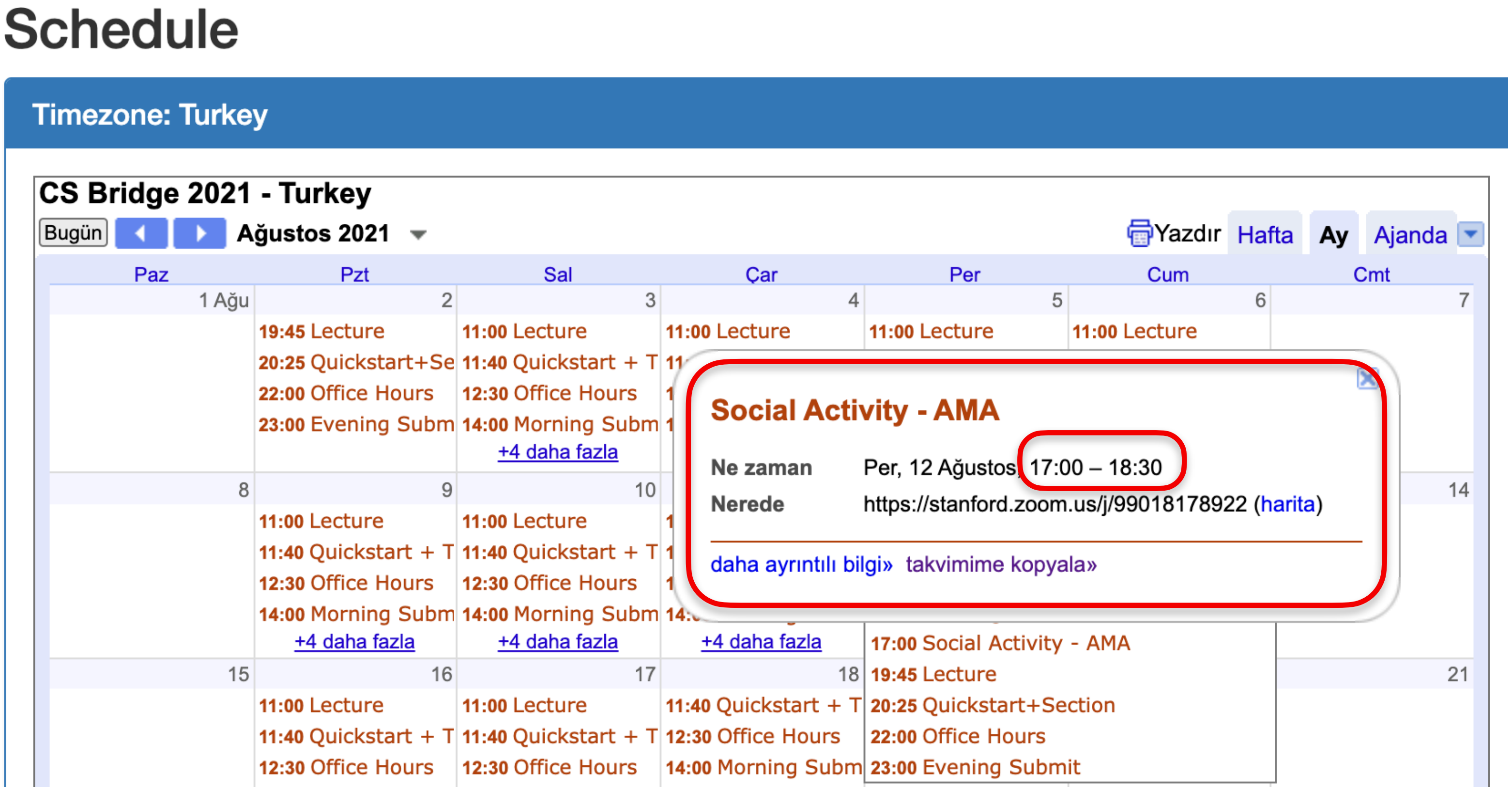
CS Bridge, Lecture 15
Breakout - Extra Features

