universidade de aveiro
theoria poiesis praxis

RMI ASSIGNMENT2

João Torrinhas (98435), Diogo Torrinhas (98440)

Aspetos melhorados do último projeto

- Como este projeto se baseia no projeto anterior, nós tivemos de fazer algumas melhorias nomeadamente:
  - Na escrita no ficheiro.
  - Na deteção de ciclos e interseções, por causa do ruído.
  - No cálculo do caminho mínimo.

Challenge 4

- **Objetivos**
  - Implementar um sistema de localização.
  - Explorar o mapa todo e voltar ao inicio.
  - Escrever o mapa num ficheiro de texto.
  - Descobrir o caminho mínimo que passa por todos os beacons e volta ao início.

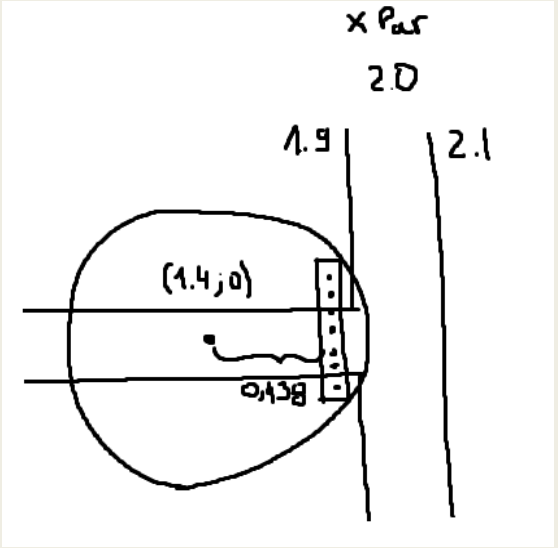
- **Implementação**
  - GPS/Mapping.
  - Escrita no ficheiro.
  - Cálculo caminho mínimo.

GPS/Mapping

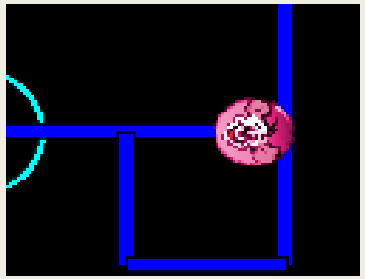
- **Objetivos**
  - Implementar um sistema de localização para o robô se orientar no mapa.
  - Explorar o mapa todo.

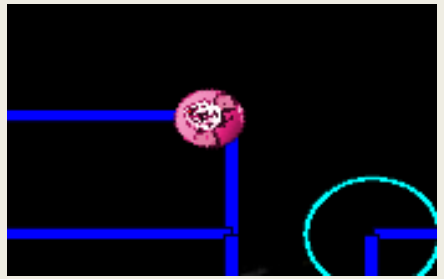
- **Implementação**
  - Foram usadas as fórmulas dadas pelos professores mas, em vez de calcularmos o teta, transformámos o *self.measures.compass* para radianos e usámos esse valor.

GPS/Mapping - Correction



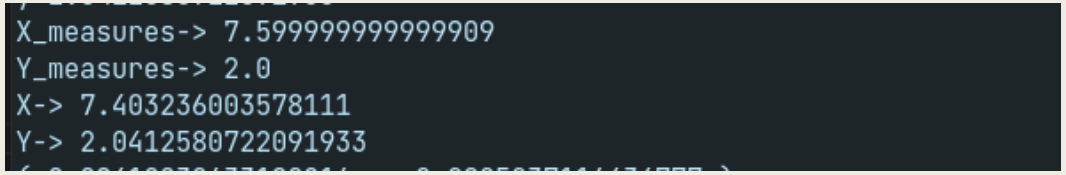
- Implementação
  - A correção do X/Y é efetuada em todas as interseções e curvas.
  - Para detetar se tem de haver correção ou não, vemos se o valor da soma do x, centro do robô, com 0.438 é inferior ao (Xpar – 0.1) ou superior a (Xpar + 0.1). Se for ligeiramente inferior ou superior corrigimos o valor, caso contrário (está no intervalor 1.9-2.1) mantemos o valor do X.
  - A correção, para este exemplo, é feita subtraindo o (Xpar – 0.1) a 0.438 obtendo o novo valor de X.
  - Para o Y corrigimos do mesmo género.





