



UNIVERSITETET I BERGEN

LABYRINT

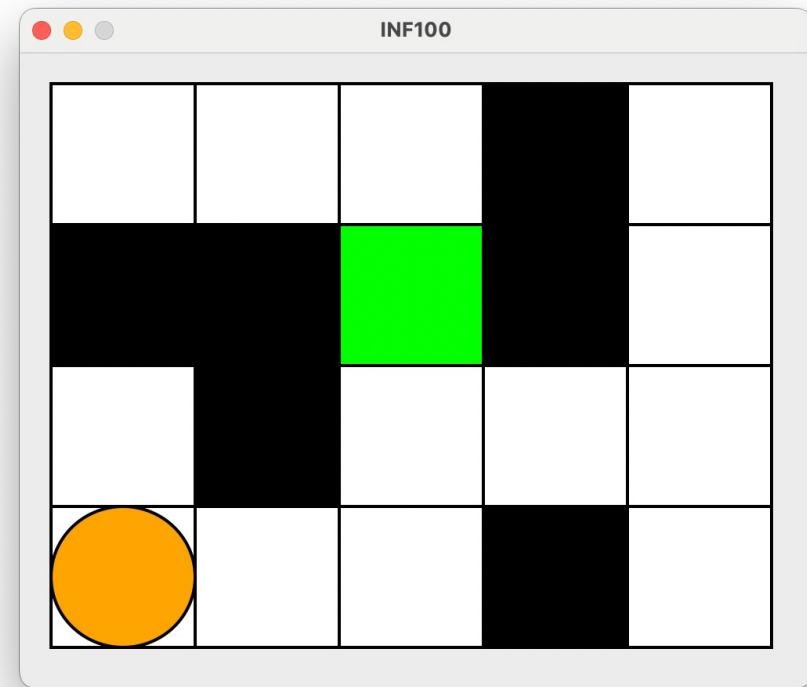
INF100

VÅR 2024

Torstein Strømme og Odin Hoff Gardå

I DAG

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

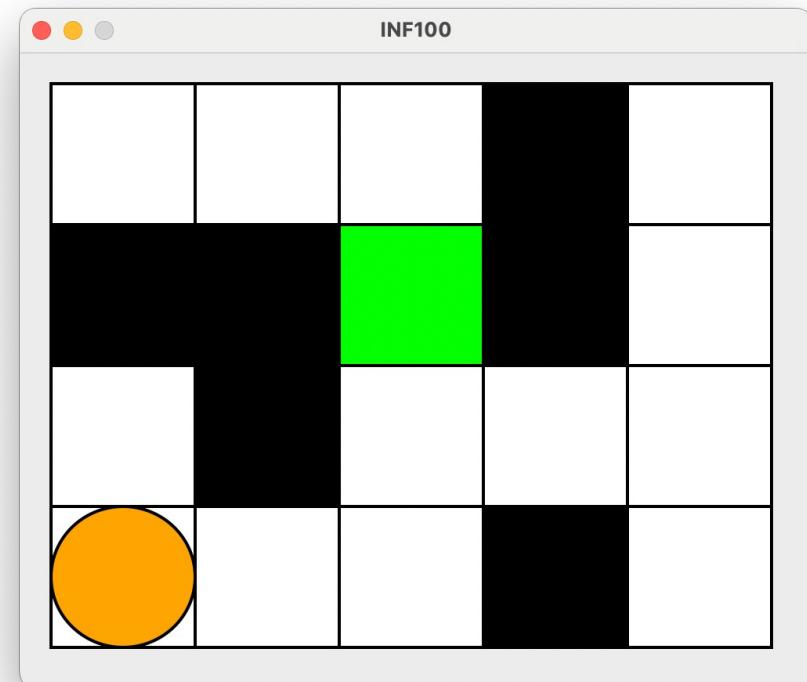


STEG 1: MODELLEN

- Last ned starter-kode fra lab8
- I laby_maze.py : init(app) opprett en modell

- Hvilke variabler trenger vi?

