

# L6.1 Κίνηση

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

Στην διάλεξη 6.1 θα μελετήσουμε τρόπους να παράγουμε κινούμενα αντικείμενα στην επιφάνεια του καμβά.

**week 6**

# Σχεδιάζουμε παιχνίδια με python

Προγραμματίζω  
με την python



L6.1 Κίνηση

L6.2 Αναπτύσσουμε παιχνίδια

L6.3 Τελική εργασία

# Μάθημα L6.1

## Κίνηση

Προγραμματίζω  
με την python 

- V6.1.1 Βασικός μηχανισμός κίνησης
- V6.1.2 Φυσική κίνησης
- V6.1.3 Κίνηση χαρακτήρα
- V6.1.4 Κίνηση τοπίου
- V6.1.5 Παράδειγμα κίνησης

Προγραμματίζω  
με την python



# V6.1.1

## Βασικός μηχανισμός κίνησης αντικειμένων

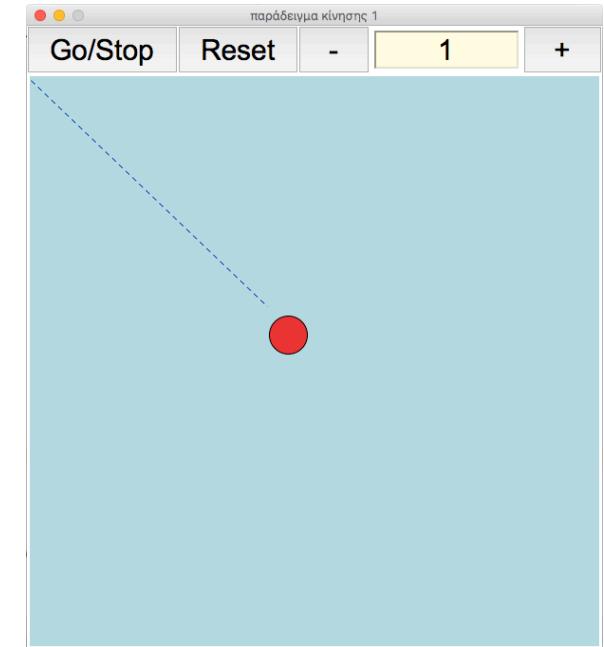
Νίκος Αβούρης, Πανεπιστήμιο Πατρών

# κίνηση αντικειμένου

Προγραμματίζω  
με την python 

Ο συνδυασμός της μεθόδου `move(item, dx, dy)` που κινεί το `item` κατά `dx`, `dy` με την μέθοδο `after(t, συνάρτηση)` που καλεί τη συνάρτηση που περιέχει τη `move` αναδρομικά μετά από `t msec` επιτρέπουν την κίνηση του αντικειμένου `item`

```
def move_thing(self, *args):
    if self.run:
        self.canvas.move("thing", *self.speed)
        self.x += self.speed[0] #νέα θέση στον άξονα x
        self.y += self.speed[1] #νέα θέση στον άξονα y
        if self.r+self.pad < self.x < self.width-self.r-self.pad and \
           self.r+self.pad < self.y < self.height-self.r-self.pad :
            self.canvas.after(10, self.move_thing) # αναδρομή
    else: return False
```



Προγραμματίζω  
με την python



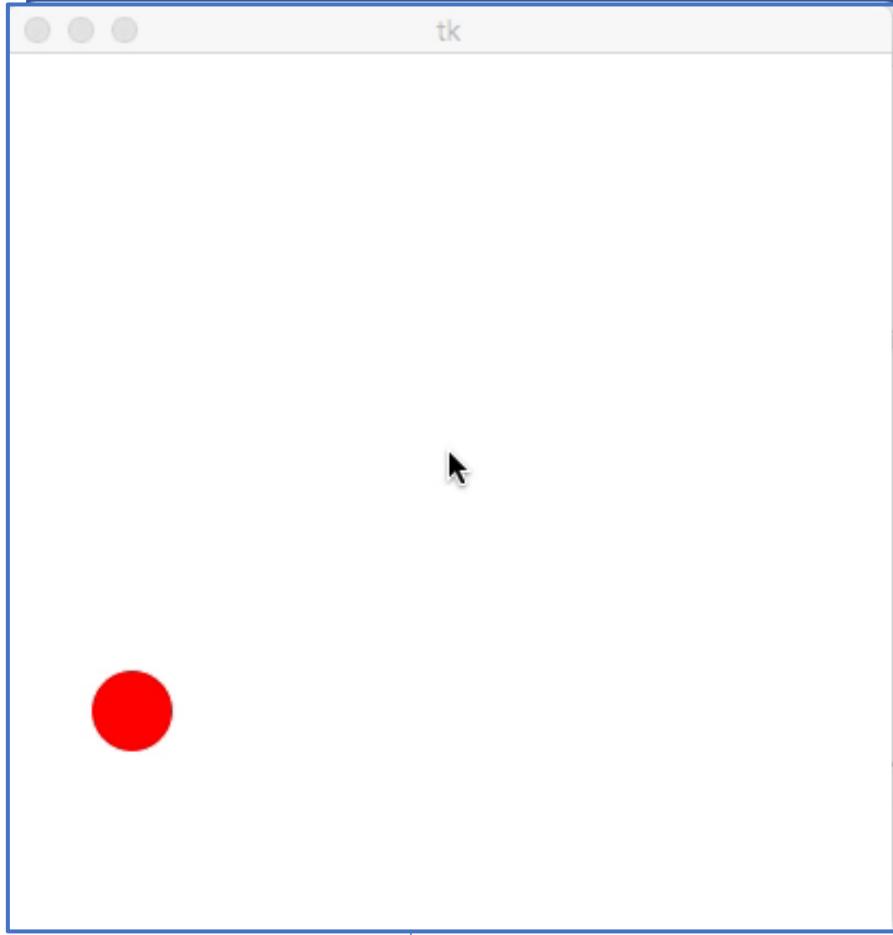
# V6.1.2

# Φυσική κίνησης

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

# φυσική κίνησης: ανάκλαση στις πλευρές

Προγραμματίζω  
με την python



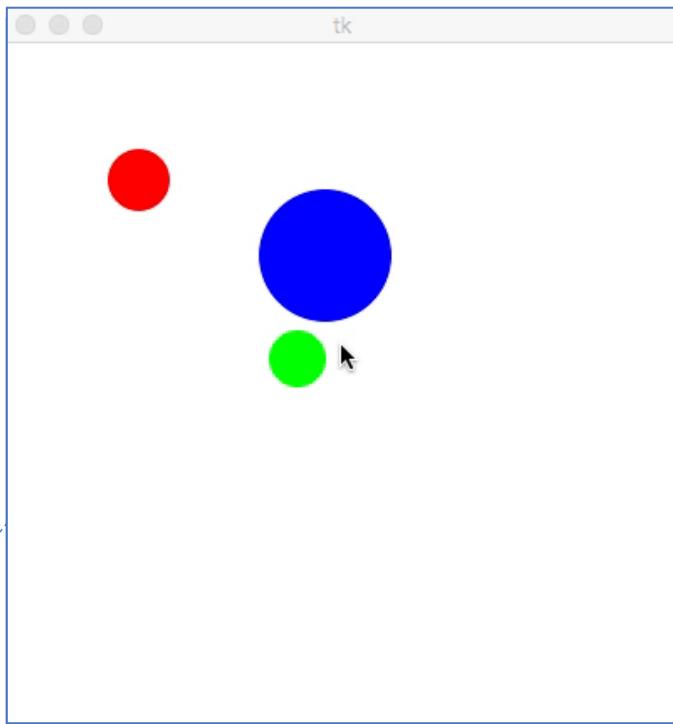
Όταν το αντικείμενο χτυπήσει κάποια πλευρά αλλάζει η φορά της μετατόπισης σύμφωνα με τη φυσική.