Reminder before we start



Schedule

Timezone: Turkey

CS Bridge 2021 - Turkey

Paz	Pzt	Sal	Çar	Per	Cum	Cmt
1 Ağu	2 19:45 Lecture 20:25 Quickstart+Se 22:00 Office Hours 23:00 Evening Subm	3 11:00 Lecture 11:40 Quickstart + T 12:30 Office Hours 14:00 Morning Subm +4 daha fazla	4 11:00 Lecture 11: 1 1	5 11:00 Lecture	6 11:00 Lecture	7
8	9 11:00 Lecture 11:40 Quickstart + T 12:30 Office Hours 14:00 Morning Subm +4 daha fazla	10 11:00 Lecture 11:40 Quickstart + T 12:30 Office Hours 14:00 Morning Subm +4 daha fazla	1 1 1 14: +4 daha fazla	17:00 Social Activity - AMA 19:45 Lecture 20:25 Quickstart+Section 22:00 Office Hours 23:00 Evening Submit		14
15	16 11:00 Lecture 11:40 Quickstart + T 12:30 Office Hours	17 11:00 Lecture 11:40 Quickstart + T 12:30 Office Hours	18 11:40 Quickstart + T 12:30 Office Hours 14:00 Morning Subm			21

Social Activity - AMA

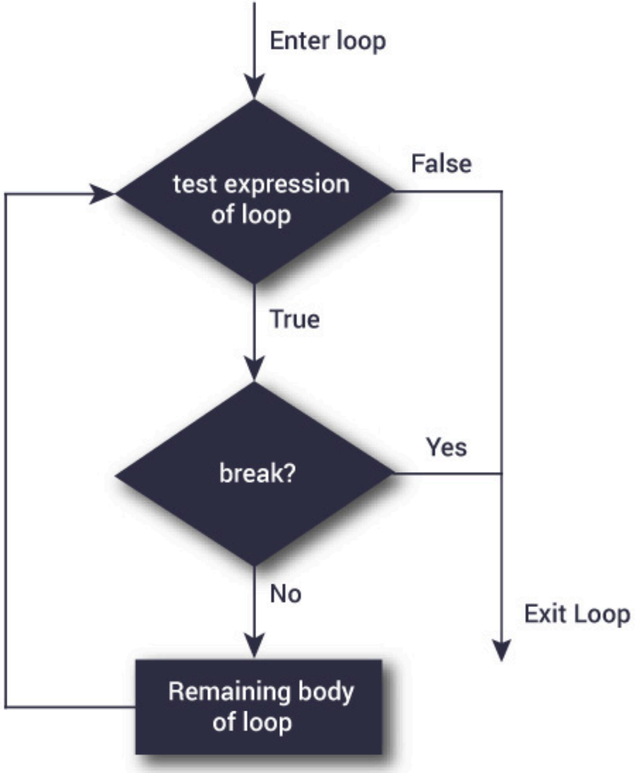
Ne zaman Per, 12 Ağustos, 17:00 – 18:30

Nerede https://stanford.zoom.us/j/99018178922 (harita)

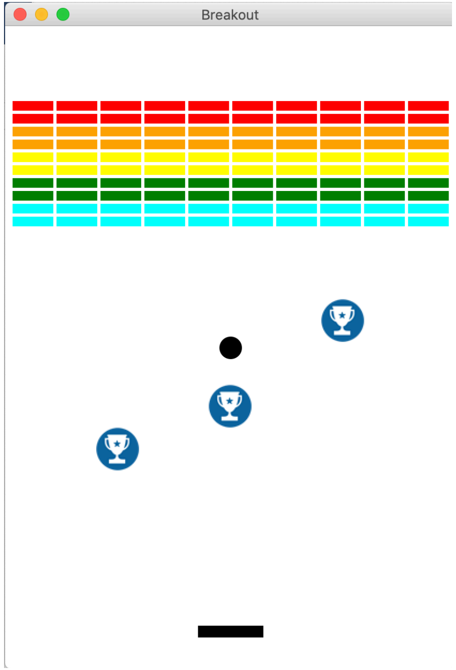
daha ayrıntılı bilgi» takvimime kopyala»

