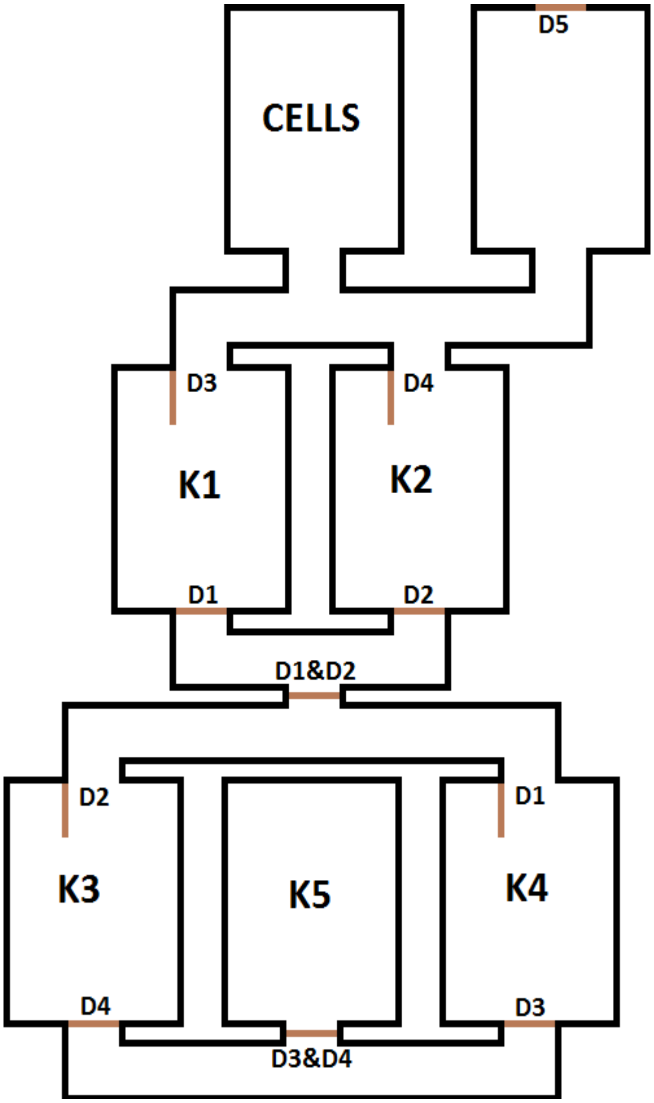
Paul Watkins

Assessment Proposal
The Fungeon

The Fungeon is a game designed to demonstrate the AI strand of the assessment. The player, an omnipotent dungeon master, finds out that their captives have escaped their cells. The captives, or minions, work together in a team to collect the keys they need in order to finally reach the master key needed to open the dungeon exit. The player's goal is simple- stop the minions from unlocking the exit and escaping.