Αλλάζει η φορά κατά τον άξονα x για πάνω και κάτω πλευρά και κατά τον άξονα y για δεξιά και αριστερή πλευρά

```
x1, y1, x2, y2 = self.canvas.bbox(self.id)
if x2 > app.width: self.vx = -self.vx
if y2 > app.height: self.vy = -self.vy
if x1 < 0: self.vx = -self.vx
if y1 < 0: self.vy = -self.vy
self.canvas.move(self.id, self.vx, self.vy)
```

# φυσική κίνησης: σύγκρουση κινούμενων αντικειμένων

Προγραμματίζω  
με την python



Μια απλοποιημένη συμπεριφορά είναι να προσομοιώσουμε ανάκλαση με γωνία  $90^\circ$  (αντιστροφή της ταχύτητας ως προς και τις δύο διαστάσεις x,y)

```
def move(self):
    x1, y1, x2, y2 = self.canvas.bbox(self.id)
    if x1 < 0: self.vx = abs(self.vx) # αριστερή πλευρά
    if x2 > 400: self.vx = -abs(self.vx) # δεξιά πλευρά
    if y1 < 0: self.vy = abs(self.vy) # πάνω πλευρά
    if y2 > 400: self.vy = -abs(self.vy) # κάτω πλευρά

    if len(self.canvas.find_overlapping(x1,y1,x2,y2))>1:
        self.handle_collision()
    self.canvas.move(self.id, self.vx, self.vy)

def handle_collision(self):
    self.vx = -self.vx
    self.vy = -self.vy
```

Προγραμματίζω  
με την python



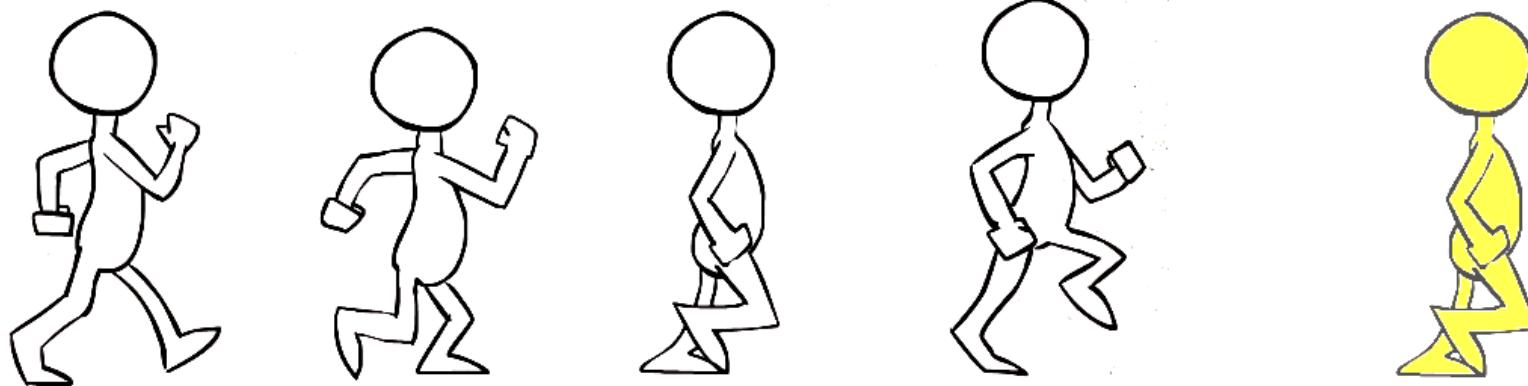
V6.1.3

Κίνηση χαρακτήρα

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

# κίνηση με διαδοχικές εικόνες

Προγραμματίζω  
με την python



ALAN BECKER - Animating Walk Cycles  
<https://www.youtube.com/watch?v=2y6aVz0Acx0&feature=youtu.be>

Στην περίπτωση αυτή δημιουργούμε μια ακολουθία από εικόνες σε διάφορες φάσεις μιας κίνησης και τις εμφανίζουμε διαδοχικά, είτε στην ίδια θέση είτε σε διαδοχικές θέσεις.

# κίνηση με διαδοχικές εικόνες

Προγραμματίζω  
με την python 

```
class Animated_character():
    def __init__(self, canvas, dir, size = 0, name = ''):
        self.canvas = canvas
        self.dir = dir
        self.canvas_height = int(self.canvas['height'])
        self.canvas_width = int(self.canvas['width'])
        self.images = []
        images = [x for x in os.listdir(self.dir) if x.split('.')[1] == 'gif']
        for _i in range(len(images)):
            file_name = os.path.join(dir, name+str(_i+1)+'.gif')
            if os.path.isfile(file_name): self.images.append(tk.PhotoImage(file=file_name))
        self.max_height = max([x.height() for x in self.images]) # max height of character
        self.speed = 200
        start_x = self.canvas_width // 2 - max([x.width() for x in self.images]) // 2
        start_y = self.canvas_height - margin - self.max_height
        self.current_image = 0
        self.art = self.canvas.create_image(start_x, start_y, image=self.images[self.current_image], anchor='nw')
    def next_image(self):
        self.current_image = self.current_image + 1 if self.current_image+1 < len(self.images) else 0
        self.canvas.itemconfig(self.art, image=self.images[self.current_image])
    def move(self):
        self.next_image()
        self.canvas.after(self.speed, self.move)
```

Προγραμματίζω  
με την python



# V6.1.4

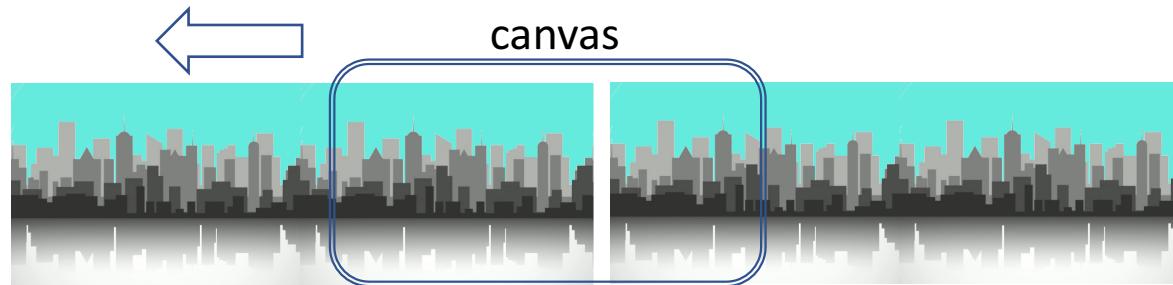
# κίνηση τοπίου

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

# κίνηση του τοπίου



[https://www.freepik.com/free-vector/black-and-white-city-skyline\\_764693.htm](https://www.freepik.com/free-vector/black-and-white-city-skyline_764693.htm)



