




Harry Potter



La force obscure

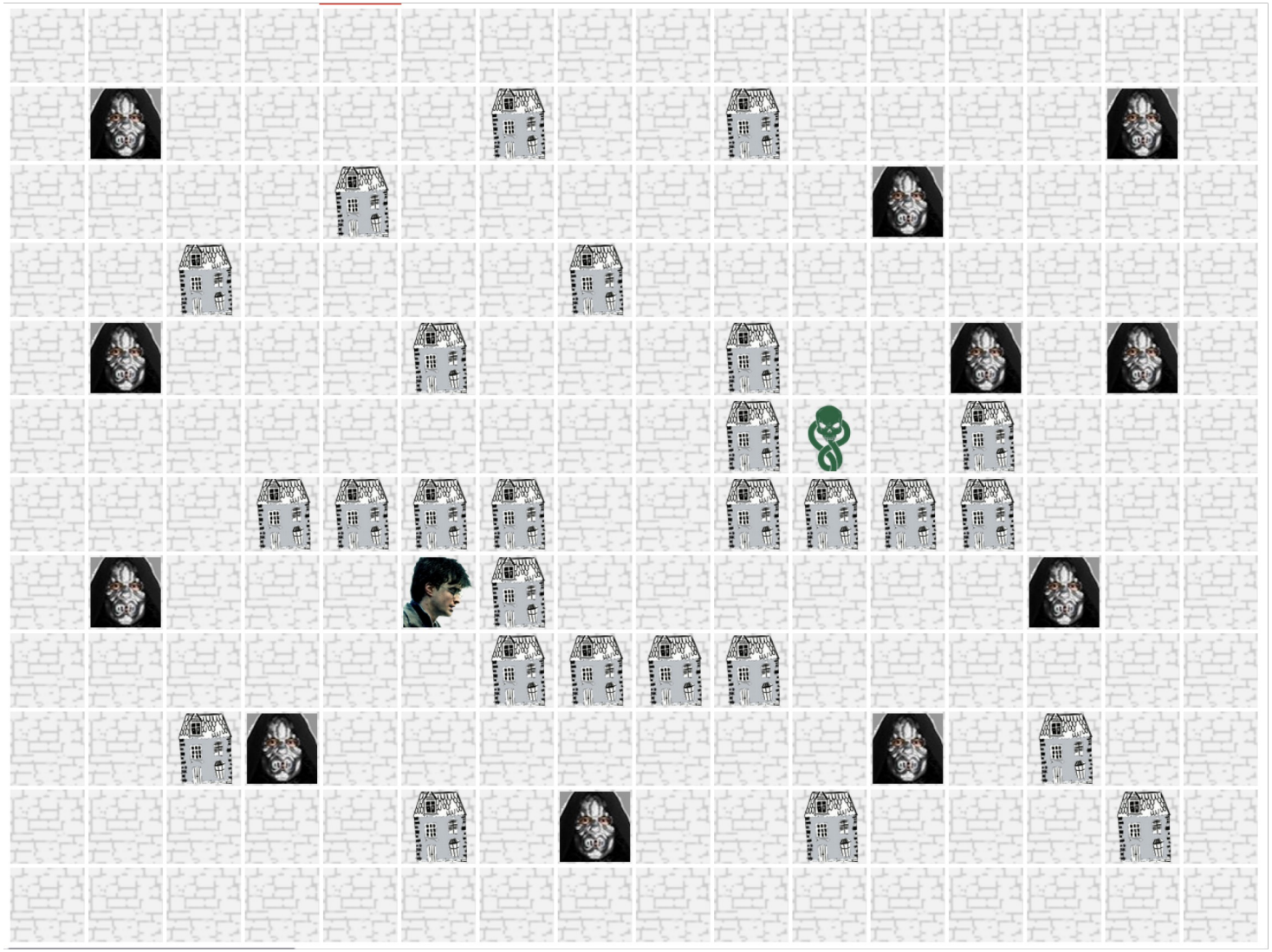


Bâtiment



Mangemort

# Trouvez la Force Obscure



# Concepts clés

- algorithme
- opération primitive
- condition
- test (SI … ALORS … SINON …)
- boucle (TANT QUE … FAIRE …)
- sous-problème (fonction)