Before we start, a few points to consider

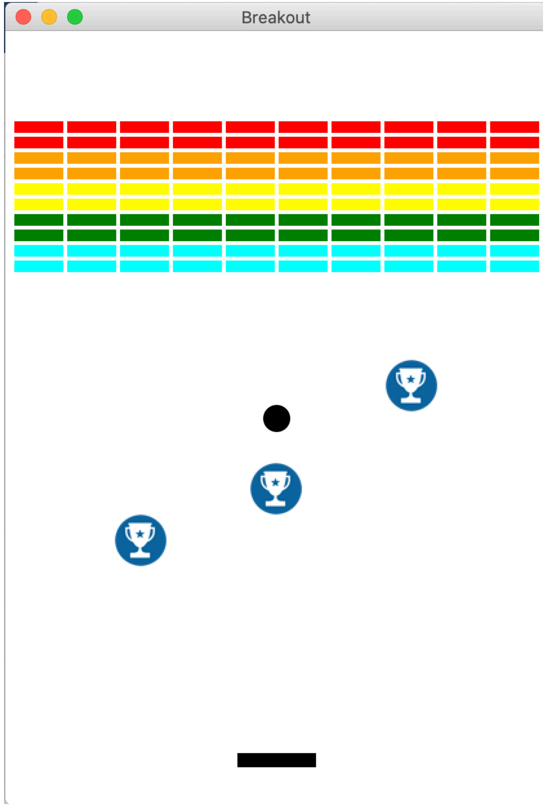
Breaking a loop



Creating a list of objects in a function and returning them



Creating a list of objects in a function and returning them



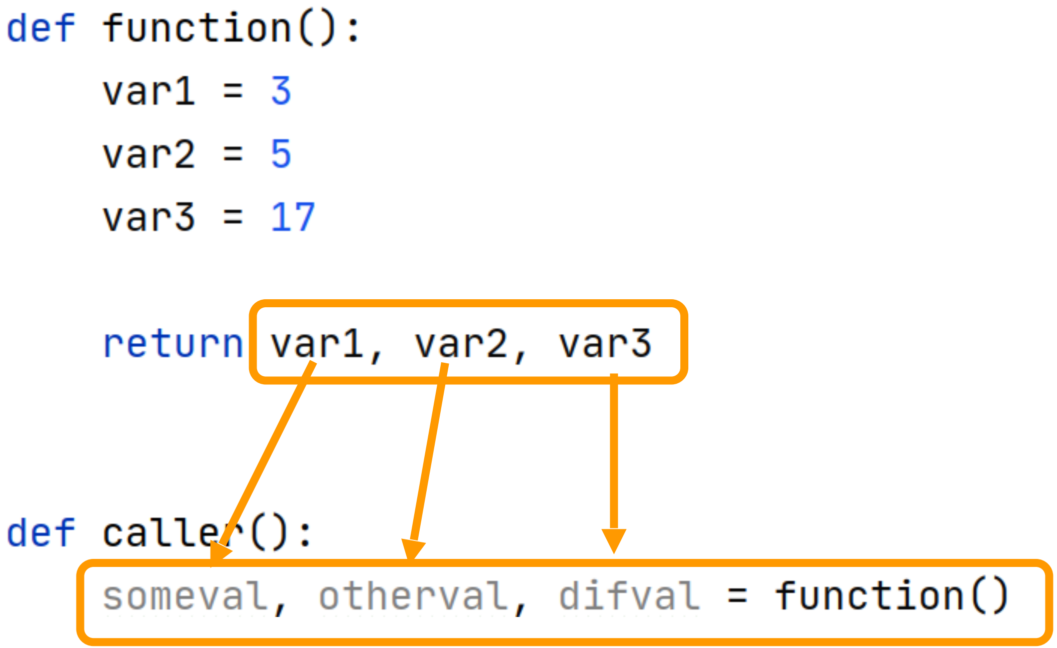
Play time

Returning graphical objects from functions

```
# A function to create a single graphical object and return
def setup_graphical_object(canvas):
    object_x = … <some expression to compute x> …
    object_y = … <some expression to compute y> …
    my_object = canvas.create_….(object_x, object_y, object_x+SIZE, object_y+SIZE)
    return my_object
```