δύο ή περισσότερες εικόνες  
εναλλάσσονται

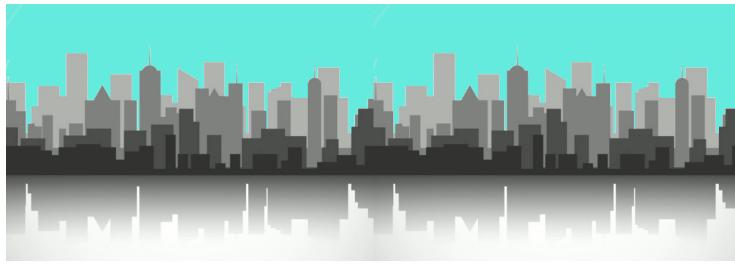
Προγραμματίζω  
με την python



Στην περίπτωση αυτή κινούμε  
το σκηνικό προσομοιώνοντας  
την κίνηση προς μια  
κατεύθυνση



# κίνηση του τοπίου



Προγραμματίζω  
με την python



```
class Background():
    def __init__(self, canvas, *args, **kwargs):
        self.canvas = canvas
        self.canvas_width = self.canvas['width']
        self.back_image = tk.PhotoImage(file= 'skyline.gif')
        self.back_width = self.back_image.width()
        self.back1 = self.canvas.create_image(0,0, image=self.back_image, anchor='nw')
        self.back2 = self.canvas.create_image(self.back_width,0, image=self.back_image,
                                             anchor='nw')
        self.run = 'go'
        self.speed = -5
        self.move_background()
```

Προγραμματίζω  
με την python



# V6.1.5

# παράδειγμα κίνησης

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

# συνδυασμός κίνησης χαρακτήρα και σκηνικού



Προγραμματίζω  
με την python 

Μπορούμε να προσθέσουμε συμπεριφορά, όπως αλλαγή κατεύθυνσης κίνησης, ο χαρακτήρας να σταματάει και να ξεκινάει, να επιταχύνει και να επιβραδύνει, να πηδάει στον αέρα (προσοχή στο φαινόμενο της βαρύτητας), ανάλογα με τις ανάγκες ενός παιχνιδιού.

# L6.2

# Αναπτύσσουμε παιχνίδια

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

Στην διάλεξη 6.2 θα δούμε τη διαδικασία ανάπτυξης απλών παιχνιδιών με χρήση του καμβά. Θα συζητήσουμε την οργάνωση του προγράμματος ενός παιχνιδιού και θα εξετάσουμε δύο πιο ειδικά θέματα, την εισαγωγή χαρακτήρων του πραγματικού κόσμου καθώς και την διαχείριση του χρόνου.

# Μάθημα L6.2

## Αναπτύσσουμε παιχνίδια

Προγραμματίζω  
με την python 

- V6.2.1 Caveman – το παιχνίδι
- V6.2.2 Caveman – η πίστα
- V6.2.3 Caveman – οι χαρακτήρες
- V6.2.4 Προσωποποίηση παιχνιδιών
- V6.2.5 Διαχείριση του χρόνου

Προγραμματίζω  
με την python



V6.2.1

Caveman: το παιχνίδι

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

# τα παιχνίδια σήμερα

Η σημασία των παιχνιδιών σήμερα

Μανιφέστο: ο 21<sup>ος</sup> αιώνας θα καθορίζεται από τα Παιχνίδια  
(Παιγνιώδης αιώνας Ludic century)

Προηγούμενοι αιώνες καθορίστηκαν από λογοτεχνία και τον κινηματογράφο, ο σχεδιαστής και θεωρητικός των παιχνιδιών *Eric Zimmerman* ισχυρίζεται ότι ο 21<sup>ος</sup> αιώνας διαμορφώνεται από τα παιχνίδια ως νέα μέσα έκφρασης