# Commandes

- `move()`
- `turn_left()`
- `turn_right()`
- `destroy_voldemort()`

# Conditions

- `can_move()`
- `is_in_front_of_enemy()`
- `is_on_target()`

- **Vrai** ou **Faux** **`True, False`**
- Possibilité de combiner des conditions
  - condition **et** condition `cond` **`and`** `cond`
  - condition **ou** condition `cond` **`or`** `cond`
  - **non** condition **`not`** `cond`

# Un peu de logique

- Vrai **et** Vrai Vrai
- Vrai **et** Faux Faux
- Faux **et** Vrai Faux
- Faux **et** Faux Faux

# Un peu de logique

- Vrai **ou** Vrai Vrai
- Vrai **ou** Faux Vrai
- Faux **ou** Vrai Vrai
- Faux **ou** Faux Faux

# Un peu de logique

- **non** Vrai Faux
- **non** Faux Vrai

# Récupération de l'état

- `get_direction()`

  Retourne EAST, WEST, SOUTH ou NORTH

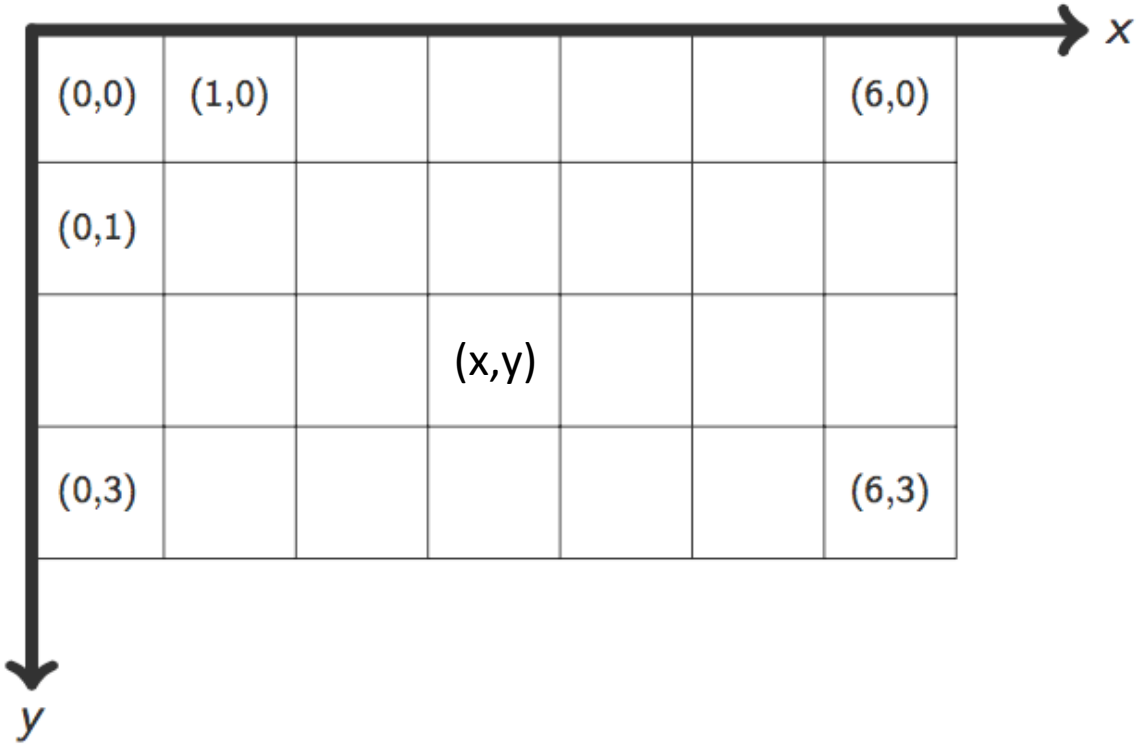
- `get_x(),get_y()`

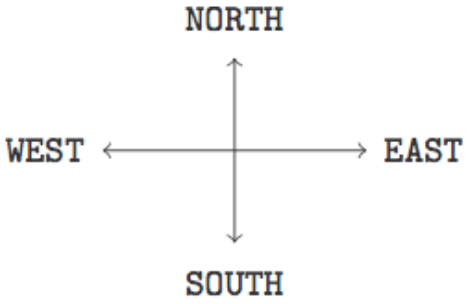
  Retourne …

- `get_target_x(),get_target_y()`

  Retourne …

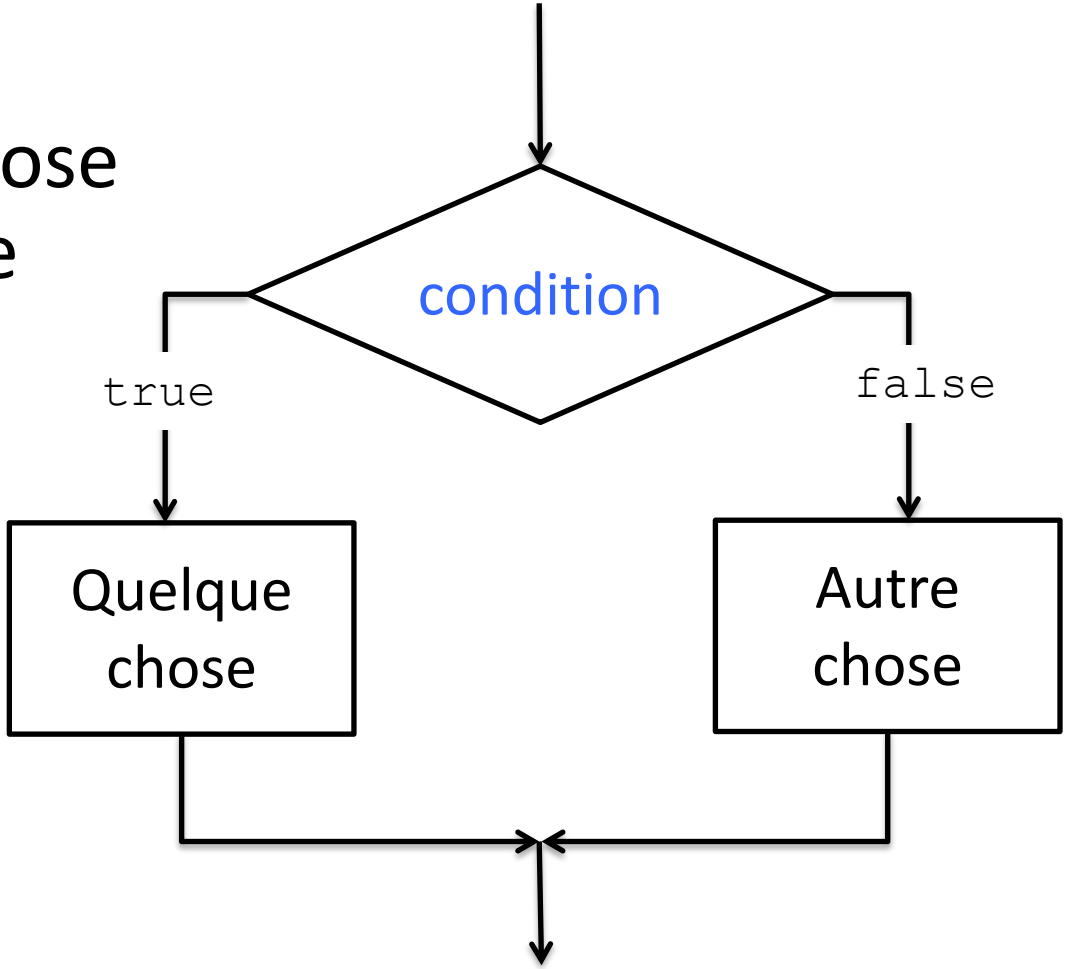
# Coordonnées et directions



# Test

**SI** condition
**ALORS** Quelque chose
**SINON** Autre chose

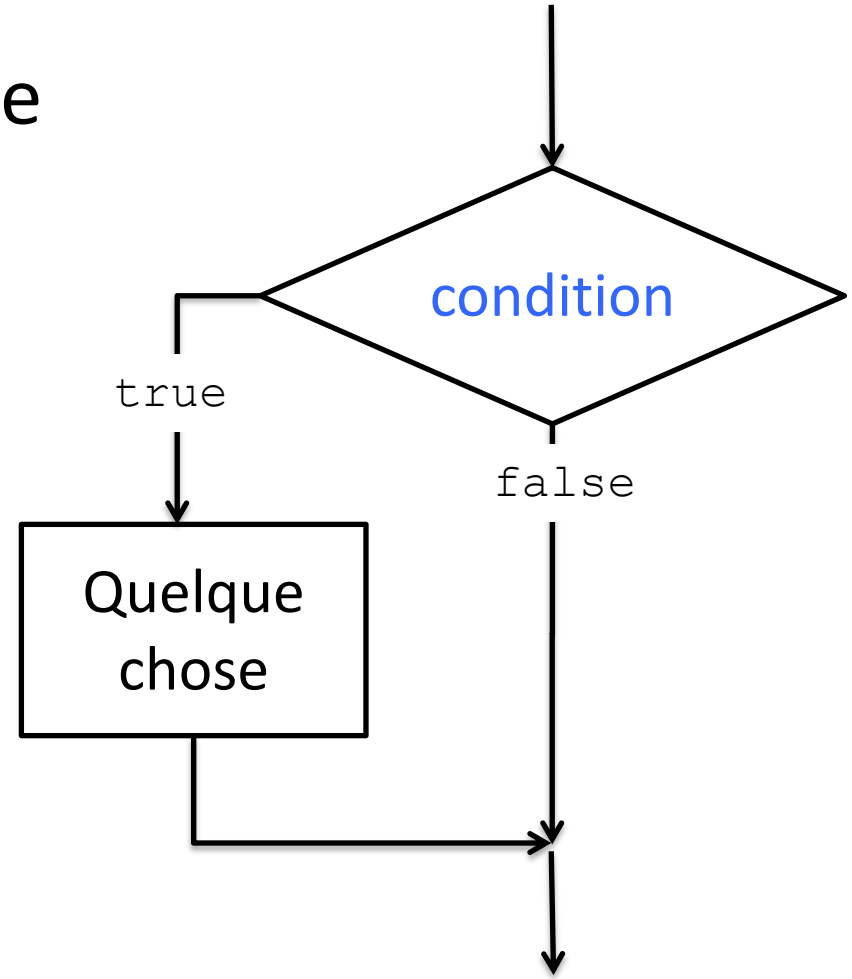



```
if cond:
   qqchose
else:
   autrechose
```