GPS/Mapping – Resultados obtidos

- O gps está próximo do desejado, tem um bocado de erro mas está bastante próximo do self.measures.x/y como se pode ver na figura abaixo.

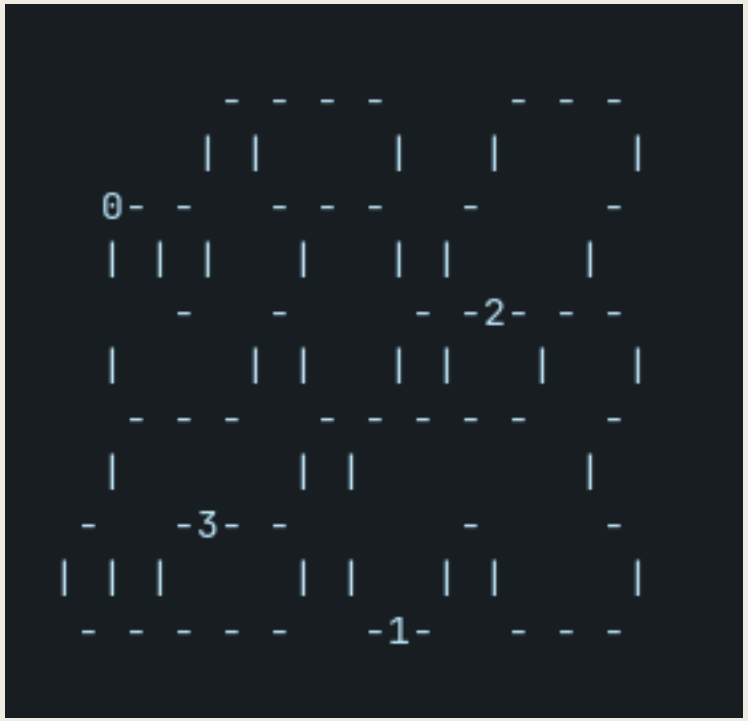


Escrita no ficheiro

- Para a escrita do mapa, como o GPS se desviava um pouco do valor que deveria ter, tivemos de adotar outra solução.
- A solução que encontramos foi então arredondar o valor do gps para 1, 2, 3 etc e somar sempre esse valor à posição inicial da matriz(24,10) e colocar "-" ou "|" em cada uma das células.
- Por fim, colocamos " " em todas as células pares e escrevemos a matriz no ficheiro.

Escrita no ficheiro – Resultados obtidos

- A escrita do mapa no ficheiro está completamente funcional.

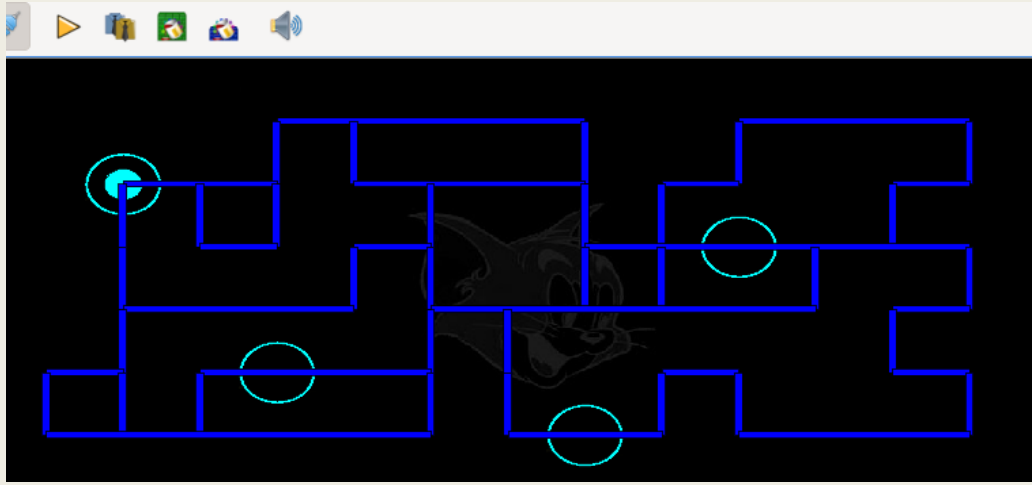


Cálculo caminho mínimo

- **Objetivos**
  - Calcular o caminho mínimo que passa por todos os beacons e volta ao início

- **Implementação**
  - Usámos o mesmo algoritmo do trabalho anterior, djirska, em que à medida que o robô conhece o mapa é criado o grafo onde cada interseção e cada beacon corresponde a um vértice do grafo.
  - Após o grafo estar concluido e o robo ter chegado outra vez à posição inicial ou o tempo ter terminado, é calculado o caminho mínimo que passa por todos os *beacons* e volta ao *beacon* inicial.
  - O custo é definido pelo número de células percorridas entre dois vértices.

