# Final Project Log:

## Week 1-2 (18/01/23 – 01/02/23):

Aims:

* Create a demo of the game.
* Create a random nxn dungeon.
* Create an AI player that can navigate from the start to the end.

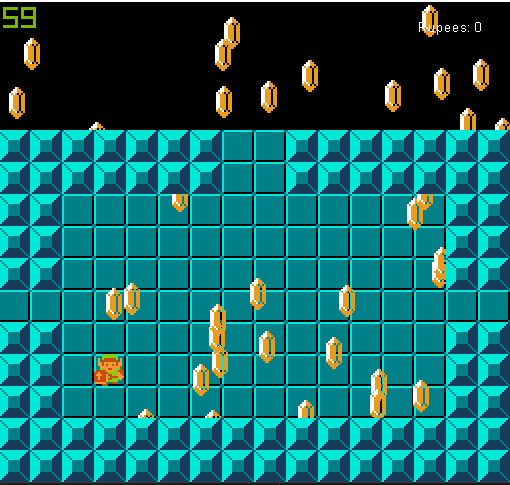
### Day 1 (20/01/23):

* Created room layout, moving player, and collectible rupees.
* Walls are Sprites that are added manually, and overlap with each other.
* 5 pixels per frame = 300 pixels per second?



Goal for next day: Turn room layout into a matrix. Add walls to list using this matrix.

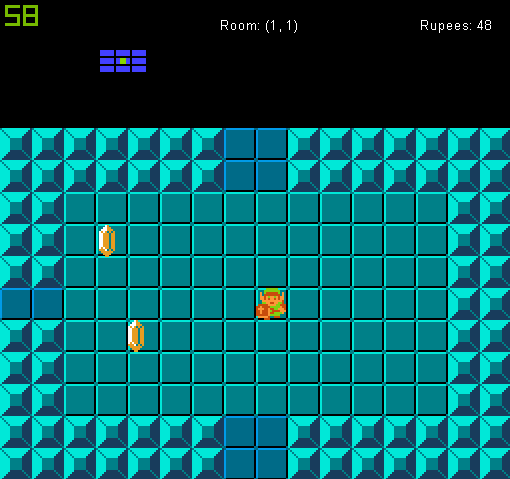
### Day 2 (21/01/23):



* Walls are now determined by 2D arrays.
* Rooms have 4 flags stating where their doors are.

Tomorrow: Transition between different rooms.

### Day 3 (22/01/23):



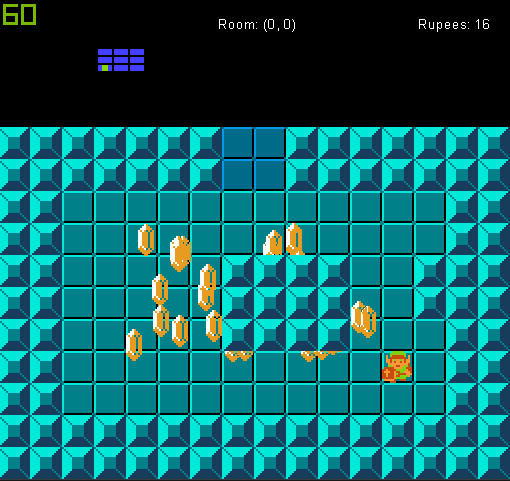
* Rooms are now stored in a matrix. Rooms are connected via 4 links that they have to neighbouring rooms. Inefficiencies from copied code removed.
* Minimap added, shows layout of dungeon, and link’s current room.
* Made a 3x3 maze with 2 branching paths.

Minimap dimensions:

* Link/triforce: 3x3 -> 9x9
* Rooms: 7x3 -> 14x6
* Room gap: 1 wide -> 2

Tomorrow: Start working on A\* to navigate to doors.

