Welcome !

2024 Fall CS101 Introduction to Programming

KAIST School of Computing

Week 2

Programming with robots
(Conditional expressions, While loops)

If-statement & While-statement

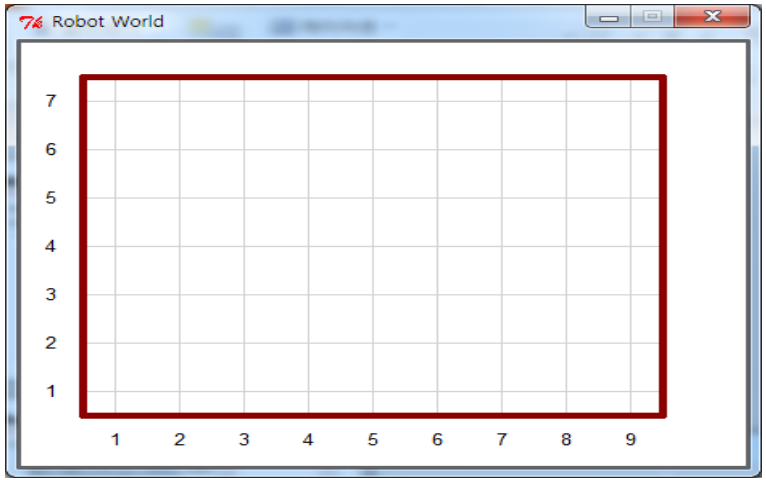
New functions

- Create a custom world

create_world(streets = 7, avenues = 9)

✓ ‘streets’ for the number of rows

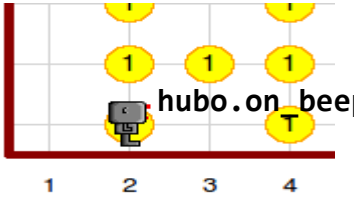
✓ ‘avenues’ for the number of columns



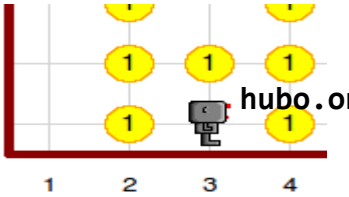
New functions

- Can check if there is a beeper

```
hubo.on_beeper()
```




```
hubo.on_beeper() == True
```

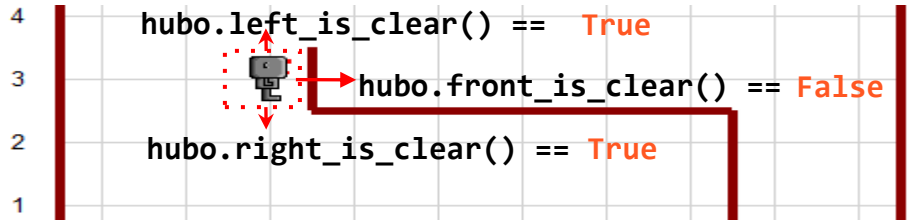



```
hubo.on_beeper() == False
```

New functions

- Can check if there is a wall on each of the three sides

```
hubo.front_is_clear()
hubo.left_is_clear()
hubo.right_is_clear()
```



If statements

- **If** statements **sequentially** checks the conditionals

```
if conditional_expression_1:
   works to do when conditional_expression_1 evaluates to True

elif conditional_expression_2:
   works to do when conditional_expression_1 evaluates to False &
                    conditional_expression_2 evaluates to True

elif conditional_expression_3:
   works to do when conditional_expression_1 evaluates to False &
                   conditional_expression_2 evaluates to False &
                    conditional_expression_3 evaluates to True
...
else:
   works to do when all the above conditions are False
```

If statements - Example

```
score = 50
if score < 60:
    print('You got F grade')
elif score < 70:
    print('You got C grade')
elif score < 80:
    print('You got B grade')
else:
    print('You got A grade')
```

Guess what will be output

- when score = 55
- when score = 65
- when score = 70
- when score = 85

respectively?

While loops

- **while** statement loops until the **conditional** evaluates to true

```
while conditional_expression:
    works to do while conditional_expression evaluates to True
```

- Example

```
n = 0
while n < 5:
    print(n)
    n = n + 1
```

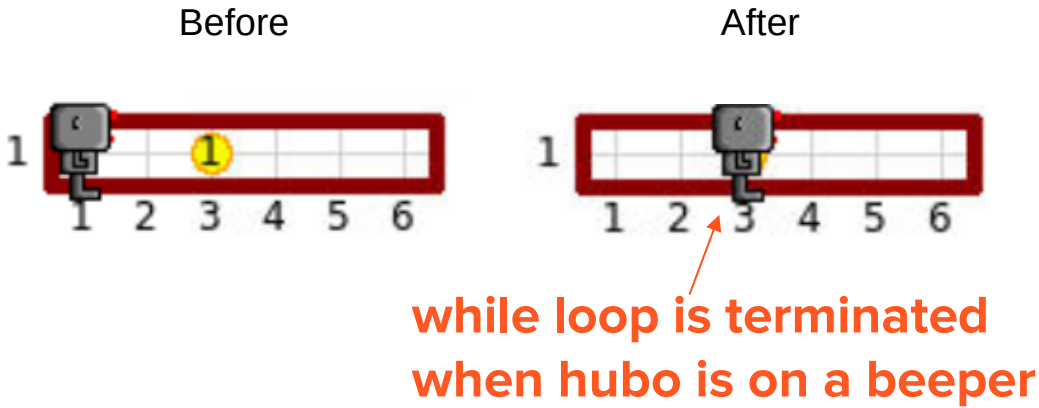
Result)
0
1
2
3
4

Break and Continue

- **break** statement terminate the loop statement regardless of the conditional expression

