



UNIVERSITETET I BERGEN

LABYRINT

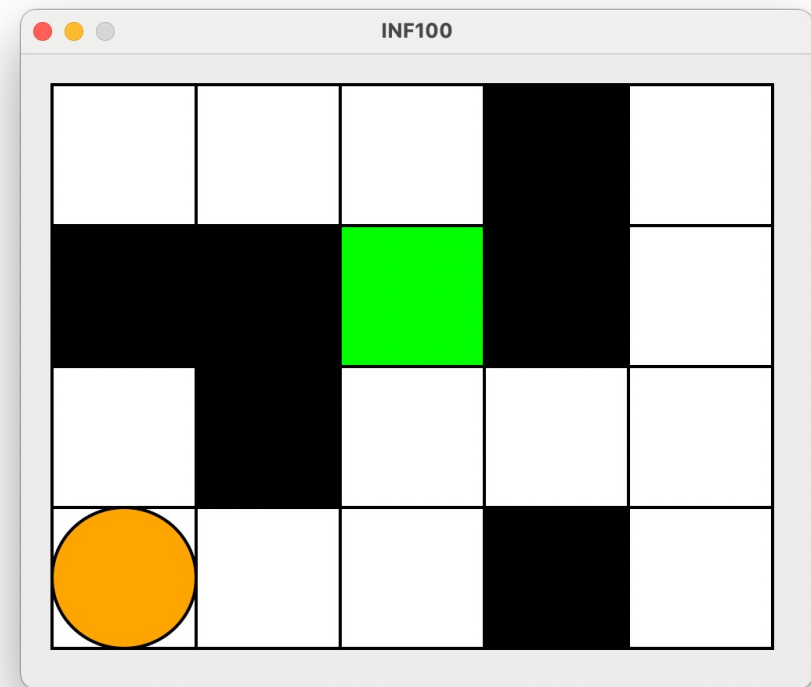
INF100

VÅR 2024

Torstein Strømme og Odin Hoff Gardå

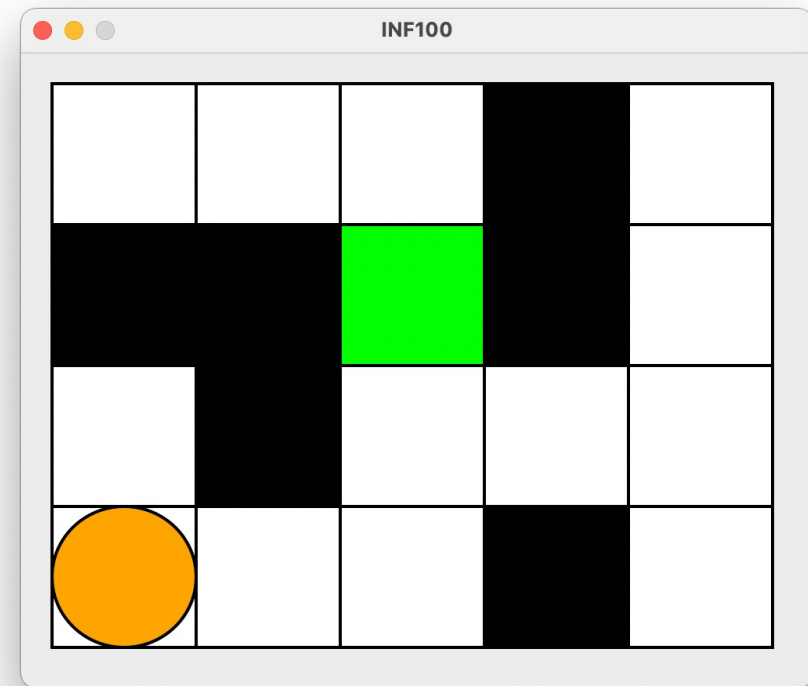
I DAG

- Vi skal lage et labyrintspill sammen
- Egentlig en forenklet versjon av Snake



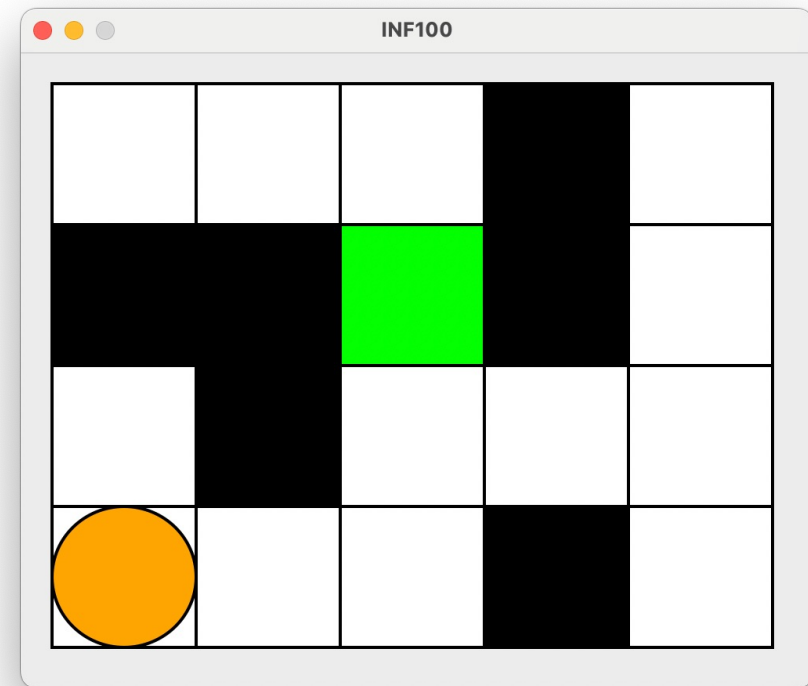
STEG 1: MODELLEN

- Last ned starter-kode fra lab8
- I `laby_maze.py` : `init(app)` opprett en modell
 - Hvilke variabler trenger vi?
 - Hva skal variablene inneholde?



STEG 1: MODELLEN

- Last ned starter-kode fra lab8