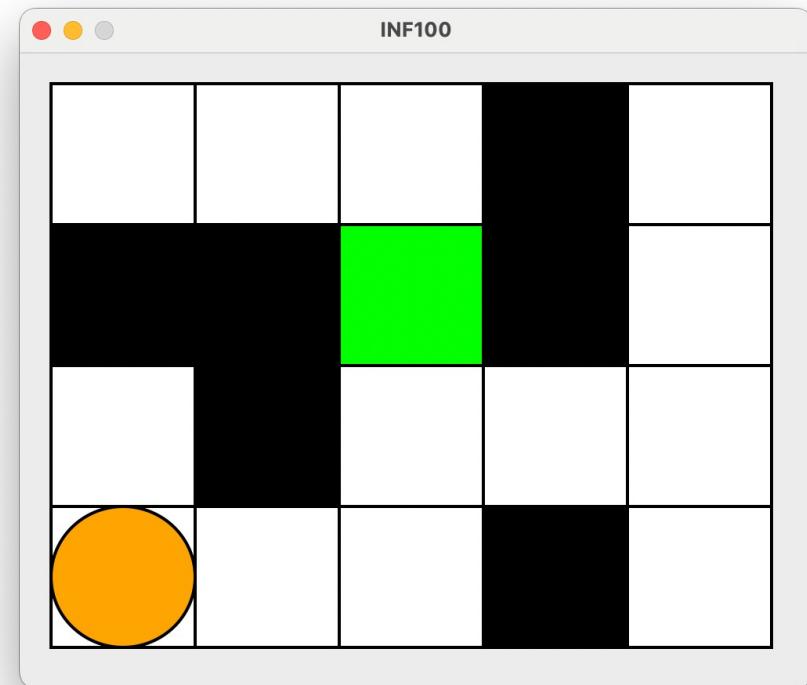
- Hva skal variablene inneholde?



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 labrinten

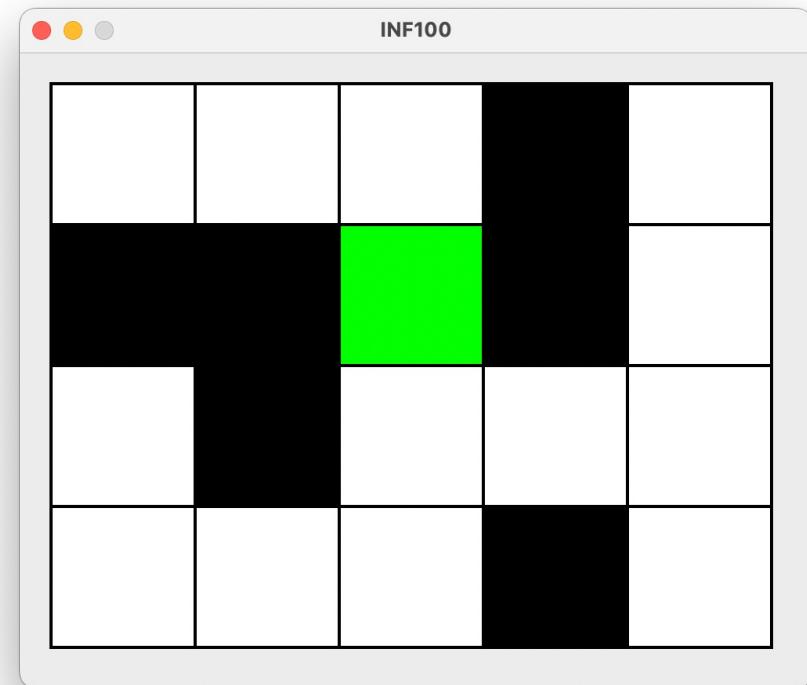
laby_maze.py : init

```
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):  
    ...
```



