



# Esercitazione 9 (11-26)



# Esercizi C++

- Es. 6.2
- Es. 5.2
- Es. 5.3 -- Usare un `vector<Vehicle*>`
- Es. 4.8 del 2015 -- Partire dall'esempio "nodes", fornito



## 9.5 Campo minato in C++, ricorsione

- Riprendere l'esercizio 8.6
- Nel metodo `play_at`
  - Se viene scoperta una cella con valore 0, non ci sono mine attorno
  - $\Rightarrow$  Chiamare `play_at` sulle celle coperte attorno
- Usare la classe C++ tramite la gui Python, fornita



## 9.6 Percorso del cavallo, backtracking

- In *Python*, riprendere l'esercizio 7.7
- Implementare il metodo `flag_at`
  - Partire dalla situazione di gioco attuale
  - Usare il *backtracking* per cercare una soluzione
  - Se esiste una soluzione, fare una sola mossa al posto dell'utente

*Per migliorare le prestazioni, ad ogni scelta, ordinare le alternative dando priorità alle celle più chiuse*

[https://en.wikipedia.org/wiki/Knight's\\_tour#Warnsdorf's\\_rule](https://en.wikipedia.org/wiki/Knight's_tour#Warnsdorf's_rule)