- I `laby_maze.py` : `init(app)` opprett en modell
 - Hvilke variabler trenger vi?
 - `app.maze` -> 2D-liste med tall
 - `app.agent_pos` -> tuple med (row, col)
 - Hva skal variablene inneholde?



STEg 2: VISNING

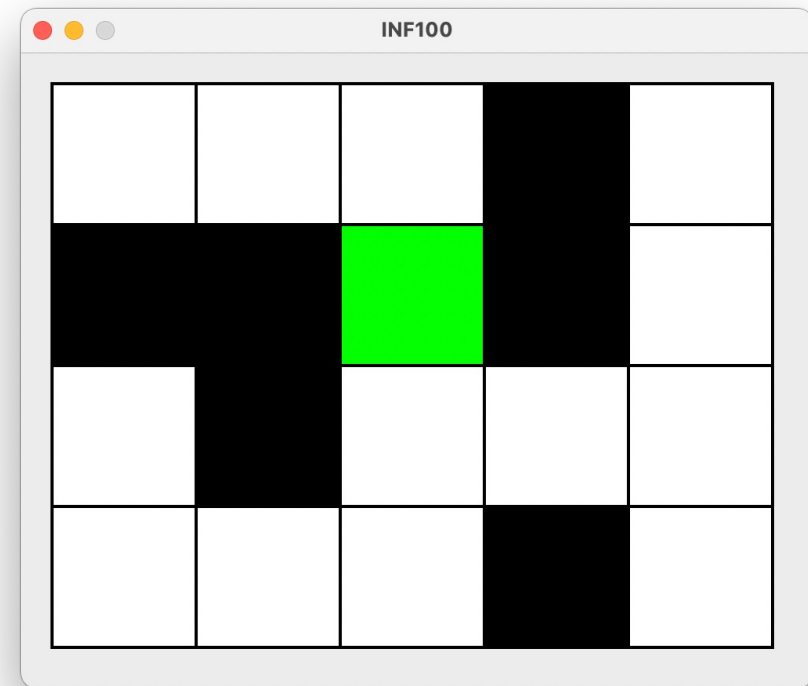
a) Tegn labyrinten

```
laby_maze.py : init
```

```
app.maze = [  
    [0, 0, 0, 1, 0],  
    [1, 1, 2, 1, 0],  
    [0, 1, 0, 0, 0],  
    [0, 0, 0, 1, 0],  
]
```

```
laby_view.py
```

```
def draw_maze(app, canvas, x1, y1, x2, y2):  
    ...
```