- Example

```
while True:
    hubo.move()
    if hubo.on_beeper():
        break
```



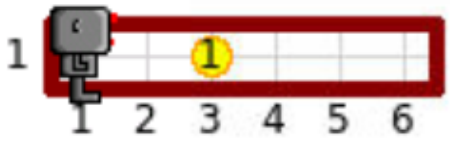
Break and Continue

- **continue** statement skips the current loop and goes to the next loop

- Example

```
while hubo.front_is_clear():
    hubo.move()
    continue
    hubo.turn_left()
```

Before



After

Never executed because of continue

Week 2
Today’s Tasks

Tasks for Today!

- Read sections 10~13 in the robot notes

Five not so simple tasks

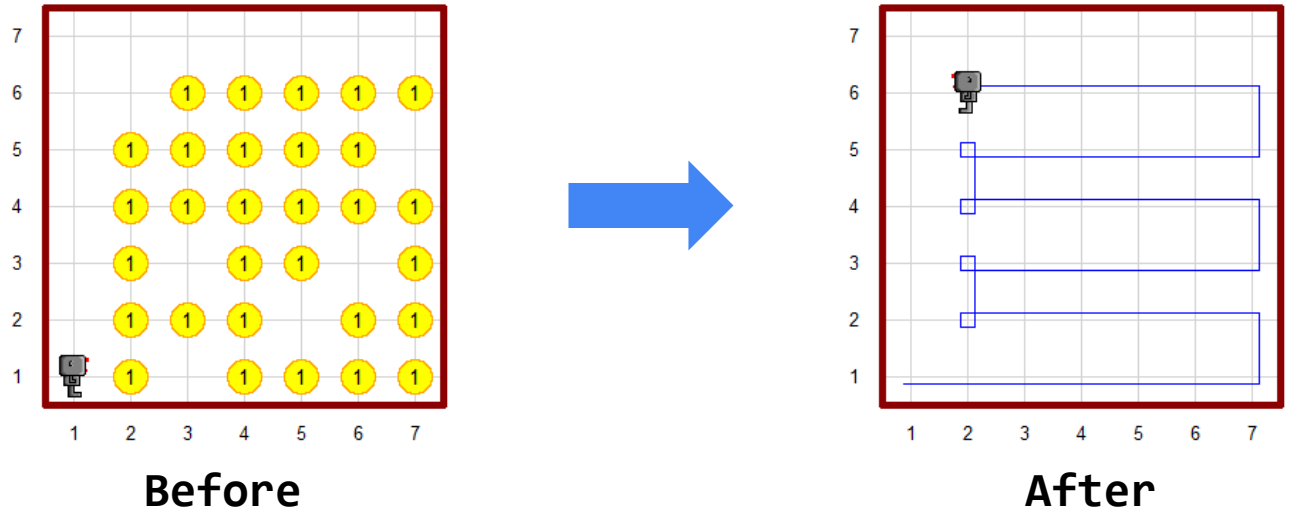
- Harvest More (page 6)
- Plant (page 7)
- Smart Hurdles (page 8 & 9)
- Harvest Even More (page 10)
- Smart ZigZag (page 10)

- When you have completed all the tasks, let a TA mark you off

Task 1 | Conditionals – Harvest More

- Modify your program from the ‘Harvest Again’ task (Week 01) so that it works for *harvest3.wld*

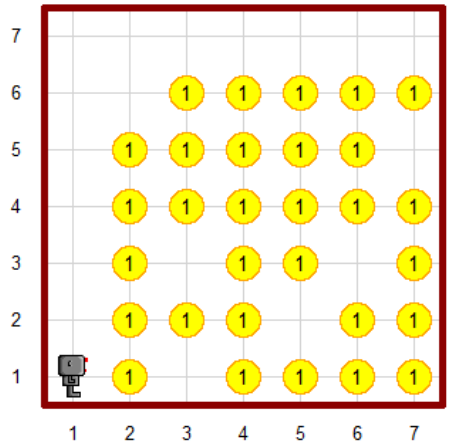
```
load_world(“worlds/harvest3.wld”)
```