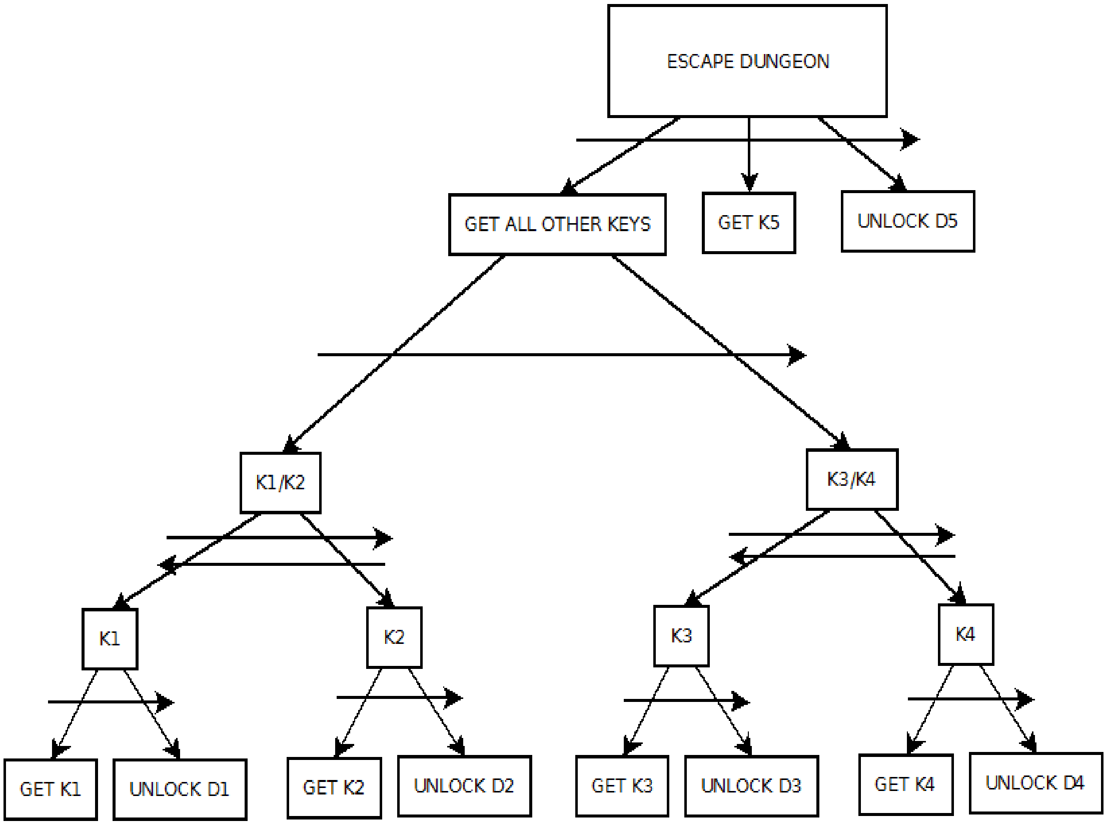
Game Progression and Logic



The above is the dungeon layout. K1 through K5 signify the spawn points of the given keys, and D1 through D5 refer to doors that the keys unlock (K3 unlocks D3 etc). Some doors require more than one key to unlock, either by a minion holding both keys or two minions holding a key each working together. K5 acts as a master key that unlocks any door. Some doors start locked, some unlocked.

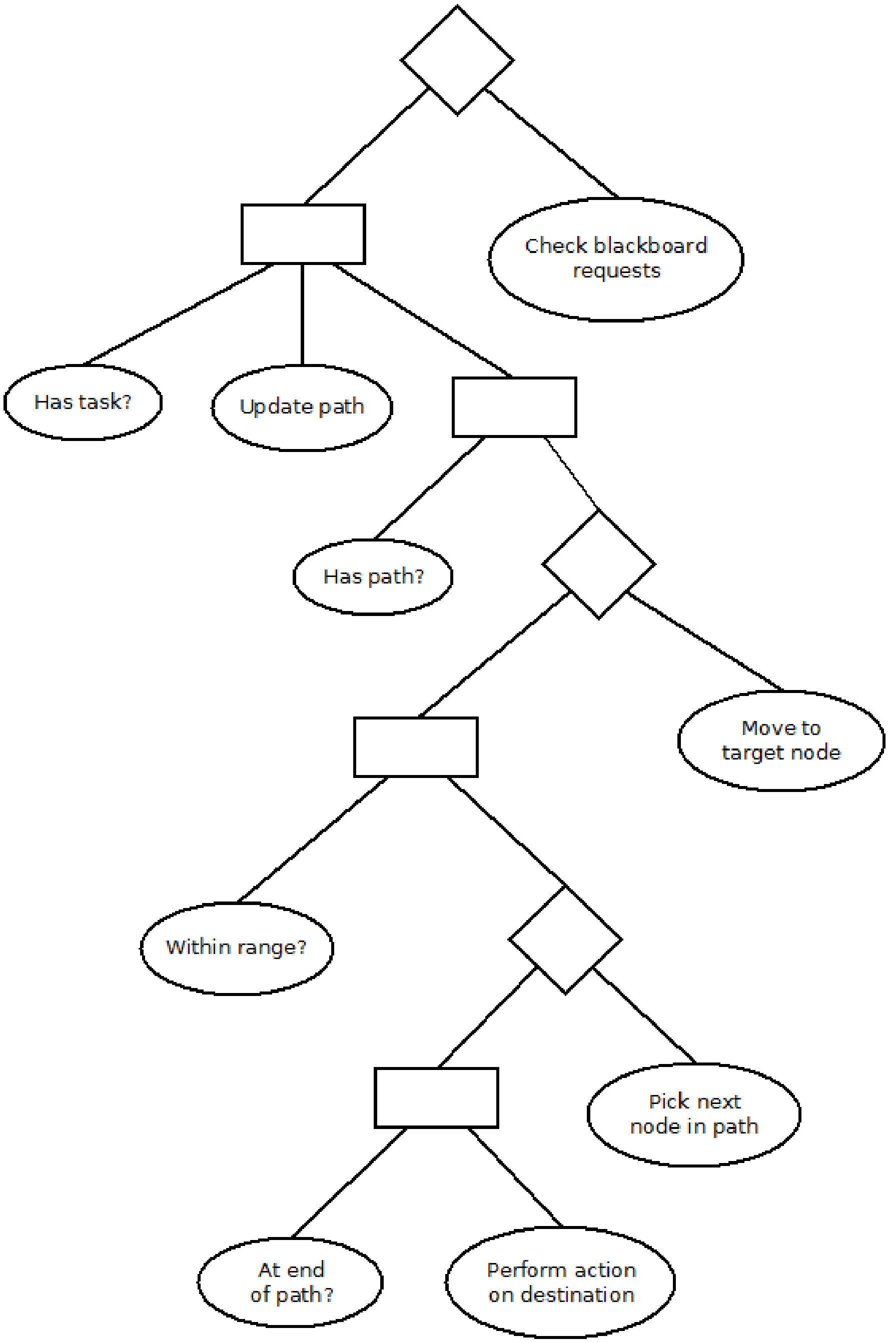
AI Logic

The minions start the game in the cells, with their goal to open the dungeon exit, D5. However, to reach K5 to do so, they must gather the other keys first. This is where a planner system comes in, to allow minions to coordinate in order to reach their goal:



The planner keeps track of which doors need unlocking/which keys are needed, then gives appropriately placed minions location-based goals through use of a blackboard system. As K1 and K2 can be collected in either order, this task is a non-linear compound task, labelled in the diagram with two arrows. Similarly with K3 and K4.

