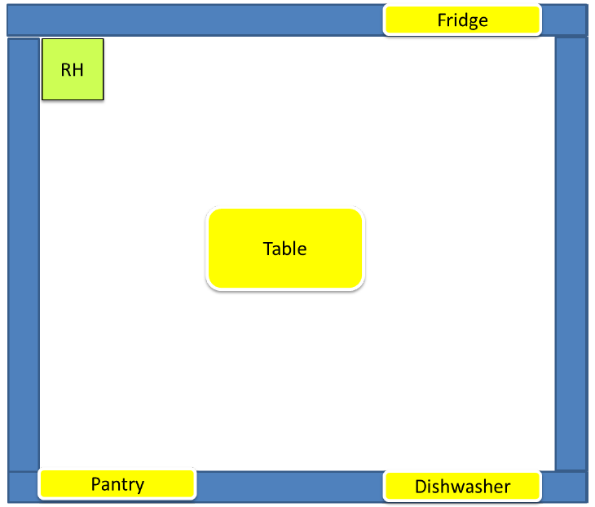
# **ANALISI DEI REQUISITI**

Il requisito fondamentale per poter affrontare la soluzione del problema presentatoci dal cliente è che il ddr robot **“rbr”** si muova all’interno di un perimetro. Analizzando più in profondità la frase precedente, si evince che non solo rbr si deve muovere all’interno del perimetro, ma deve anche avere una cella di destinazione specifica come obiettivo.

Infatti, nella stanza sono presenti diverse risorse: un **frigorifero**, una **lavastoviglie**, una **dispensa** e un **tavolo** (che si trova al centro). Inoltre, da qualche parte si trova anche la casa del robot **RH**. In particolare abbiamo i seguenti scenari:

1. La **disposizione** di frigo, lavastoviglie e dispensa è nota: **possiamo supporre che la configurazione sia quella in figura**.
2. La **disposizione** di frigo, lavastoviglie e dispensa NON è nota:
   1. Qualcuno ci dice la posizione degli elettrodomestici in una stanza che non conosciamo.
   2. Il robot ha una funzionalità più avanzata di esplorazione della stanza e capisce da solo dove sono gli elettrodomestici e il tavolo.

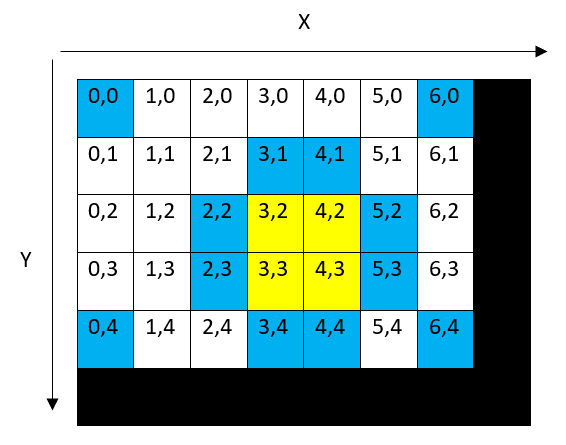
# **ANALISI DEL PROBLEMA**

Lo sprint 1 nasce perché nello sprint 0 si ha avuto la conferma che quando si ha una cella di destinazione, i percorsi per raggiungerla possono essere molteplici.

Per rendere **dinamico il percorso**, sarebbe utile avere una **strategia** che incrementi le performance per trovare il percorso migliore, cioè con minor numero di mosse.

Per soddisfare tali requisiti, ci sono diversi problemi da analizzare. A tale scopo si evidenziano le successive note:

1. Seguendo la suddivisione dello sprint 0, **la stanza risulta essere 8 celle in larghezza e 6 celle in lunghezza compresi i muri, quindi ha 5x7 celle accessibili.** Nel nostro SdR l’asse Y è verticale, l’asse X è orizzontale e l’origine (0,0) è in alto a sinistra. Il **tavolo** occupa le celle colorate in giallo.