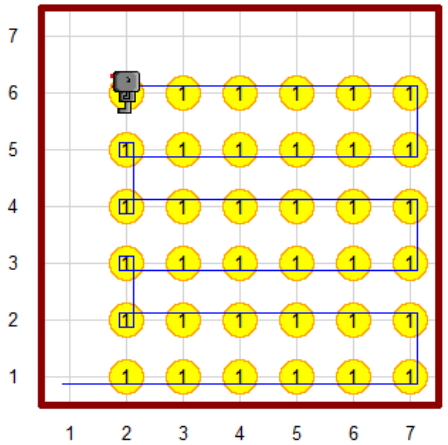
Task 2 | Conditionals – Plant

- Write a program so that Hubo plants beepers in empty spots
- The finished screen should look like “harvest1.wld”

```
load_world(“worlds/harvest3.wld”)
```



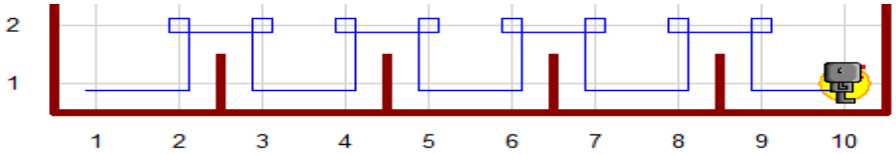
Before



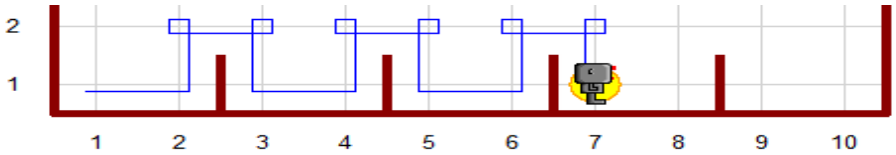
After

Task 3 | Conditionals – Smart Hurdles

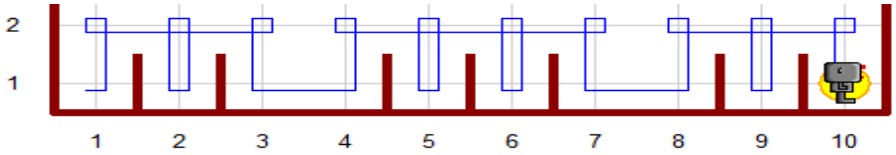
- Write **jump_one_hurdle()** in section 11
  - move_jump_or_finish() should be able to handle all three maps, “hurdles1.wld”, “hurdles2.wld” and “hurdles3.wld”. Check it yourself.
- Write a new program (similar to Hurdles3 in section 11) that uses a while loop. DO NOT USE a for-loop of fixed length
  - It should also work for all three hurdles



hurdles1.wld



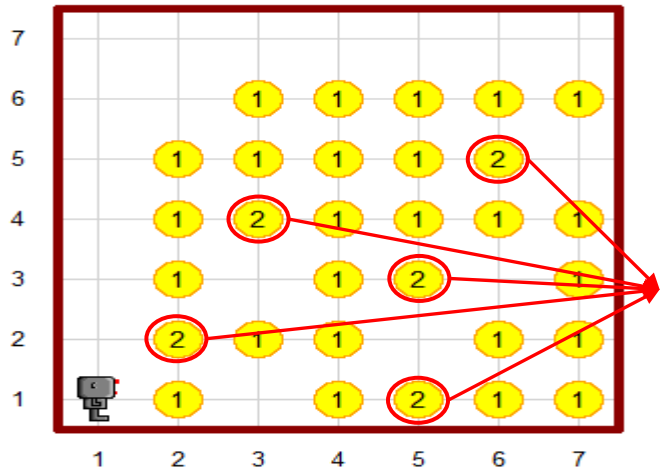
hurdles2.wld



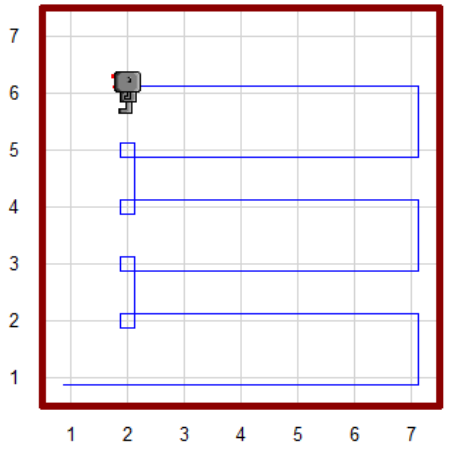
hurdles3.wld

Task 4 | While loop – Harvest Even More

- Modify Harvest More task
  - It should work even when there are more than one beeper on a spot (“harvest4.wld”)
  - It should also work for the previous worlds (“harvest1.wld” and “harvest3.wld”)



Have to harvest all carrots on one place



Task 5 | While loop – Smart ZigZag

- Rewrite ZigZag program
  so that the robot can visit every spot in an empty world
  of any size in zigzag fashion
  - It should work for even and odd numbers of streets and avenues
  - Hint: You can use **break** to terminate a loop

m
(streets,
number of
rows)

n
(avenues,
# of columns)

m and **n** can be any integer
except for **m**=1 or **n**=1

questions?