Προγραμματίζω  
με την python



## Manifesto: The 21st Century Will Be Defined By Games



Eric Zimmerman & Heather Chaplin  
9/09/13 2:17pm • Filed to: MANIFESTO ▾

42.1K 86 10



# παιχνίδια

Προγραμματίζω  
με την python 

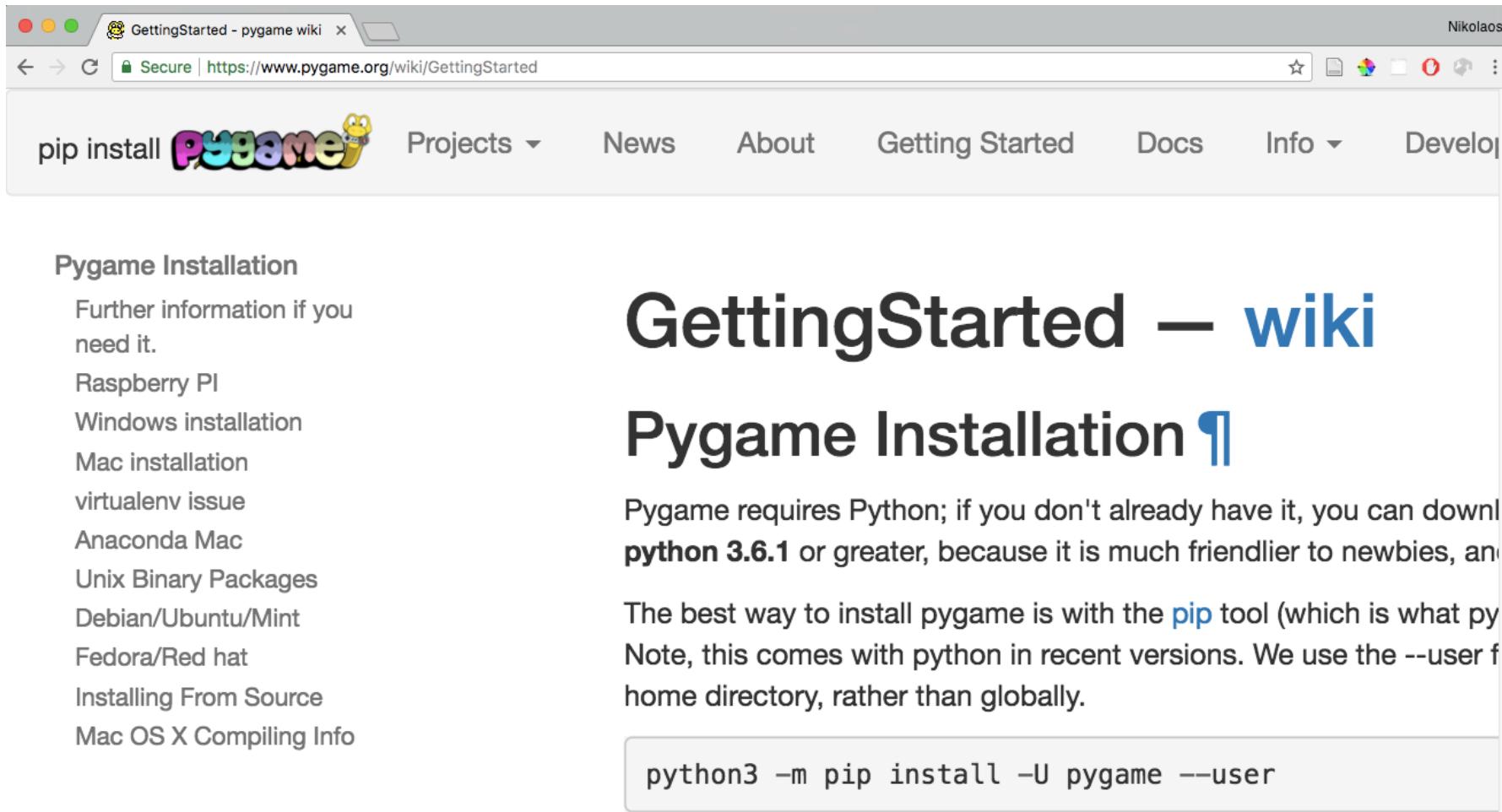


- Η σημασία των παιχνιδιών ως μέσων για εισαγωγή στον προγραμματισμό
- πώς σχεδιάζουμε ένα παιχνίδι με την tkinter/python
- Ένα παράδειγμα: ο caveman: η ιστορία – οι πόροι – η διαμόρφωσή τους – οι κανόνες – η υλοποίηση

# pygame

Προγραμματίζω  
με την python 

## Η βιβλιοθήκη της python για ανάπτυξη παιχνιδιών



The screenshot shows a web browser window with the title "GettingStarted - pygame wiki". The URL in the address bar is "https://www.pygame.org/wiki/GettingStarted". The page content includes a sidebar with links for Pygame Installation (Further information if you need it, Raspberry Pi, Windows installation, Mac installation, virtualenv issue, Anaconda Mac, Unix Binary Packages, Debian/Ubuntu/Mint, Fedora/Red hat, Installing From Source, Mac OS X Compiling Info) and a main article titled "GettingStarted – wiki". The main article discusses Pygame requirements (Python 3.6.1 or greater), the best way to install (using pip), and a command-line example: "python3 -m pip install -U pygame --user".

Pygame Installation

- Further information if you need it.
- Raspberry Pi
- Windows installation
- Mac installation
- virtualenv issue
- Anaconda Mac
- Unix Binary Packages
- Debian/Ubuntu/Mint
- Fedora/Red hat
- Installing From Source
- Mac OS X Compiling Info

## GettingStarted – wiki

### Pygame Installation ¶

Pygame requires Python; if you don't already have it, you can download [python 3.6.1](#) or greater, because it is much friendlier to newbies, and

The best way to install pygame is with the [pip](#) tool (which is what pygame uses). Note, this comes with python in recent versions. We use the --user flag to install it in your home directory, rather than globally.

```
python3 -m pip install -U pygame --user
```

# φάσεις στην ανάπτυξη ενός παιχνιδιού

Προγραμματίζω  
με την python 

- Δημιουργία **ιστορίας** (χαρακτήρες, συμπεριφορές, στόχοι)
- Οι **πόροι** (assets) γραφικά αντικείμενα που υλοποιούν τη συμπεριφορά που έχουμε αποφασίσει. Μπορεί να είναι 3διάστατοι ή 2διάστατοι. Στα παιχνίδια με tkinter είναι 2διάστατοι.
  - Caveman
  - Dragons
  - Mammoths
- Οι **πίστες** (backdrops, background)
  - Cave
- Επεξεργασία των πόρων
- Ορισμός **κανόνων** (game mechanics), awards, score, levels

Προγραμματίζω  
με την python



V6.2.2

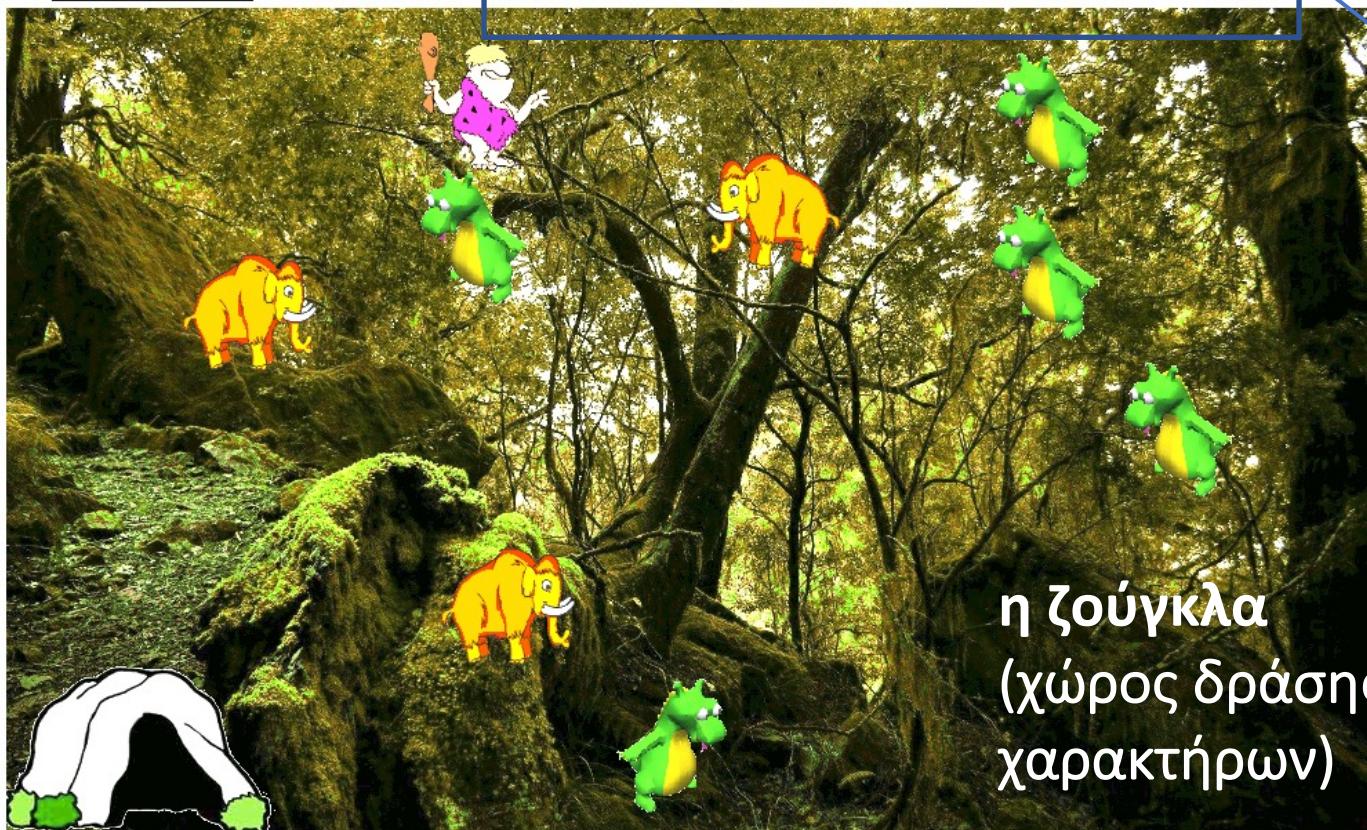
Caveman: η πίστα

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

# σχεδιασμός της πίστας

ρολόι

σκορ



αποθήκη

Προγραμματίζω  
με την python



μενού:

- νέο παιχνίδι
- αποθήκευση σκορ
- [ info]

ζωές

Προγραμματίζω  
με την python



V6.2.3

Caveman: οι χαρακτήρες

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

# Οι κλάσεις των χαρακτήρων

## Artifact

```
__init__  
create_artifact  
move  
handlecollision  
set_direction  
stop  
hit_object
```

## Mamooth

```
reset  
__init__  
handlecollision
```

## Dragon

```
__init__  
handlecollision
```

## Caveman

```
__init__  
hit_object  
fat_caveman  
dead_caveman  
back_to_normal  
moveright  
moveleft  
moveup  
movedown
```

Προγραμματίζω  
με την python



Προγραμματίζω  
με την python



V6.2.4

# Προσωποποίηση παιχνιδιών

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

Προγραμματίζω  
με την python



# V6.2.5

# Διαχείριση του χρόνου

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

# χρόνος: η βιβλιοθήκη time

Προγραμματίζω  
με την python 

```
import time
print(time.time())
1517699082.389348 # πραγματικός αριθμός που
μετράει sec που πέρασαν από την αρχή του
χρόνου
>>> time.localtime()
time.struct_time(tm_year=2018, tm_mon=2,
tm_mday=4, tm_hour=1, tm_min=5, tm_sec=21,
tm_wday=6, tm_yday=35, tm_isdst=0)
time.gmtime() # GMT
```

Η αρχή του χρόνου ορίζεται η 00:00:00 UTC Πέμπτη, 1 Ιανουαρίου 1970

# χρόνος: η βιβλιοθήκη time

Προγραμματίζω  
με την python 

ανάκτηση χρήσιμης πληροφορίας για παρουσίαση με χρήση της μεθόδου `time.strftime(spec, t)` όπου `spec` μία συμβολοσειρά μορφοποίησης του χρόνου: `%A` μέρα εβδομάδας, `%B` όνομα μήνα, `%d` μέρα μήνα, `%m` μήνας, `%H %M %S` (ώρα λεπτό δευτερόλεπτο)

```
t=time.localtime() αν δεν δοθεί τιμή  
>>> time.strftime("%H:%M:%S")  
'01:35:15'
```

# χρόνος: η βιβλιοθήκη time

**time.process\_time()**

μέτρηση του χρόνου του επεξεργαστή διαδοχικές κλήσεις επιτρέπουν να μετρήσουμε χρόνο διεργασίας

Προγραμματίζω  
με την python 

```
import time

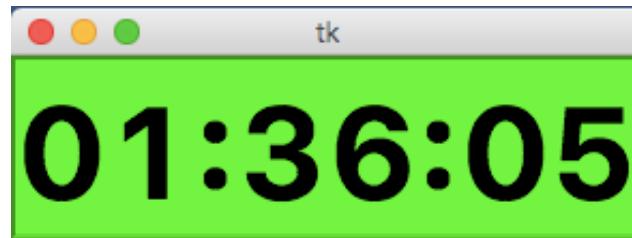
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n-1)

t0 = time.process_time()
for i in range(10000):
    factorial(200)
print(time.process_time() - t0)
```

# ρολόι:

```
import tkinter as tk  
import time
```

```
class App():  
    def __init__(self, root):  
        self.root = root  
        self.fnt = ('Century', 60, 'bold')  
        self.f1 = tk.Frame(self.root)  
        self.f1.pack(side='top', fill='both', expand=1)  
        self.clock = tk.Label(self.f1, font= self.fnt, bg='green', relief='sunken')  
        self.clock.pack(fill='both', expand=1)  
        self.tick()  
    def tick(self):  
        s = time.strftime('%H:%M:%S' )  
        if s != self.clock["text"]:  
            self.clock["text"] = s  
            self.clock.after(200, self.tick)  
root = tk.Tk()  
App(root)  
root.mainloop()
```



Προγραμματίζω  
με την python



# χρονόμετρο

Προγραμματίζω  
με την python 

```
def start_timer(self):
    """ Ξεκίνησε τη μέτρηση του χρόνου """
    self._elapsedtime = 0.0
    self._start = time.time()
    self._update_timer()

def _update_timer(self):
    """ Ανανέωσε το label με τον χρόνο που έχει περάσει. """
    self._elapsedtime = time.time() - self._start
    self._set_time(self._elapsedtime)
    self._timer = self.root.after(200, self._update_timer)

def _set_time(self, elapsed):
    """ Όρισε με μορφή Minutes:Seconds το StringVar timestr """
    minutes = int(elapsed / 60)
    seconds = int(elapsed - minutes * 60.0)
    self.timestr.set('%02d:%02d' % (minutes, seconds))
    return '%02d:%02d' % (minutes, seconds)
```

# προδιαγραφές time.strftime(spec)

<b>%a</b>	Ημέρα εβδομάδας πχ Fri
<b>%A</b>	Ημέρα εβδομάδας πλήρης πχ Friday
<b>%b</b>	Μήνας, σύντομος, πχ Feb
<b>%B</b>	Μήνας πλήρης πχ Frbruary
<b>%c</b>	πλήρης αναπαράσταση ημερομηνίας, ώρας Fri Apr 20 22:49:38 2018
<b>%d</b>	Ημέρα μήνα [01,31].
<b>%H</b>	Ώρα [00,23].
<b>%I</b>	Ώρα [01,12].
<b>%j</b>	Ημέρα του χρόνου [001,366].
<b>%m</b>	Μήνας [01,12].
<b>%M</b>	Λεπτό [00,59].
<b>%p</b>	AM ή PM.