The logic of an individual minion is fairly simple. Given a goal by the planner, an A-Star path is calculated using the level's nav-mesh. This path may change dynamically based on how the player changes the level by introducing obstacles, locking doors etc. When the minion reaches its goal, its actions include picking up a key, or using a key to unlock a door.



Minions can hold multiple keys, but if there are multiple minions alive, the planner will attempt to use different minions to gather multiple keys at the same time.

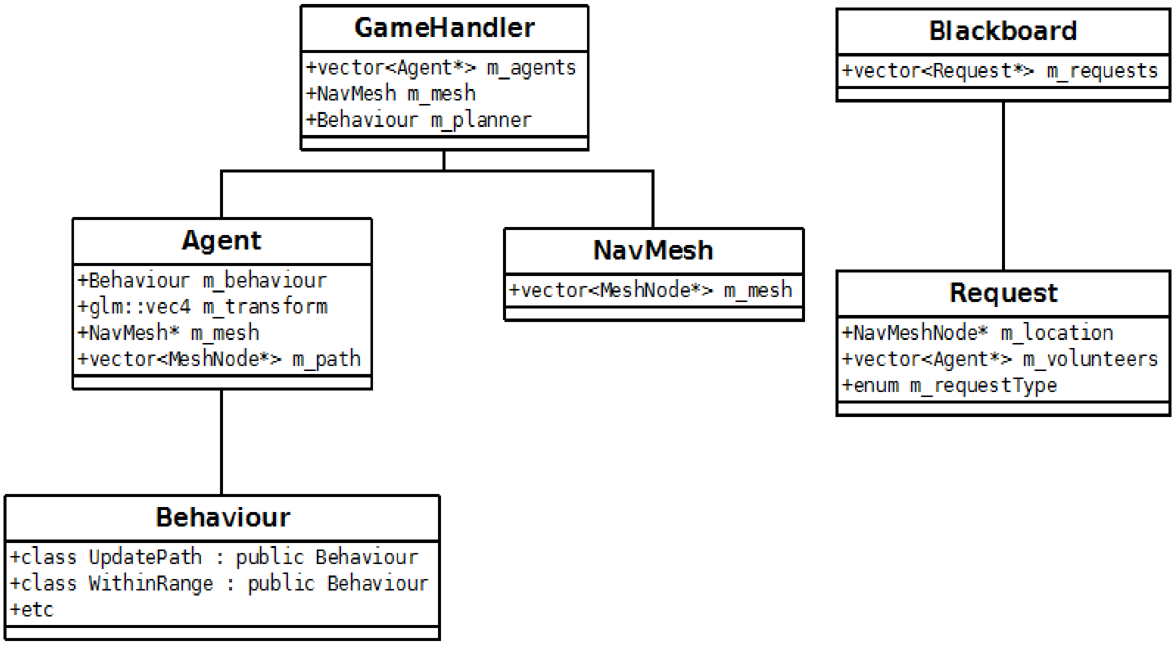
Player Abilities

The player has access to a few abilities to impede the minions of their goal.

> -They can place electrified tiles down on the ground anywhere in the dungeon, of which there are a set amount (not sure yet, maybe 5?). Attempting to put down more will remove the earliest placed tiles. These tiles are inactive until activated, after which they have a recharge period. These tiles will affect the calculated paths of the minions by changing the weight of the area they are placed. If a minion is on the tile when it is electrified, they will die and respawn in the cells area.
>
> If that killed minion is carrying a key, the key returns to its spawn point; however, it will also start flashing until it is collected again by another minion. This ties into the player's second ability.
>
> -While a key is flashing, the player has the use of that key, that is, they are able to unlock and lock doors it can open or close. If all the doors of a room are locked, poison gas is deployed in that room and any minion trapped inside dies permanently. The doors then unlock again. This poison gas has also has a recharge period- if not fully recharged, a room cannot have all its doors locked.

The player wins if all the minions are killed permanently.

Class Structure



The Blackboard singleton is one of the main ways the different AI systems communicate together. A list of requests are stored, which the planner sets requests to. Inactive agents will check the requests and subscribe to them, and the planner will then pick the best agent for the job. That agent is then assigned the request and it is labelled 'in progress'. If there are currently no available agents, the request remains. If the request is completed, the planner removes the request.

An agent stores its own path and behaviour. Its behaviour tells the agent to generate/edit a path from the nav-mesh, but the behaviour never has direct access to that data. The nav-mesh is imported within GameHandler and then passes its pointer to each of the agents.