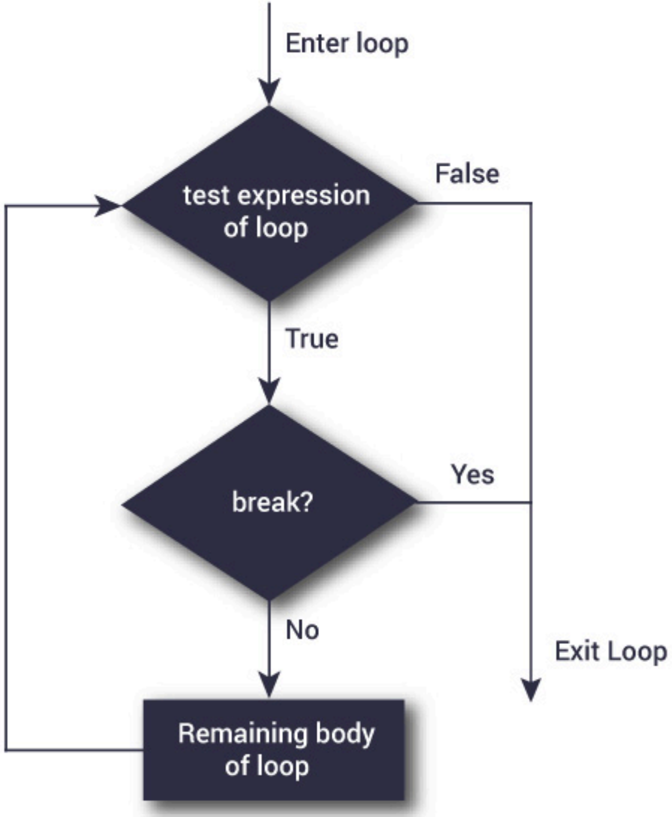
```
# A function to create multiple graphical objects, put those in a list and return the list of objects
def setup_graphical_objects(canvas):
    objects_list = []
    for i in range(NUM_OBJECTS):
        object_x = … <some expression to compute x> …
        object_y = … <some expression to compute y> …
        objects_list.append(canvas.create_….(object_x, object_y, object_x+SIZE, object_y+SIZE))
    return objects_list
```

Returning multiple values from functions



```python
def function():
    var1 = 3
    var2 = 5
    var3 = 17

    return var1, var2, var3


def caller():
    someval, otherval, difval = function()
```

Breaking a loop



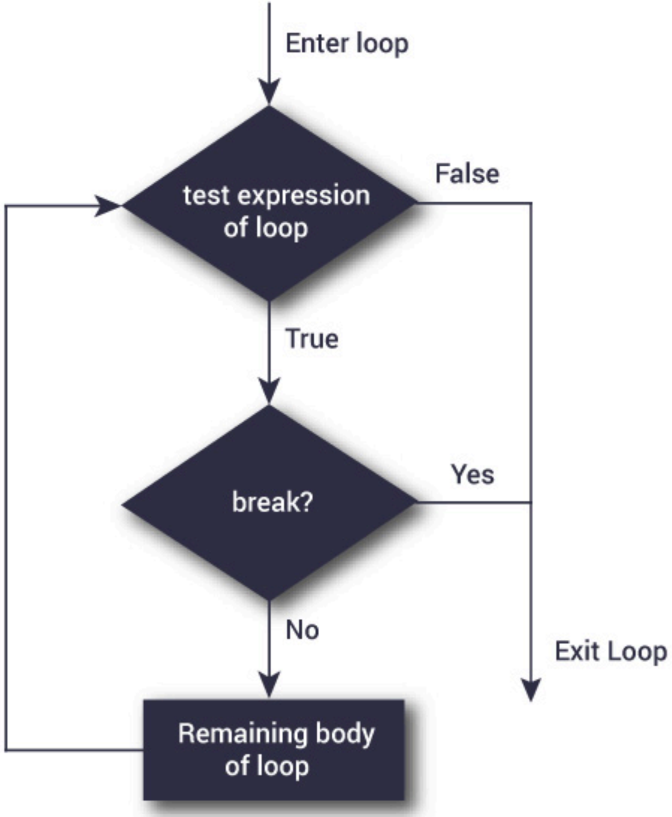
```
# A script that takes some time: check-prime
import time
start_time = time.time()
n = 151153234

is_prime = True
for i in range(2, n):
    if n % i == 0:
        is_prime = False

if is_prime:
    print("Is prime")
else:
    print("Is not prime")

print("--- {} seconds ---".format(time.time() - start_time))
```

Better to stop the loop here

Output:

```
Is not prime
--- 9.443579196929932 seconds ---
```

Breaking a loop



```
# A script that takes some time: check-prime
import time
start_time = time.time()
n = 151153234

is_prime = True
for i in range(2, n):
    if n % i == 0:
        is_prime = False
        break
if is_prime:
    print("Is prime")
else:
    print("Is not prime")

print("--- {} seconds ---".format(time.time() - start_time))
```

We add a break statement

Output:

```
Is not prime
--- 3.123283386230469e-05 seconds ---
```

Breaking an infinite loop

Reading positive integers from a user:

```
user_ints= []
value = int(input("Enter an integer:"))
while value > 0:
    user_ints.append(value)
    value = int(input("Enter an integer:"))
print("Your inputs:" + str(user_ints))
```