<b>%S</b>	δευτερόλεπτο [00,61].
<b>%U</b>	εβδομάδα του χρόνου 1 <sup>η</sup> εβδομάδα αρχίζει την πρώτη Κυριακή του χρόνου [00,53].
<b>%w</b>	Ημέρα εβδομάδας ως αριθμός [0(Sunday),6].
<b>%W</b>	εβδομάδα του χρόνου 1 <sup>η</sup> εβδομάδα αρχίζει την πρώτη Δευτέρα του χρόνου [00,53].
<b>%x</b>	Ημερομηνία πχ 04/20/18.
<b>%y</b>	Έτος τελευταία ψηφία [00,99].
<b>%Y</b>	Έτος πλήρης αναπαράσταση πχ 2019
<b>%z</b>	Ζώνη χρόνου ως διαφορά από UTC/GMT με μορφή +HHMM or -HHMM, πχ +0300
<b>%Z</b>	Ζώνη χρόνου πχ EEST

Προγραμματίζω  
με την python



# Άσκηση

(για συζήτηση στο forum)

Προγραμματίζω  
με την python



Χρειαζόμαστε βοήθεια. η εφαρμογή Caveman έχει ένα σοβαρό πρόβλημα. Όταν στη μέση του παιχνιδιού ο χρήστης πατήσει το πλήκτρο [i] το παιχνίδι σταματάει, το ίδιο και ο χρονομέτρης, όταν όμως επιστρέψει στο παιχνίδι, υπολογίζει και τα δευτερόλεπτα της παύσης. Κάντε τις κατάλληλες διορθώσεις στις μεθόδους διαχείρισης του χρόνου.

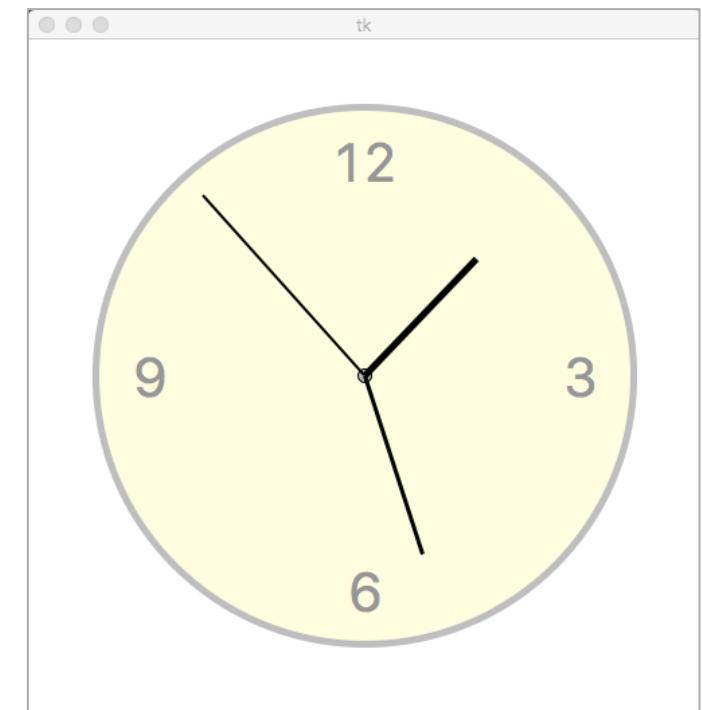
# Άσκηση

(για συζήτηση)

Να κατασκευάσετε ένα αναλογικό ρολόι,  
να χρησιμοποιήσετε τις βιβλιοθήκες time,  
math, tkinter.

Παραλλαγή: να προσθέσετε ημερολόγιο  
και άλλα στοιχεία της ημέρας, (πχ σε ποια  
βδομάδα του χρόνου ανήκει, ποια μέρα  
της βδομάδας κλπ).

Προγραμματίζω  
με την python



# L6.3

## Τελική εργασία: το παιχνίδι 31

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

Στην διάλεξη 6.3 θα αναπτύξουμε ένα παιχνίδι, το 31. Θα ζητηθεί στο τέλος της διάλεξης να επεκτείνετε το παιχνίδι ως μέρος της τελικής εργασίας του μαθήματος.

# Μάθημα L6.3

## Τελική εργασία

Προγραμματίζω  
με την python 

- V6.3.1 Το παιχνίδι 31: Αρχική σχεδίαση του καμβά
- V6.3.2 Το παιχνίδι 31: Η κλάση CardImages
- V6.3.3 Το παιχνίδι 31: Η κλάση GUICard
- V6.3.4 Το παιχνίδι 31: Οι κλάσεις των παικτών
- V6.3.5 Τελική εργασία: Ολοκληρώνουμε το παιχνίδι 31
- V6.3.6 Κατευθύνσεις: από δω και πέρα τι;

Προγραμματίζω  
με την python



V6.3.1

Το παιχνίδι 31.