# Test (une branche)

**SI** condition
**ALORS** Quelque chose

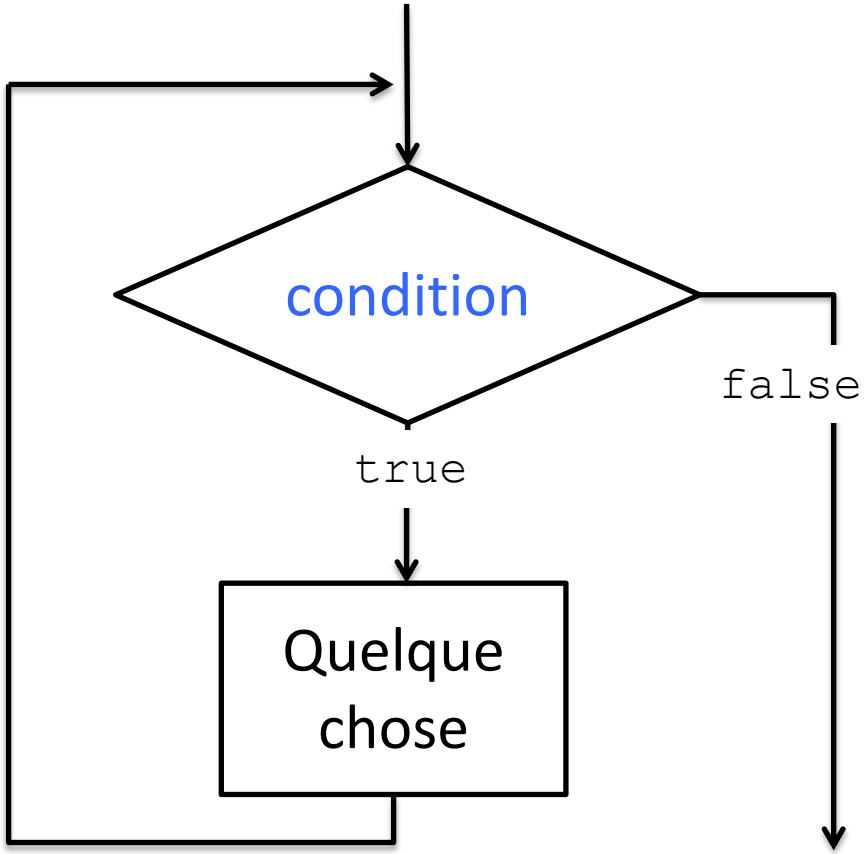
```
if cond:
   qqchose
```



# Boucle

**TANT QUE** condition
**FAIRE** Quelque chose

```
while cond:
    qqchose
```



# Sous-problèmes ⟷ Fonctions

**POUR** problème
| sous-problème 1
| sous-problème 2
| sous-problème 3
| sous-problème 2
| sous-problème 2

**POUR** sous-problème 1
| sous-problème 1a
| sous-problème 1b

**POUR** sous-problème 2
| …

```
def probleme():
    sous_probleme_1()
    sous_probleme_2()
    sous_probleme_3()
    sous_probleme_2()
    sous_probleme_2()
```

```
def sous_probleme_1():
    sous_probleme_1a()
    sous_probleme_1b()
```

```
def sous_probleme_2():
    ...
```

# Exemple

## Pseudo-code

```
TANT QUE Harry peut bouger FAIRE
|
|     avancer d'une case
|     SI Harry peut bouger ALORS
|     |
|     |     avancer d'une case
|     |
|     SINON
|     |
|     |     tourner vers la gauche
|     |
|
```