STEG 2: VISNING

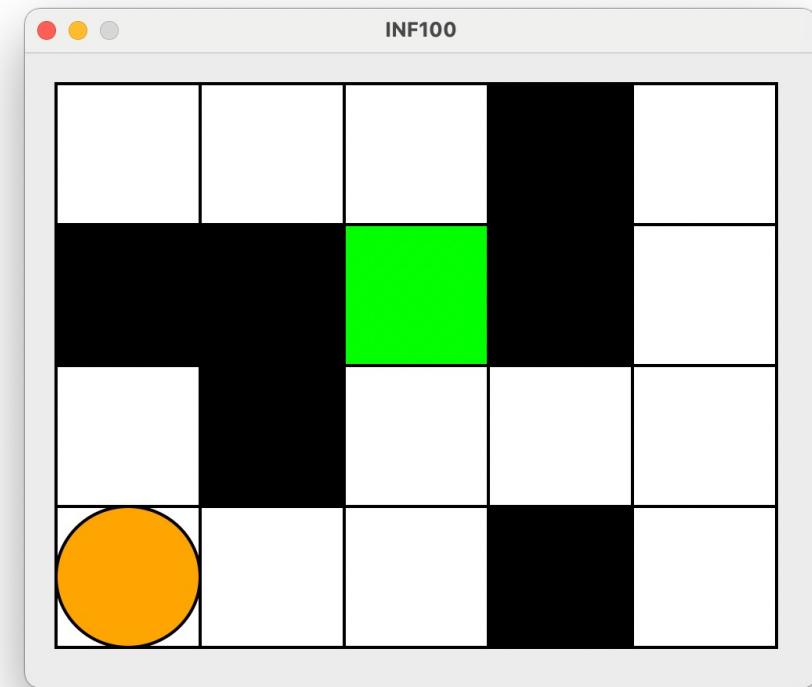
- a) Tegn labrinten
- b) Tegn agenten

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)
```

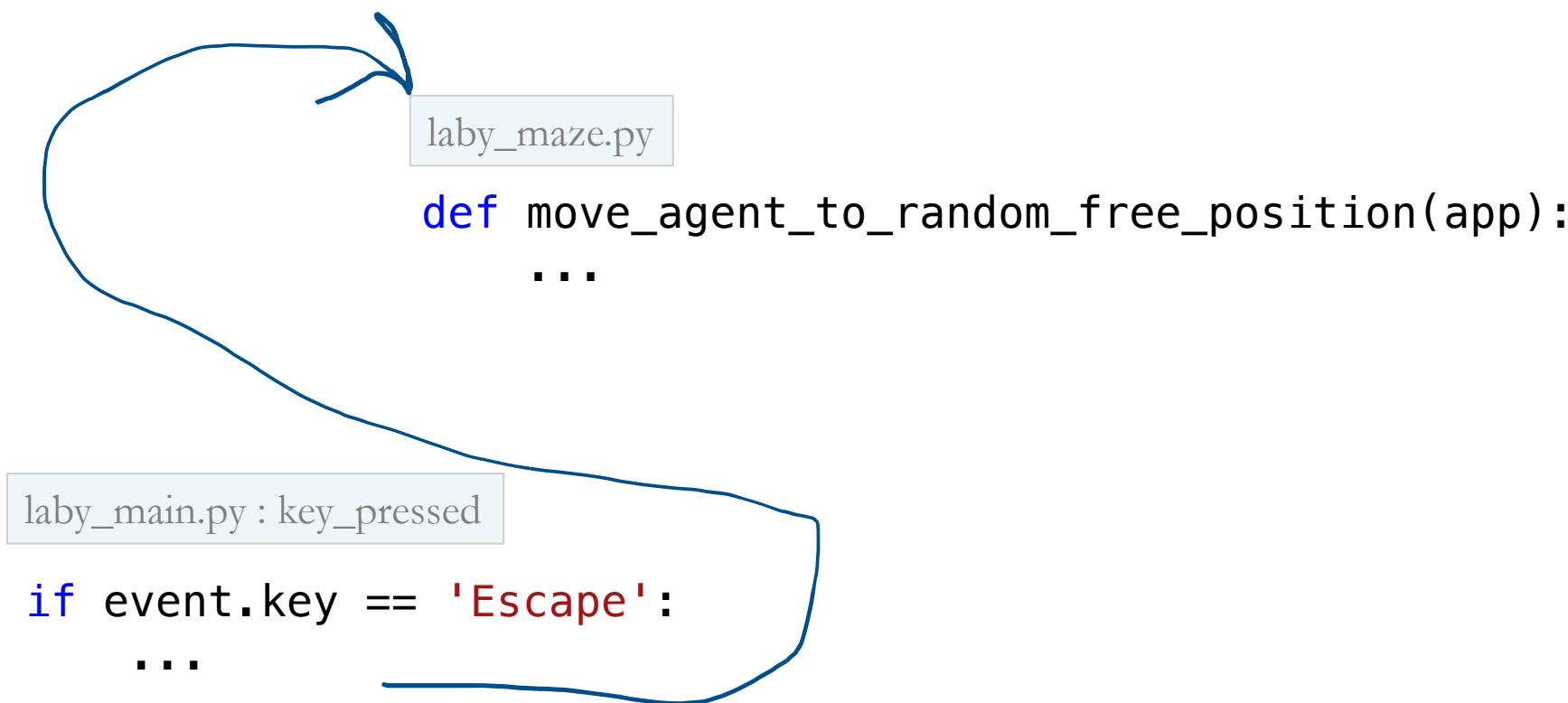
laby_view.py

```
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.



STEG 4: FLYTT AGENT MED PILTASTER

- Når brukeren trykker på piltastene skal agenten flyttes tilsvarende

The diagram illustrates a callout from a code snippet to a maze diagram. The code snippet is contained within a light gray box labeled "laby_maze.py". It shows a function definition:

```
def move_agent_in_direction(app, direction):
    ...
    (3, 1), 'Right'
```

Below the code, a blue callout arrow originates from the position "(3, 1)" and points to a small square on a grid representing a maze. The grid has four squares per row and four rows per column. The square at (3, 1) is highlighted in green. Above the grid, four arrows point outwards from the center: a blue downward-pointing arrow labeled "Down", a blue leftward-pointing arrow labeled "Left", a blue upward-pointing arrow labeled "Up", and a blue rightward-pointing arrow labeled "Right". The text "def get_neighbour(pos, direction):" and "..." are also present near the bottom of the callout.