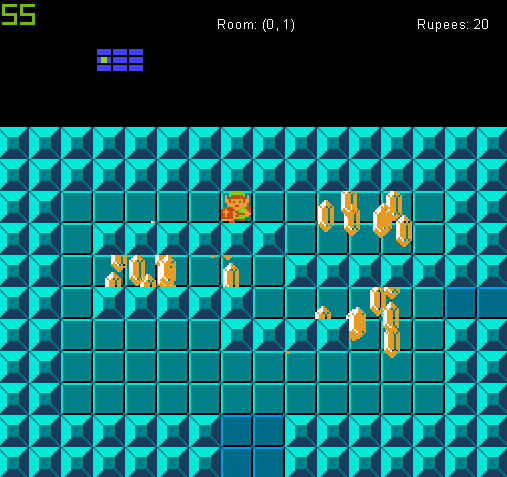
### Day 4 (23/01/23):



* Started working on A\* navigation.
* Arcade’s in-built algorithm was functional but failed to account for walls somehow and was too restricting.
* Began work on my own algorithm that I can tailor to my liking.
* Uses a grid of nodes rather than a list.
* Reaches goal, but not very well.
* Takes a very roundabout path due to no heuristic.
* Does not seem to care about walls, tries to ram through them.
* If it tries to go through 3 or more walls the game crashes.

Tomorrow: Try and fix A\*. Check if neighbours are correct, walls are not added etc.

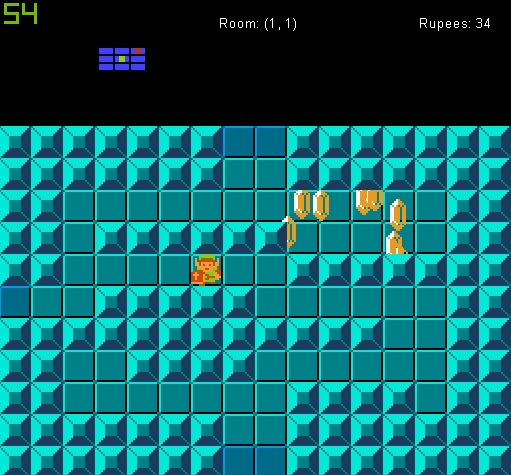
### Day 5 (25/01/23):



* Fixed A\* algorithm, it now always goes to the correct location, taking the fastest path (this was due to not removing nodes from open set), and without going through walls (due to misaligning the graph creation).
* Link will now restart A\* again when travelling to a new room, meaning he will soon be able to fully navigate between rooms.
* Split room and A\* related functions and classes into other files to structure the code. Will likely do this with the player in the future as well.

Tomorrow: Make Link locate a door to walk towards to initiate A\* without me specifying his goal.

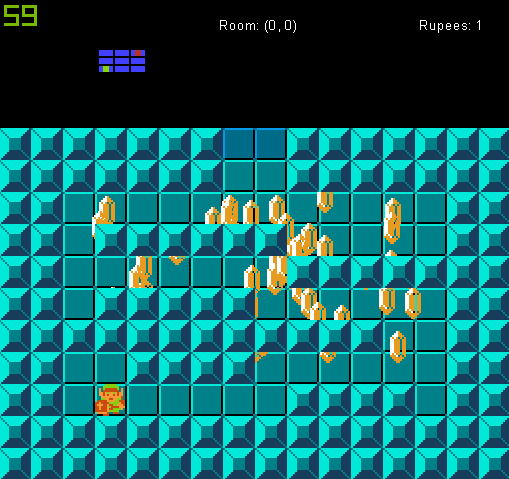
### Day 6 (26/01/23):



* A\* now restarts automatically when entering a new room.
* Will target a random door in the room that isn’t the one Link just came from. Stops when at a dead end.
* Added a marker to show where the final room should be.
* Made a GitHub repository to backup my files.

Tomorrow: Implement a depth-first search to reach the goal.

### Day 7 (29/01/23):



* Moved much all of Link’s AI into a player class to organise the code better, and keep all of the functionality self-contained.
* Began preparing for Link to choose between his own objective.
* Currently does not start moving automatically as the way A\* worked previously no longer matters.
* Began implementing the depth-first search.

Tomorrow: Continue with depth-first search. Hopefully help guide it to the end, backtracking may be difficult.