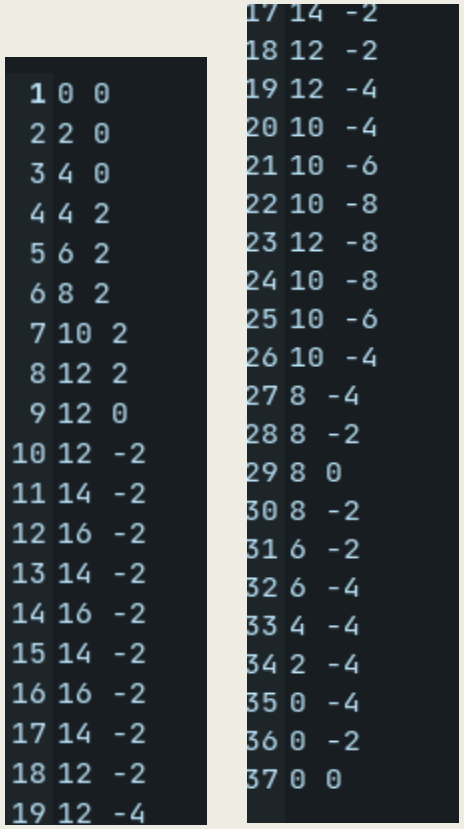
Cálculo caminho mínimo



- **Implementação**
  - Após o robô conhecer o mapa todo, ou seja, o vetor com as direções estar vazio, vemos em que vértice é que o robô se encontra.
  - Depois, aplicamos o algoritmo *Djikstra* do vértice onde ele está ao vértice inicial. Após aplicarmos o algoritmo, este vai retornar-nos um vetor com os vértices do caminho mínimo.
  - Tendo o vetor com os vértices do caminho mínimo, vemos a célula correspondente a cada um dos vértices e comparamos os vértices dois a dois, ou seja, onde o robô se encontra e o adjacente a ele.
  - A comparação é feita com base no valor da célula. Imaginando que ele está na célula (32,10) e o seu vértice adjacente é (30,8) então ele vira para esquerda.
  - Caso o vértice adjacente, imediatamente após ele ter terminado o mapa, estar atrás dele, ou seja, ele está na posição (32,10) e o adjacente está na (26,14), então ele roda até se virar ao contrário e continua o seu caminho.

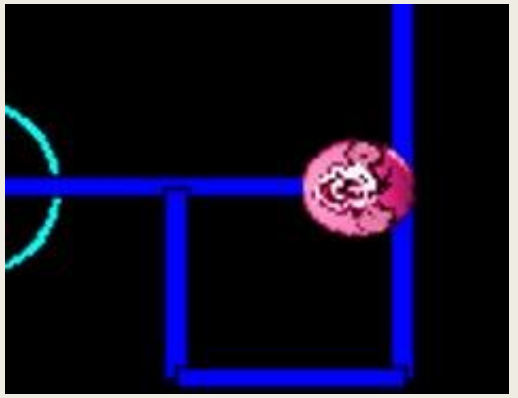
Cálculo do caminho mínimo– Resultados Obtidos

- Em mapas menos complexos calcula bem o caminho mas em mapas mais complexos às vezes o caminho obtido não é 100% correto.



Aspetos a melhorar

- Remover as interseções em algumas ocasiões.
- Voltar à posição inicial depois de ter explorado o mapa todo, pois devido ao cálculo do gps, as células da matriz são par passado 2/3 ciclos após ele ter detetado a interseção e não imediamente antes de ter detetado a interseção.
- Ainda que raro, às vezes não deteta interseções por isso, também é um aspeto que pode ser melhorado.
- Às vezes quando o robô volta a passar por um beacon remove, no ficheiro de escrita, o número associado a esse beacon.



Conclusão

- Aprendemos a trabalhar com algoritmos de pesquisa.
- Pode ser usado para trabalho futuro na área de Inteligência Artificial.

Bibliografia

- https://benalexkeen.com/implementing-djikstras-shortest-path-algorithm-with-python/