Output:

```
Enter an integer:1
Enter an integer:2
Enter an integer:3
Enter an integer:-8
Your inputs:[1, 2, 3]
```

```
user_ints= []
while True:
    value = int(input("Enter an integer:"))
    if value > 0:
        user_ints.append(value)
    else:
        break
print("Your inputs:" + str(user_ints))
```

Output:

```
Enter an integer:3
Enter an integer:2
Enter an integer:4
Enter an integer:5
Enter an integer:-7
Your inputs:[3, 2, 4, 5]
```

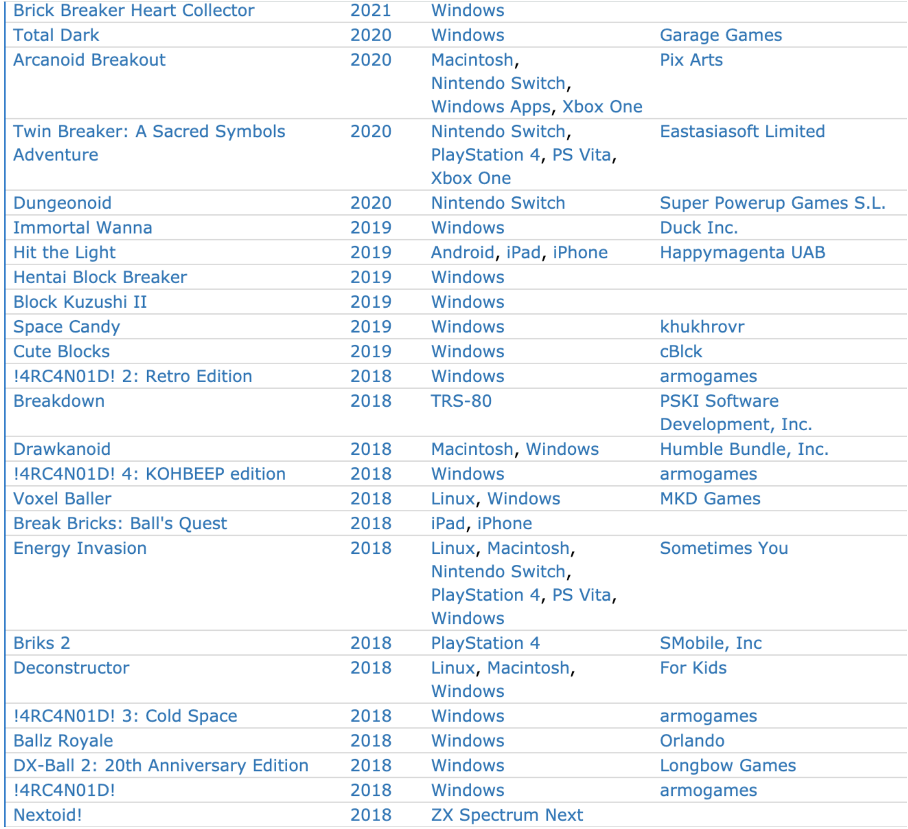
Breaking a loop in Breakout

You may like to implement the paddle move by key-strokes and collect objects.
In case of multiple colliding object, you may prefer to remove only one of the colliding objects

```
<list of colliding objects> = canvas.find_overlapping(<paddle coordinates>)

for collider in <list of colliding objects>:
  if collider in <list of objects paddle can remove>:
      canvas.delete(collider)
      break
```

