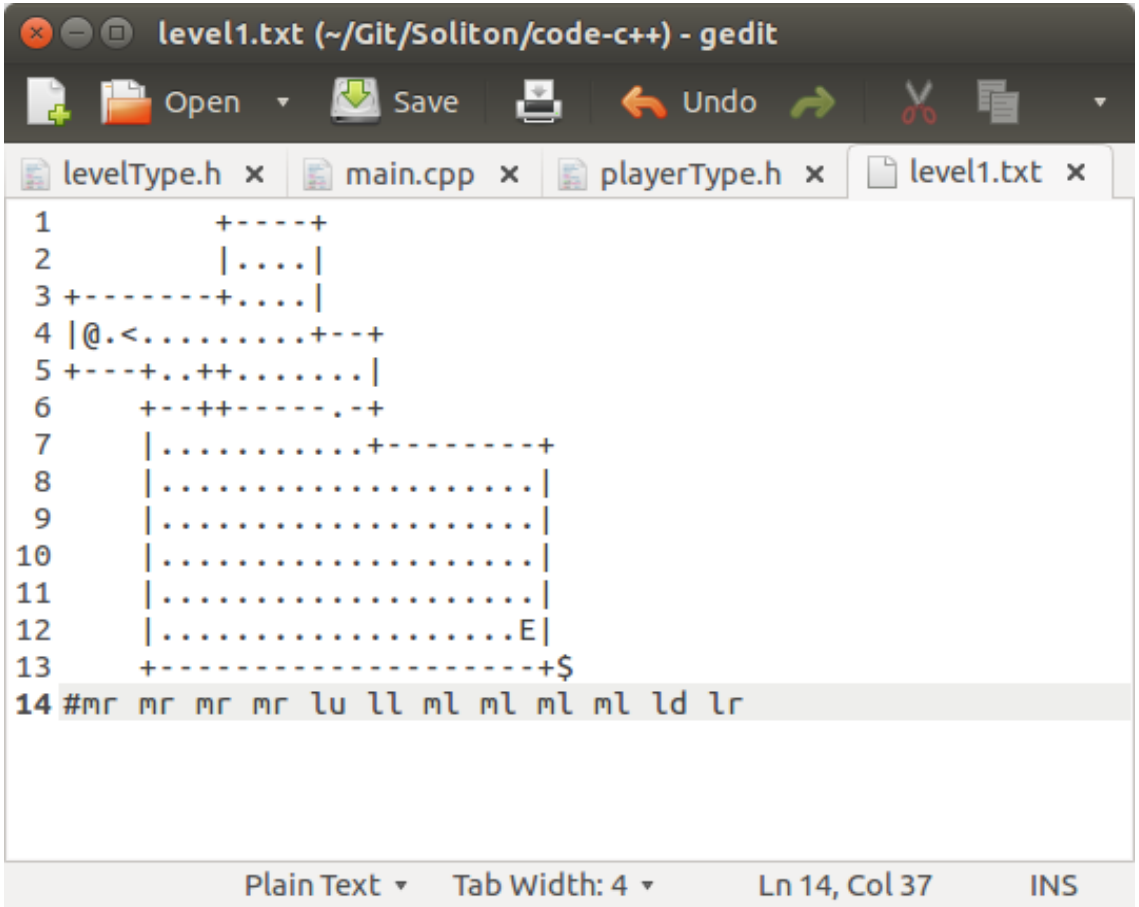
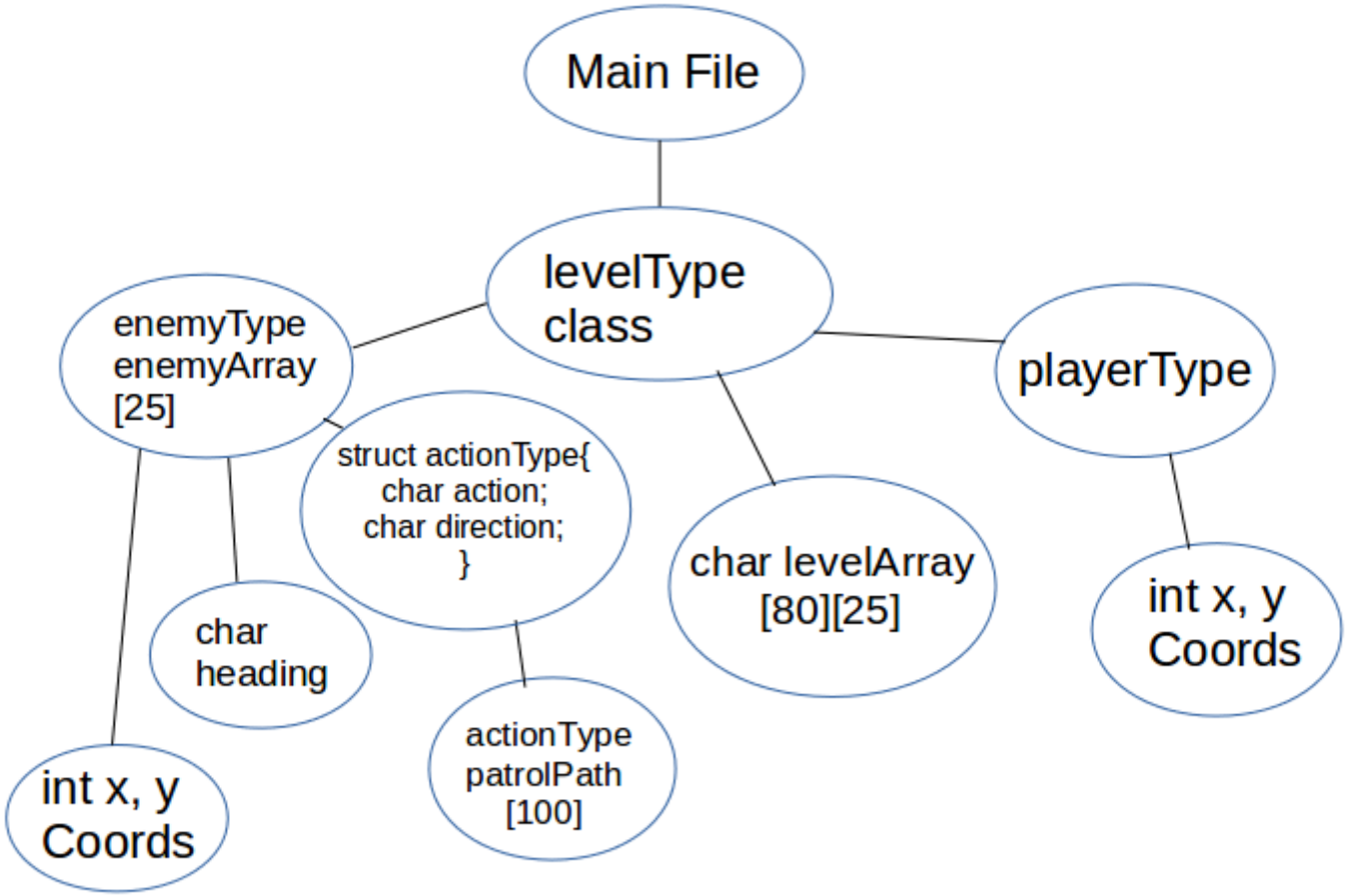
Soliton

Gameplay:

- Build your own levels
- Design guard paths
- Sneak around levels, avoiding guards, to make it to the goal



Data Structure



Things I did wrong:

1. Wrote code too early without a solid outline.

2. Didn't derive enemyType and playerType classes from a parent class, leading to lots of redundant functions.

3. Spaghetti code: The file buffer during level building is passed to four different functions in two different classes.

4. Used fixed-size arrays for enemy list and enemy path. Linked lists would have worked much better, wasted less memory, and allowed for more flexibility in level creation.

5. Underestimated amount of time custom enemy pathing would cost me.

Things I did right:

1. Allowed for imported levels instead of hard-coding them.
2. Built classes for everything.
3. Started using an outline (eventually).
4. Used version control.
5. Tried to use a separate function for every specific thing I wanted to do.
6. Picked a relatively simple design to implement.

You can see it at:

https://github.com/r0but/Soliton