STEG 4: FLYTT AGENT MED PILTASTER

- Når brukeren trykker på piltastene skal agenten flyttes tilsvarende



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

laby_maze.py

```
def move_agent_in_direction(app, direction):
    ...
    (3, 1), 'Up'
    def get_neighbour(pos, direction):
        ...
(2, 1) ←
```

The diagram illustrates a 4x4 grid with arrows indicating movement directions: 'Left', 'Right', 'Up', and 'Down'. A callout from the 'Up' direction points to a specific cell at position (3, 1). Another callout from the 'Up' direction points to a function definition for 'get_neighbour'. A blue arrow points from the cell (2, 1) to the left, indicating the 'Left' direction.

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):
    ...
```

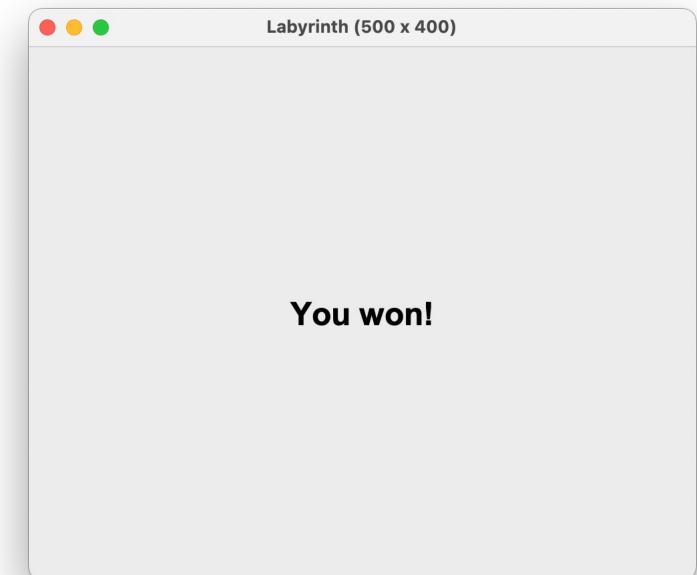
```
def is_within_bounds(maze, pos):
    ...
```

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

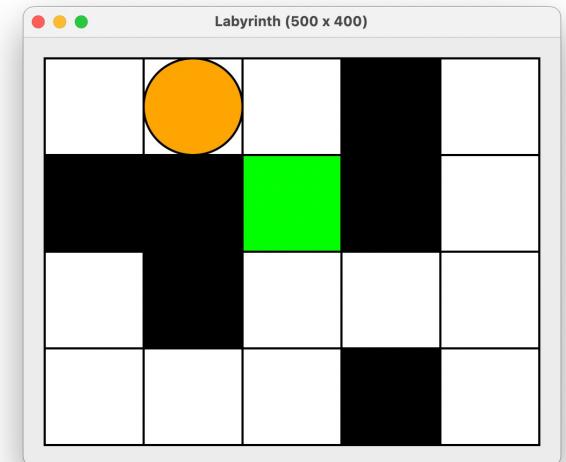
```
def get_direction(app):
    ...
```

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):  
    ...
```

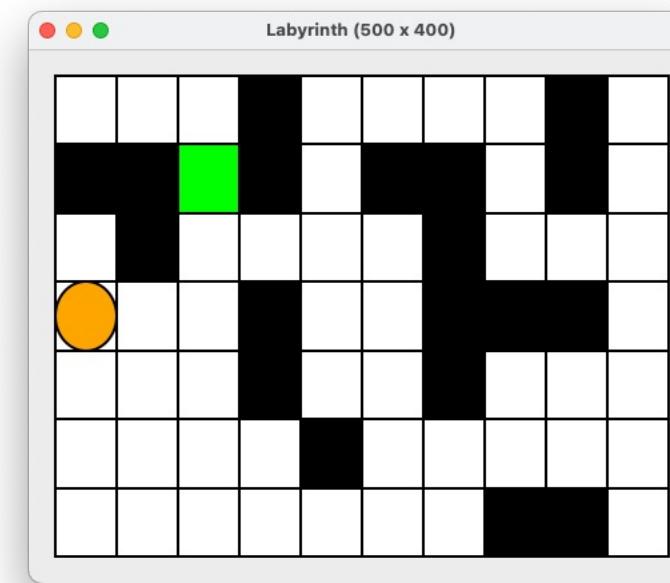


STEG 7: LESE BRETT FRA FIL

- Last labyrinten fra en tekstfil
- (Valgfritt) initier hele programmet på nytt med ny tekstmater 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)
```