STEP 2: VISNING

- a) Tegn labirinten
- b) Tegn agenten

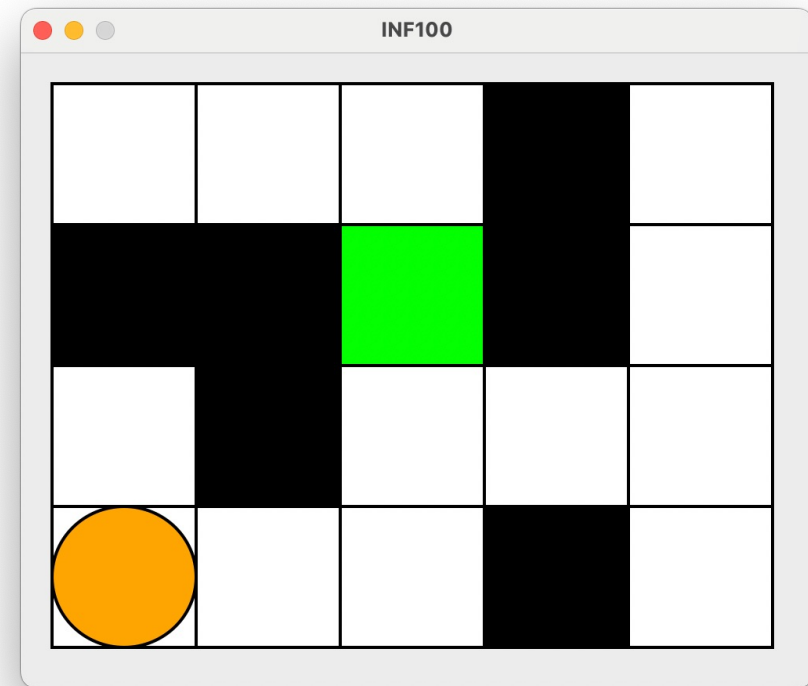
laby_maze.py : init

```
app.maze = [  
    [0, 0, 0, 1, 0],  
    [1, 1, 2, 1, 0],  
    [0, 1, 0, 0, 0],  
    [0, 0, 0, 1, 0],  
]
```

```
app.agent_pos = (3, 0)
```

laby_view.py

```
def draw_maze(app, canvas, x1, y1, x2, y2):  
    ...
```



STEG 3: FLYTT AGENT TILFELDIG

- Når brukeren trykker på **Escape** skal agenten flyttes til en tilfeldig ledig rute på brettet.



```
laby_maze.py
```

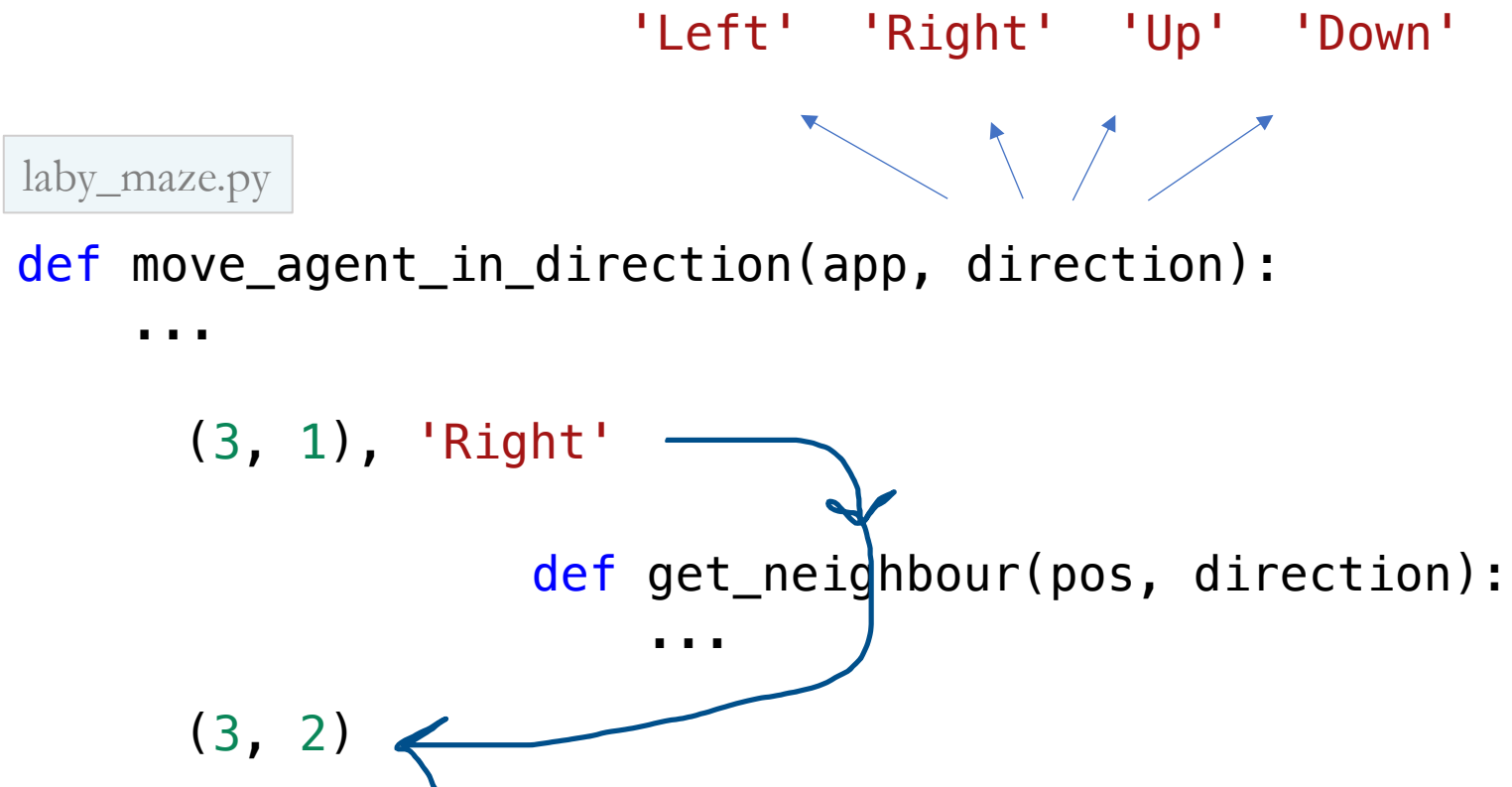
```
def move_agent_to_random_free_position(app):  
    ...
```

```
laby_main.py : key_pressed
```

```
if event.key == 'Escape':  
    ...
```

