

JSLIGO

MORE INTERESTING EXEMPLE: MOUVEMENTS DE LINK

```
// Variant types https://ligolang.org/docs/variants/?lang=jsligo
type direction = ["North"] | ["East"] | ["South"] | ["West"];
type state = ["Idle"] | ["Moving"];
// Mandatory
type storage =
{
  state: state,
  orientation: direction
};
// Utility
type return_ = [list<operation>, storage];
const noop : list<operation> = list([]);

@entry
const start = (_: int, store: storage): [list<operation>, storage] => match(store.state){
  when(Moving()): failwith("Unreachable state");
  when(Idle()): [noop, {...store, state: Moving()}];
};

@entry
const stop = (_: int, store: storage): [list<operation>, storage] => match(store.state){
  when(Moving()): [noop, {...store, state: Idle()}];
  when(Idle()): failwith("Unreachable state");
};

@entry
const face = (d: direction, store: storage): [list<operation>, storage] => match(store.state){
  when(Moving()): failwith("Unreachable state");
  when(Idle()): [noop, {state: Idle(), orientation : d}];
};
```

JSLIGO

QUICK NOTES

En LIGO:

- Un contrat doit définir un type « storage »
- Un contrat doit avoir au moins 1 entrypoint
- Failwith stope l'exécution du contrat, c'est une bonne propriété, on veut limité la consommation de gas!
- Possibilité d'utiliser des « Variant » (proche des tagged union TS)
 - Les valeurs sont crée avec un constructeur
 - On accède aux valeurs uniquement avec l'expression match/when