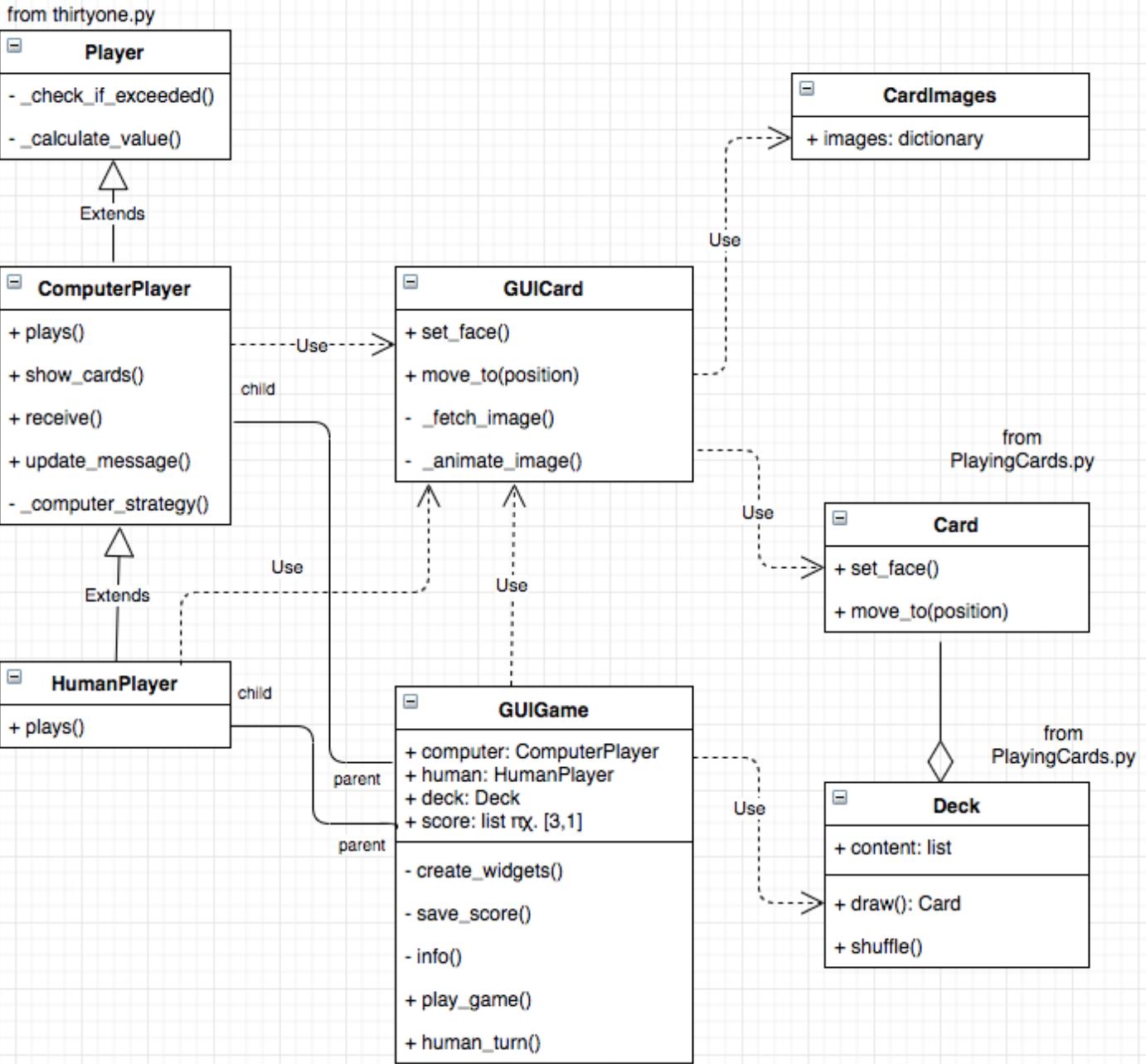
Αρχική σχεδίαση του καμβά

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

# Προγραμματίζω με την python



## thirtyone\_gui διάγραμμα κλάσεων



Προγραμματίζω  
με την python



V6.3.2

# Το παιχνίδι 31: Η κλάση CardImages

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

Προγραμματίζω  
με την python



V6.3.3

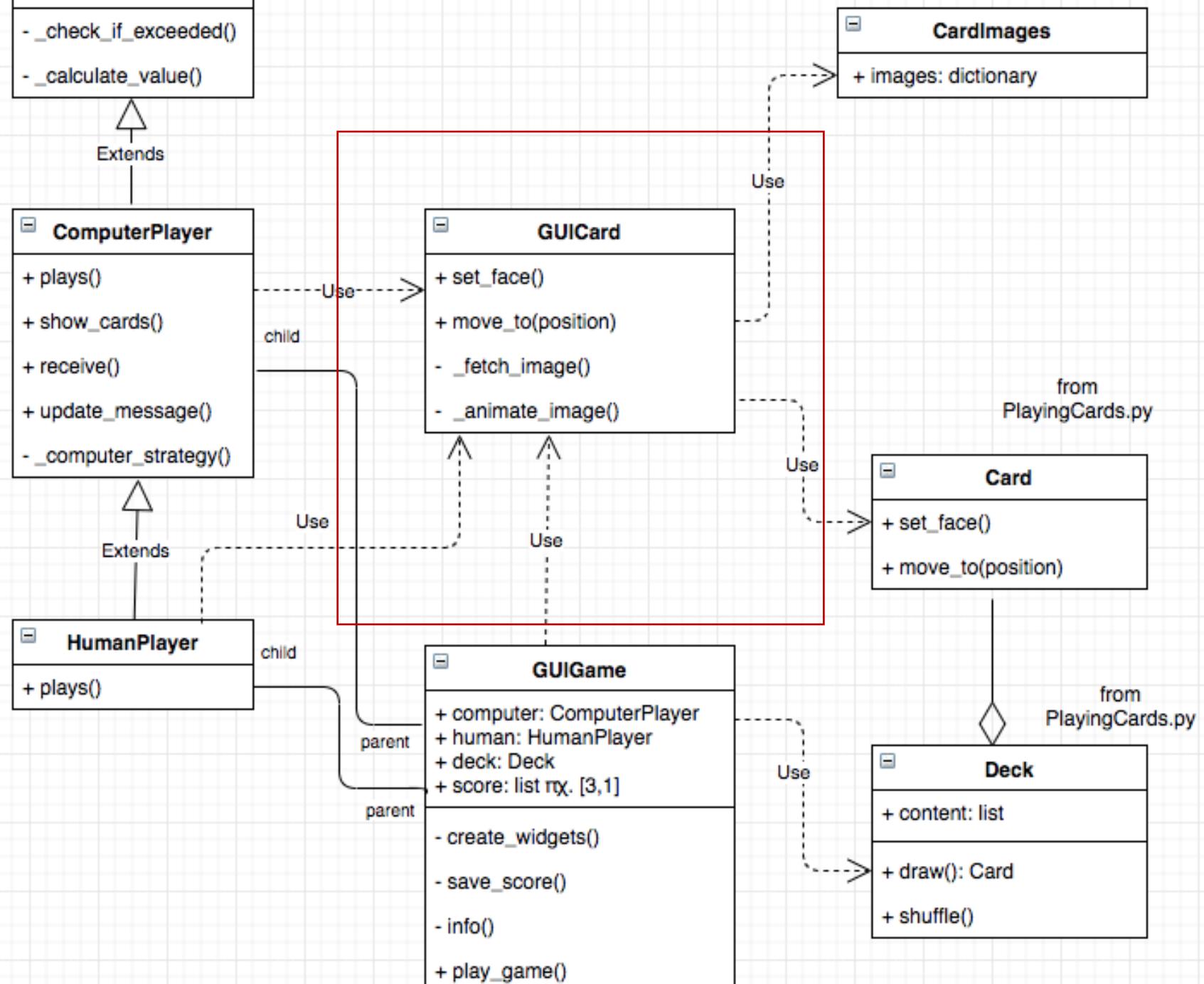
Το παιχνίδι 31:  
η κλάση GUICard

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

# Προγραμματίζω με την python



## GUICard



Προγραμματίζω  
με την python



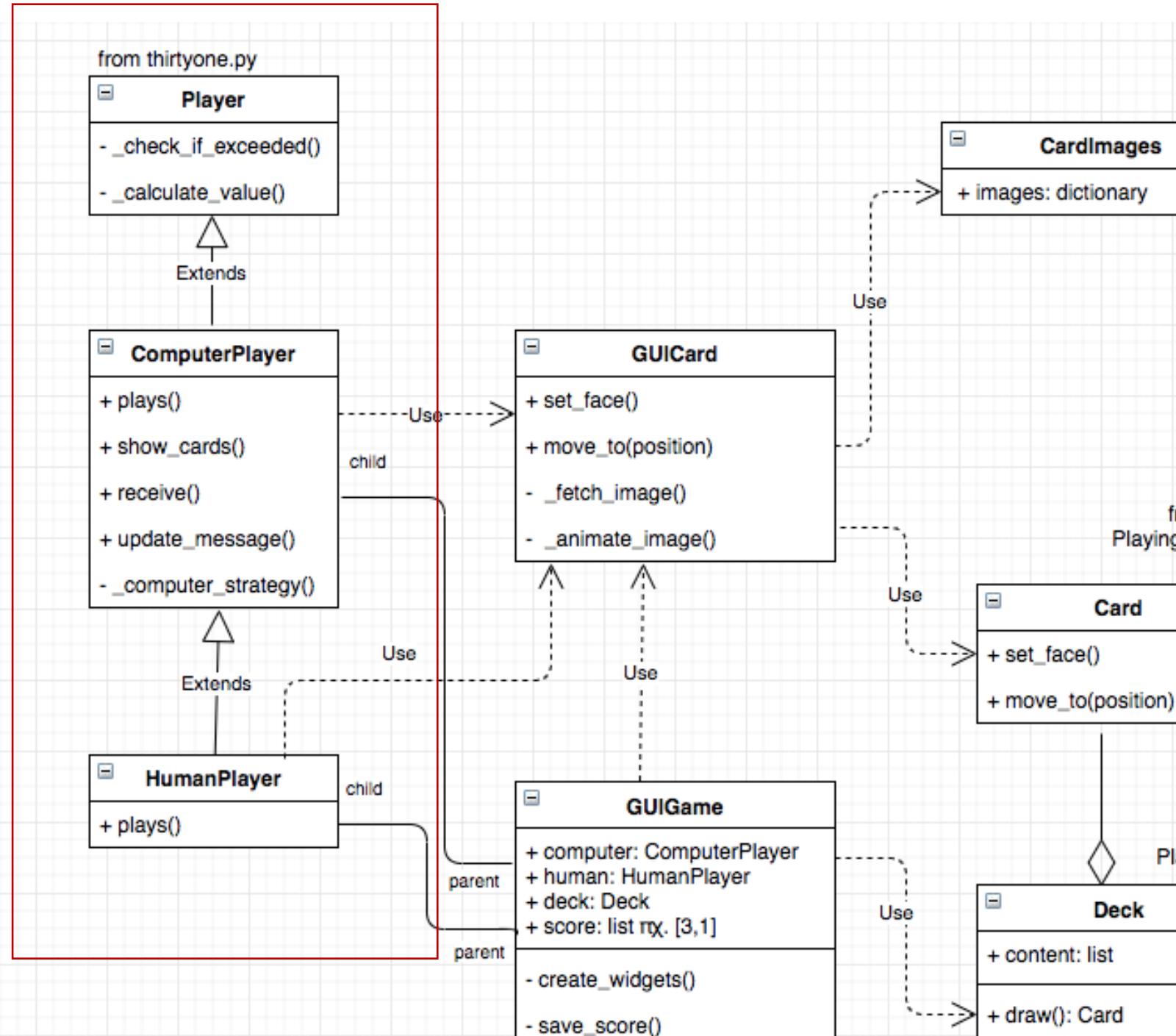