STEG 4: FLYTT AGENT MED PILTASTER

- Når brukeren trykker på piltastene skal agenten flyttes tilsvarende



STEG 4: FLYTT AGENT MED PILTASTER

- Når brukeren trykker på piltastene skal agenten flyttes tilsvarende

laby_maze.py

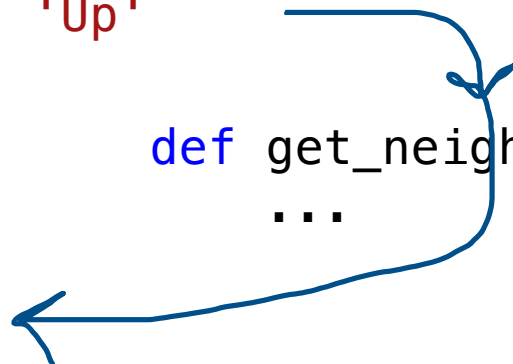
```
def move_agent_in_direction(app, direction):  
    ...
```

(3, 1), 'Up'

```
def get_neighbour(pos, direction):  
    ...
```

(2, 1)

'Left' 'Right' 'Up' 'Down'



STEG 4: FLYTT AGENT MED PILTASTER

- Når brukeren trykker på piltastene skal agenten flyttes tilsvarende
- Utfør flyttet kun dersom det er et lovlig trekk

```
laby_maze.py
```

```
def move_agent_in_direction(app, direction):  
    ...
```

```
def is_free(maze, pos):  
    ...
```

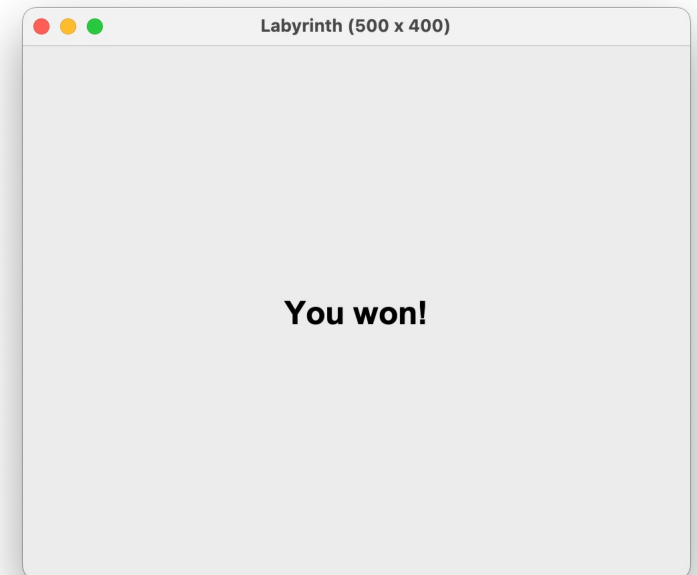
```
def is_within_bounds(maze, pos):  
    ...
```