1. Il frigorifero, la dispensa e la lavastoviglie non occupano nessuna cella all’interno della stanza perché sono incastonati nel muro. Al contrario, il tavolo occupa un numero finito di celle, così come RH. Alla luce di quanto appena detto, per il robot è utile sapere la posizione della loro **cella di accesso** (nella figura quelle colorate in azzurro).
2. **Se la dimensione della stanza e le posizioni degli elettrodomestici sono note**, esse risiedono nella base di conoscenza della stanza. In particolare abbiamo che:
   1. **Il frigorifero** si trova in (6,0)
   2. **La lavastoviglie** si trova in (6,4)
   3. **La dispensa** si trova in (0,4)
   4. **RH** si trova in (0,0)
   5. **Il tavolo occupa 4 celle e** si trova in (3, 2), (4, 2), (3, 3), (4, 3) e le sue celle di accesso sono (3, 1), (4, 1), (5, 2), (5, 3), (3, 4), (4, 4), (2, 2), (2, 3).
3. **RH** si trova sempre in (0,0), cioè nell’origine.
4. **Qualora le dimensioni della stanza non fossero note**, seguendo l’immagine fornita dal customer, si ha come conseguenza che:
   1. **Siccome il tavolo è al centro della stanza**, ma la stanza può non essere un quadrato equilatero, di conseguenza esso si trova ad una **distanza di almeno una cella su tutti i lati** e cioè non ha nessun lato adiacente ad un muro.
   2. **Il frigorifero** si trova sul lato nord della stanza vicino all’angolo est.
   3. **La dispensa** si trova sul lato sud della stanza vicino all’angolo ovest.
   4. **La lavastoviglie** si trova sul lato sud della stanza vicino all’angolo est.
5. Qualora la posizione del table non fosse nota, è necessario pensare ad un algoritmo di esplorazione che ne trovi le coordinate.
6. Il tavolo non può subire **spostamenti accidentali**, **ha dimensione quadrata, non ha forme irregolari e occupa 4 celle.**
7. È sufficiente che **una cella** sia **parzialmente occupata,** indipendentemente dalla percentuale di occupazione, e quella cella deve essere marcata come **non accessibile.**

**Ipotesi di Product Backlog:**

* Finalizzazione del work plan preliminare
* Creazione del metamodello/modello eseguibile
* Analisi del planner e del suo funzionamento
* Decidere quali e quante strategie di esplorazione usare
* Creazione Knowledge Base room
* Creazione di base della KB del tavolo e degli elettrodomestici
* Realizzazione logger per gli spostamenti

# **PROGETTO**

Partendo dall’ipotesi di Product Backlog e da quanto scritto in precedenza, ci sono ovviamente diverse possibilità di realizzazione. Noi, in particolare:

* Abbiamo scelto di utilizzare **il planner fornitoci dal customer** per implementare la ricerca ottimale del percorso migliore per arrivare ad un obiettivo. Tale planner si basa su algoritmi di ricerca caratteristici **dell’intelligenza artificiale**.
* Nel caso in cui **le dimensioni della stanza** e le **posizioni degli elettrodomestici** **non** **fossero note**, abbiamo adottato le seguenti strategie:
  + **Prima di tutto** il robot deve trovare il **perimetro della stanza**. Per fare ciò, deve percorrere i bordi e quando si trova davanti un muro, significa che il lato che stava percorrendo è finito e deve proseguire ruotando di 90°. Questo ciclo finché non ha scoperto quanto sono lunghi tutti e 4 i lati.
  + **In secondo luogo,** avendo trovato le dimensioni della stanza effettive (X=numero di colonne, Y=numero di righe), si possono **inferire le celle di accesso dei vari elettrodomestici**, ad esclusione del tavolo:
    - Il frigorifero si trova in (X-2, 0)
    - La dispensa si trova (0, Y-2)
    - La lavastoviglie si trova in (X-2, Y-2)
  + **Dopodiché** deve procedere **all’esplorazione** della stanza **per trovare la posizione esatta del tavolo**: siamo certi che il tavolo non si trova nelle celle adiacenti ai muri per i fatti evinti in precedenza, tuttavia rimane una grossa porzione di area da analizzare. A tale scopo, il robot cercherà di individuare il tavolo coprendo tutta la superficie, riga per riga.
* Per verificare gli spostamenti di rbr e per avere un tracciamento cronologico di tali spostamenti, abbiamo ritenuto necessario l’inserimento di un logger. Tale logger crea **un unico file di log comune a tutti gli attori** per evitare problemi di concorrenza e per vedere cronologicamente quali attori fanno quali azioni. Il file in questione si chiama “**FileLog”** ed è così strutturato**:**
  + Ogni volta che viene pianificato un nuovo obiettivo, lo scriviamo nel file. Esempio: GOAL: RH | 0, 0.
  + Ogni volta che viene raggiunto un obiettivo, lo scriviamo nel file.

Esempio: GOAL REACHED: RH | 0, 0.

* + Tutte le altre righe sono così composte:

*“Nome dell’attore | Vecchia posizione del robot | Mossa da fare | Nuova posizione del robot”.*