Checking if colliding object is one of the objects in the list

Extra Features for Breakout



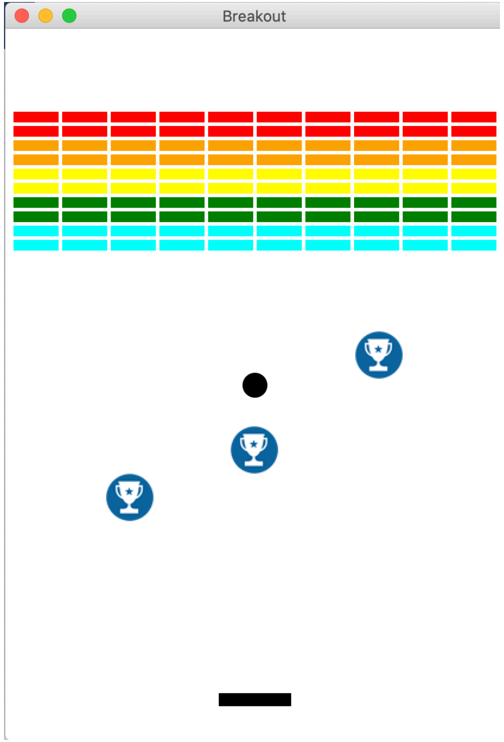
Brick Breaker Heart Collector	2021	Windows	
Total Dark	2020	Windows	Garage Games
Arcanoid Breakout	2020	Macintosh, Nintendo Switch, Windows Apps, Xbox One	Pix Arts
Twin Breaker: A Sacred Symbols Adventure	2020	Nintendo Switch, PlayStation 4, PS Vita, Xbox One	Eastasiasoft Limited
Dungeonoid	2020	Nintendo Switch	Super Powerup Games S.L.
Immortal Wanna	2019	Windows	Duck Inc.
Hit the Light	2019	Android, iPad, iPhone	Happymagenta UAB
Hentai Block Breaker	2019	Windows	
Block Kuzushi II	2019	Windows	
Space Candy	2019	Windows	khukhrovr
Cute Blocks	2019	Windows	cBlck
!4RC4N01D! 2: Retro Edition	2018	Windows	armogames
Breakdown	2018	TRS-80	PSKI Software Development, Inc.
Drawkanoid	2018	Macintosh, Windows	Humble Bundle, Inc.
!4RC4N01D! 4: KOHBEEP edition	2018	Windows	armogames
Voxel Baller	2018	Linux, Windows	MKD Games
Break Bricks: Ball's Quest	2018	iPad, iPhone	
Energy Invasion	2018	Linux, Macintosh, Nintendo Switch, PlayStation 4, PS Vita, Windows	Sometimes You
Briks 2	2018	PlayStation 4	SMobile, Inc
Deconstructor	2018	Linux, Macintosh, Windows	For Kids
!4RC4N01D! 3: Cold Space	2018	Windows	armogames
Ballz Royale	2018	Windows	Orlando
DX-Ball 2: 20th Anniversary Edition	2018	Windows	Longbow Games
!4RC4N01D!	2018	Windows	armogames
Nextoid!	2018	ZX Spectrum Next	

https://www.mobygames.com/game-group/breakout-variants

Extra Features for Breakout

Collecting objects with paddle

```
def get_key_press(canvas):
    presses = canvas.get_new_key_presses()
    for press in presses:
        ….
    return ….
```

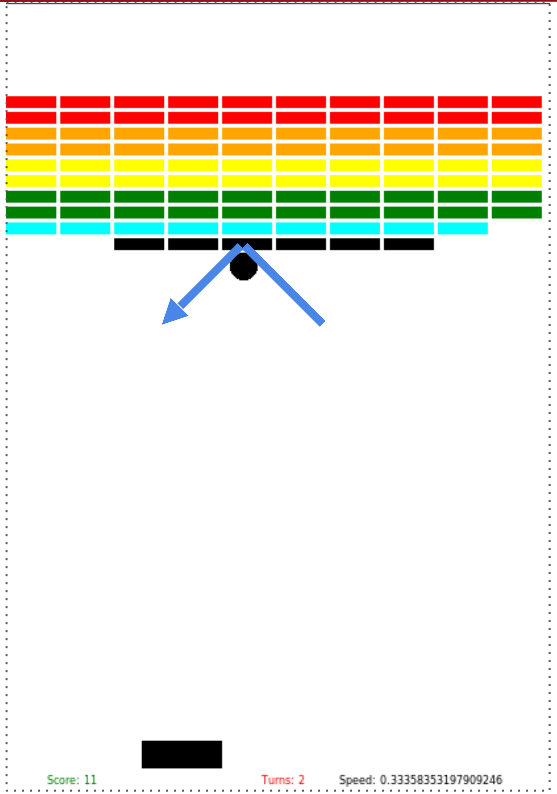


```
….
bullet = None

# animation loop
while <some condition>:
    …
    if bullet:# checking existence
        <Move bullet>
        if <some condition>:
            canvas.delete(bullet)
            bullet = None

…
```

Extra Features for Breakout

Special Bricks

How would you implement that?



Complete reference for class canvas: https://turkey21.csbridge.org/en/resources/api-docs/graphics.html

Extra Features for Breakout

- Increasing speed
- User selected game level - difficulty control
- Cheats
- Showing score
- Tracking time
- Lives
- … any other suggestions?