STEG 5: VIS VINNER-SKJERM

- Når brukeren flytter agenten i mål, vis en vinner-skjerm

```
laby_maze.py
```

```
def is_won(app):  
    ...
```



STEG 6: DUMMY AI

- Hver gang vi trykker på **m**, flytt agenten en tilfeldig retning.

```
laby_ai.py
```

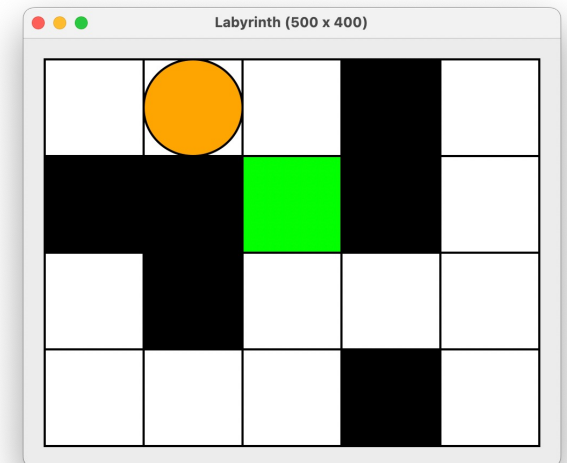
```
def get_direction(app):  
    ...
```

STEG 6: DUMMY AI

- Opprett «auto-modus» hvor agenten flyttes tilfeldig periodisk
- Slå av og på auto-modus med **a**

```
laby_main.py
```

```
def timer_fired(app):  
    ...
```

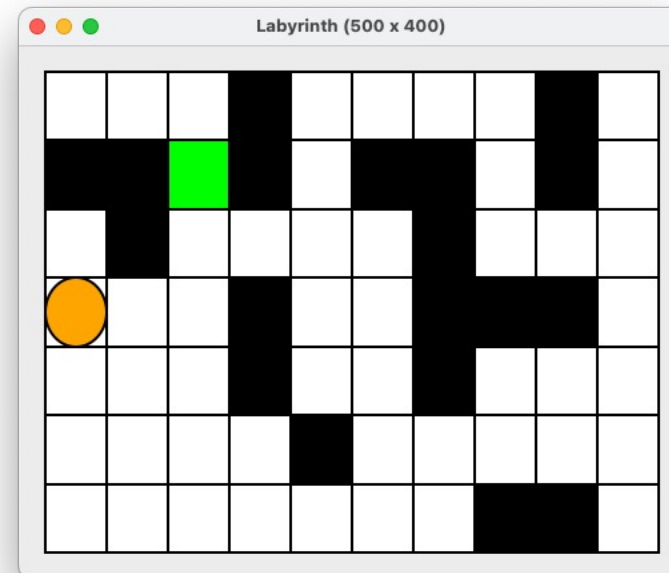


STEG 7: LESE BRETT FRA FIL

- Last labyrinten fra en tekstfil
- (Valgfritt) initier hele programmet på nytt med ny tekstfiler nå bruker trykker på tall-tastene 1-3

level2.lev

```
0001000010  
1121011010  
0100001000  
0001001110  
0001001000  
0000100000  
0000000110
```



NESTE STEG: SMARTERE AI

- For neste uke
- Sneak peak:

```
app.maze = [  
  [0, 0, 0, 1, 0],  
  [1, 1, 2, 1, 0],  
  [0, 1, 0, 0, 0],  
  [0, 0, 0, 1, 0],  
]
```

```
app.q_table = [  
  [*, *, *, *, *],  
  [*, *, *, *, *],  
  [*, *, *, *, *],  
  [*, *, *, *, *],  
]
```

laby_ai.py

```
{  
  'Left': 0.1,  
  'Right': 0.9,  
  'Up': 0.9,  
  'Down': -1.0,  
}
```

*høy verdi: god erfaring
med å gjøre dette valget*

*lav verdi: dårlig erfaring
med å gjøre dette valget*

```
def get_direction(app):  
  next_moves = # [moves w/max q-val]  
  app.agent_pos = random.choice(next_moves)
```