

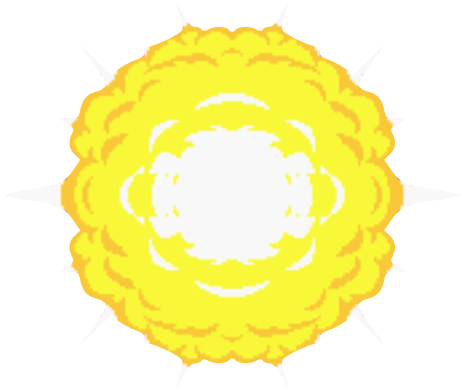
QUE SE PASSE-IL SI ON RÉCUPÈRE LES DIRECTION D'UNE LIB

LINK TO THE PATH

```
/** lib tierce */
interface Direction{
    public int toInt();
};
class North implements Direction{
    public int toInt(){ return 1;};
};
class East implements Direction{
    public int toInt(){ return 2;};
};
class South implements Direction{
    public int toInt(){ return 3;};
};
class West implements Direction{
    public int toInt(){ return 4;};
};

/** notre code */
interface DirectionImprove{
    public String toString();
};
class NorthImprove extends North implements DirectionImprove{
    public String toString(){ return "north";};
};
class EastImprove extends North implements DirectionImprove{
    public String toString(){ return "east";};
};
class SouthImprove extends North implements DirectionImprove{
    public String toString(){ return "south";};
};
class WestImprove extends North implements DirectionImprove{
    public String toString(){ return "west";};
};
```



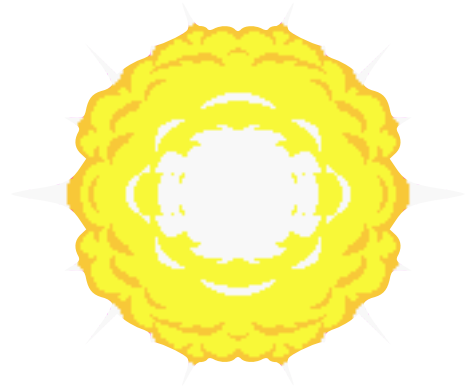


**We are stuck ! Assez joué avec Java.
Malheureusement C#, Typescript ne
font pas beaucoup mieux**

MAGNET 2-DUALS - THOUGHTS

LINK TO THE PATH

QUE SE PASSE-T-IL SI ON RÉCUPÈRE LES DIRECTIONS D'UNE LIB



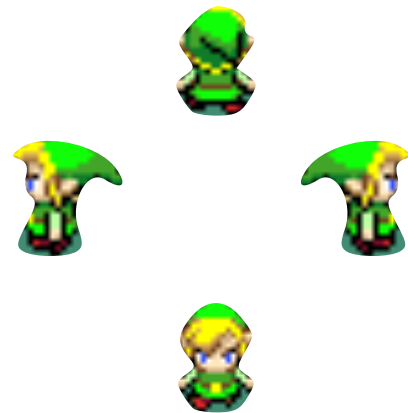
We are stuck ! Assez joué avec Java.
Malheureusement C#, Typescript ne
font pas beaucoup mieux

```
/** lib tierce */
interface Direction{
    public int toInt();
};
class North implements Direction{
    public int toInt(){ return 1;};
};
class East implements Direction{
    public int toInt(){ return 2;};
};
class South implements Direction{
    public int toInt(){ return 3;};
};
class West implements Direction{
    public int toInt(){ return 4;};
};

/** notre code */
interface DirectionImprove{
    public String toString();
};
class NorthImprove extends North implements DirectionImprove{
    public String toString(){ return "north";};
};
class EastImprove extends North implements DirectionImprove{
    public String toString(){ return "east";};
};
class SouthImprove extends North implements DirectionImprove{
    public String toString(){ return "south";};
};
class WestImprove extends North implements DirectionImprove{
    public String toString(){ return "west";};
};
```

LINK TO THE PATH

UTILISATION D'UN TYPE « OU » DANS UN SYSTÈME DE TYPE ML



```
type direction = | North | East | South | West

let label d = match d with
| North -> "north"
| East -> "east"
| South -> "south"
| West -> "west"

let _ = print_endline(label North);;
let _ = print_endline(label East);;
let _ = print_endline(label South);;
let _ = print_endline(label West);;
```