## Python

```
while can_move():

    move()
    if can_move():

        move()

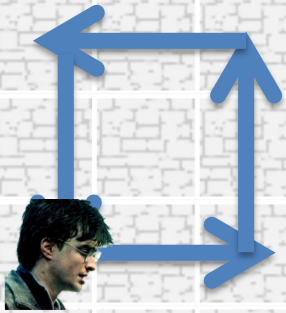
    else:

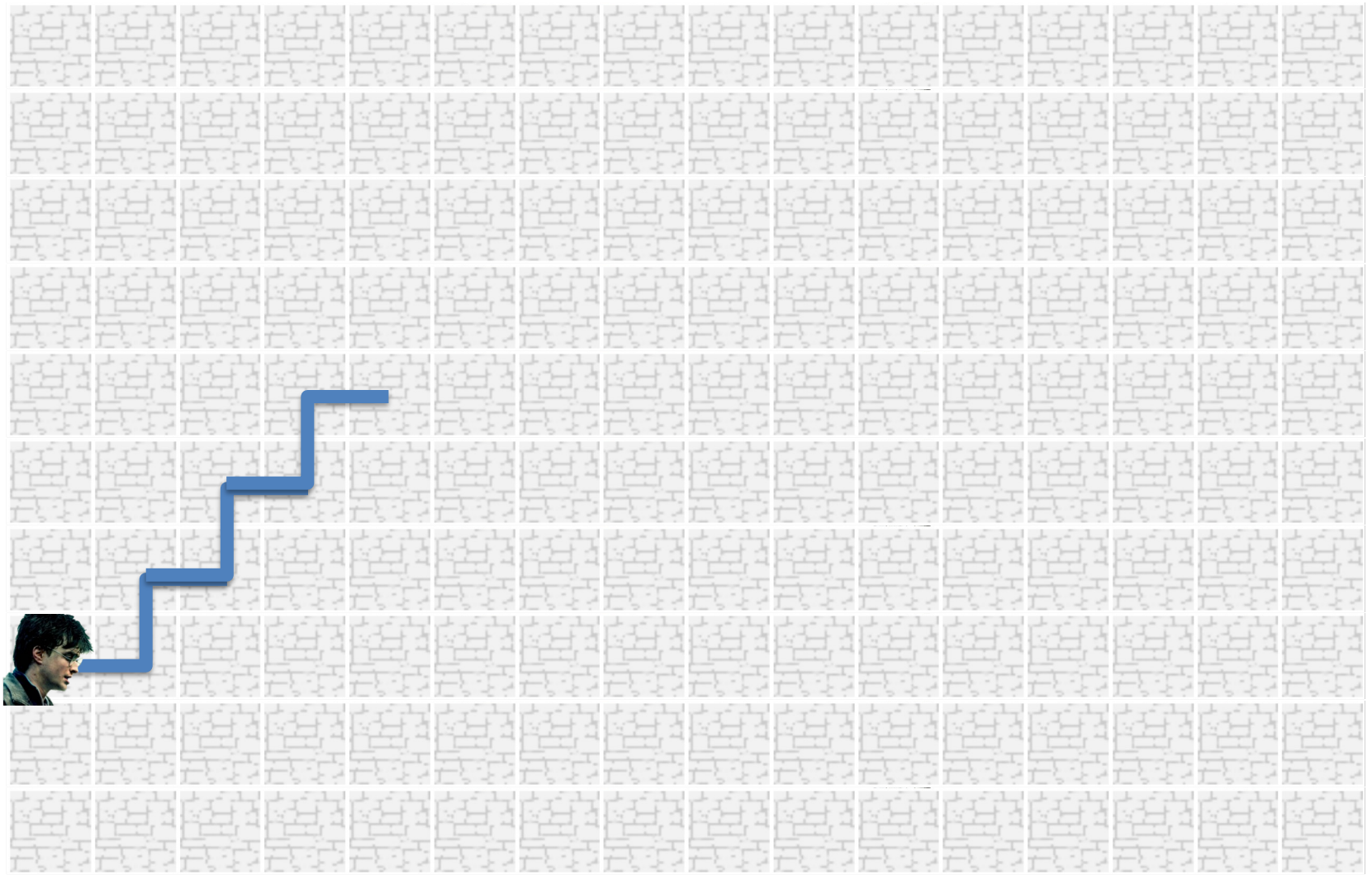
        turn_left()
```

