Implementation – Reference Implementation – Goal implementation

The goal program is designed to work directly with our reference implementation, as it is just a show case of how such a program might look like, it will make assumptions based on how the reference implementation interact. For instance it will assume that there are entities called walls that are meant to block off tiles.

To see the source code of our goal program commented, look in appendix **[GOALCODEAPPENDIX].**

# Agent decision

A full flow chart of the goal program decision chart can be found on appendix **[INSERT GoalFlowChart Appendix reference]**.

As can be seen from the flow chart, the agent will try and find packages and bring them to a dropzone, if no such packages can be found or if no dropzone is found, the agent will start exploring the entire world.

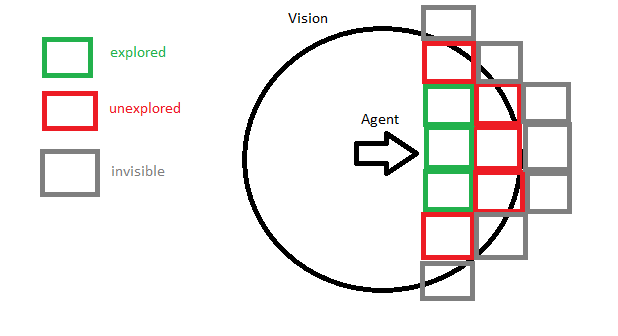
The goal program operates with a few different notions;

**Street** the first notion is the notion of streets. A tile is a street if it contains no wall types such a normal walls or impassableWalls(map boundaries walls) this means that the agent can move on this tile.

**Route** all the agents decisions are preplanned this means that the agent determines where to move to, this plan is put into a route, the agent will follow this route whenever it has nothing else to do, such as grabbing/releasing packages.

**Explored** the agent’s goal is to eventually have all tiles explored as this means that all packages has been collected from the world. The agent determines that a tile has been explored if it has seen all its adjacent tiles. This works great for the agent because until it reaches a wall the unexplored tiles that it has stored as a street will always be considered unexplored, no matter how far it moves, this makes the agent work much like putting a carrot in front of a mule, no matter how much the agent explores whenever it explores something, there is always something new that becomes unexplored. As such this will continue until a wall has been reached on all its paths.

A tile is determined to be explored if all tiles adjacent to has been seen by the agent, fig. **VisionExploredGoalAgent** shows an image of this.



**[Note(VisionExploredGoalAgent): An image of an agent’s vision and which it would determine to be explored]**