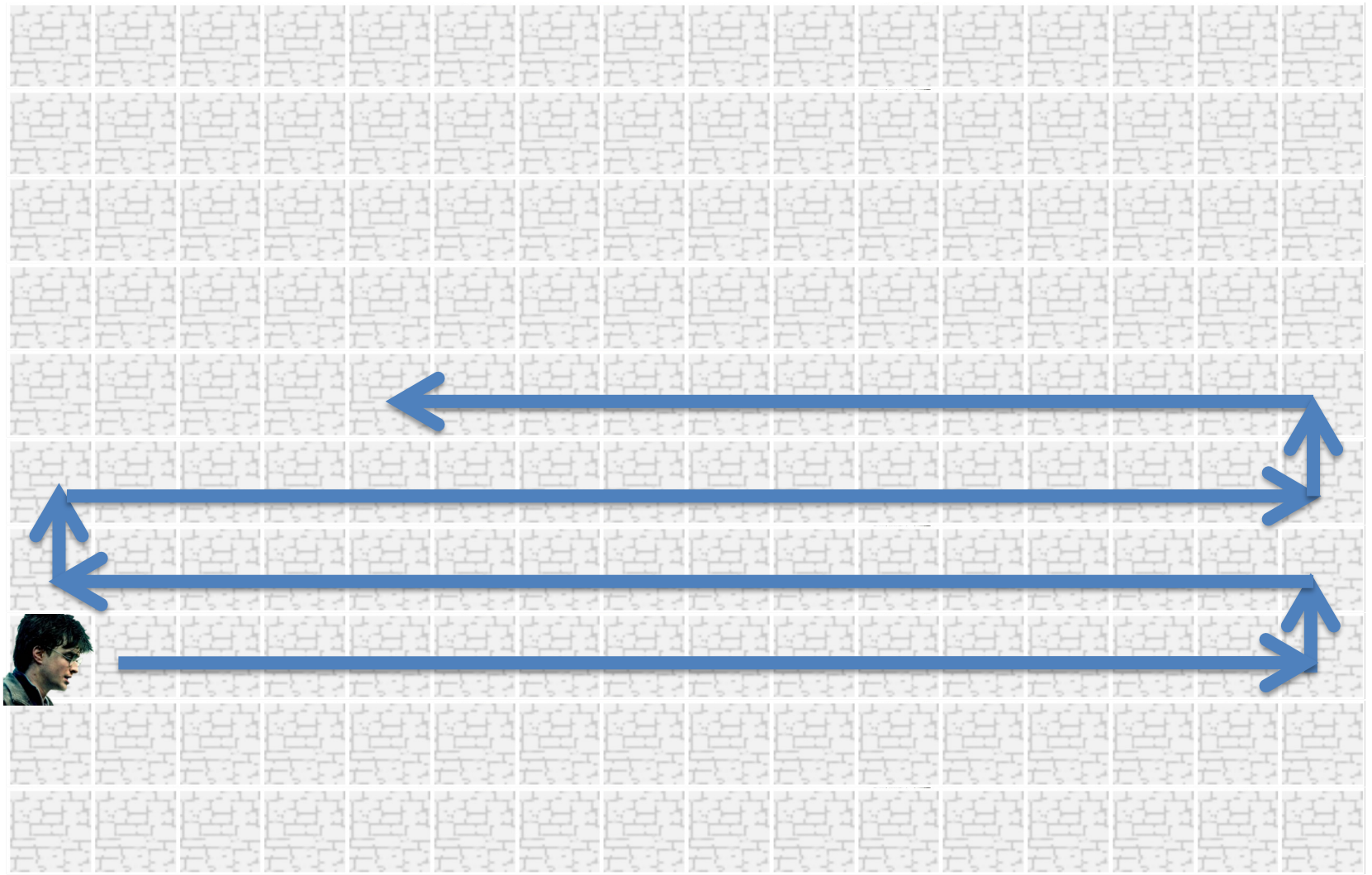



not is_on_target()

```
while is_on_target() == False:
    if can_move():
        move()
    else:
        turn_left()
        move()
        turn_right()
        move()
        move()
        turn_left()
        move()
        turn_right()
```







# Quel est votre objectif ?

- Des programmes 
  - Rapides
  - Ecrits rapidement
  - Les plus courts possibles
  - Les plus obscurs possibles

```
B=print
A=5
while A>=0:
	if A!=0:B(A)
	else:B('Decollage')
	A=A-1
B('Reussi')
```

*https://python-minifier.com/*

- Des programmes 
  - Corrects
    - Doivent résoudre correctement le problème posé (même si celui-ci est simple)
  - Clairs
    - Doivent être lisible et documenté
  - Testés
    - Vous devez pouvoir vérifier les limites de votre programme