V6.3.4

# Το παιχνίδι 31: Οι κλάσεις των παικτών

Νίκος Αβούρης, Πανεπιστήμιο Πατρών



# Οι κλάσεις των παικτών ComputerPlayer HumanPlayer



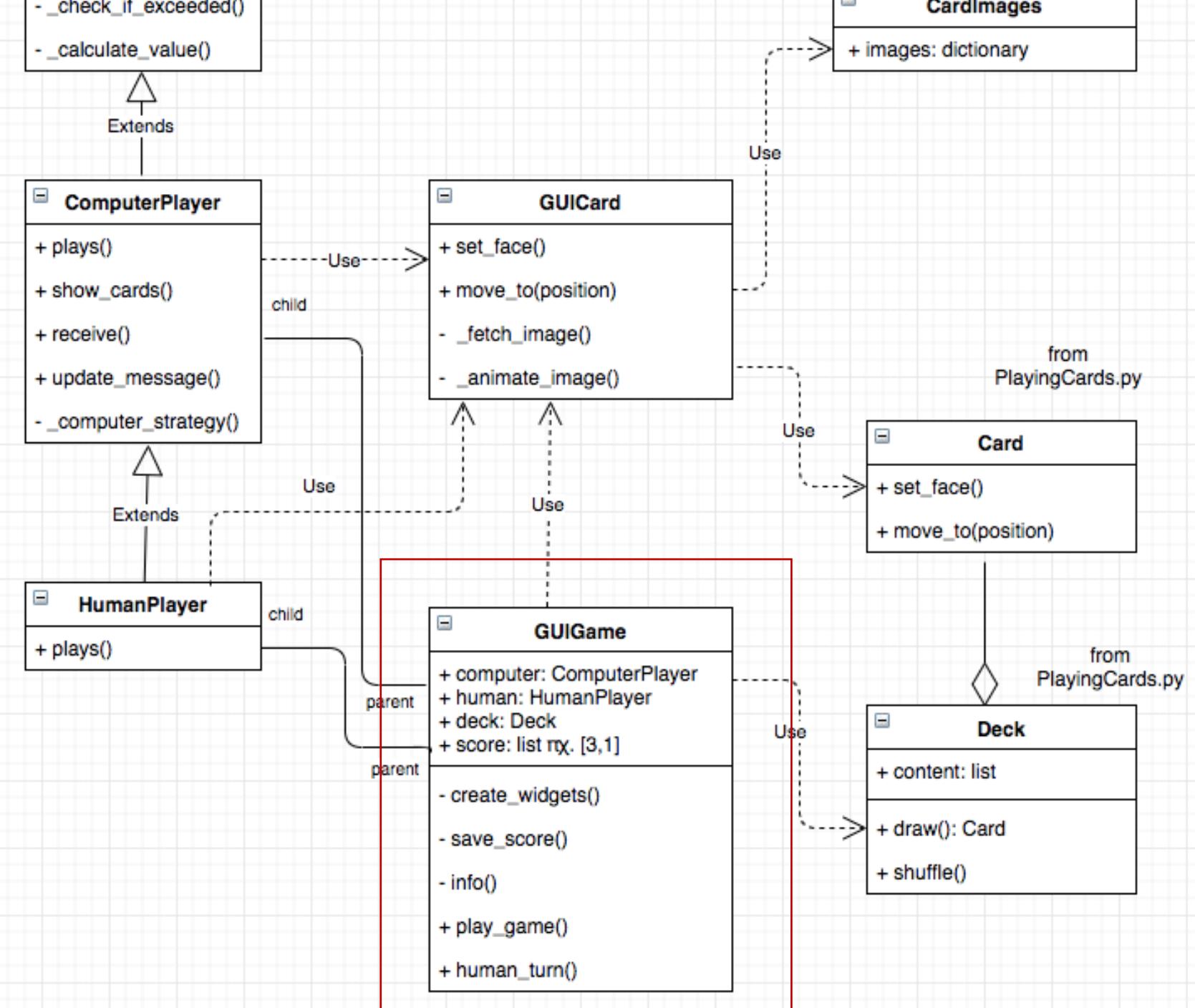
Προγραμματίζω  
με την python



# V6.3.5

## Το παιχνίδι 31: Η κλάση GUIGame

Νίκος Αβούρης, Πανεπιστήμιο Πατρών



# Προγραμματίζω με την python



# GUIGame

Προγραμματίζω  
με την python



# V6.3.6

## Τελική εργασία:

# Ολοκληρώνουμε το παιχνίδι 31

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

**Στην τελική εργασία ζητείται να ολοκληρώσετε τη σχεδίαση του παιχνιδιού 31. Συγκεκριμένα στον κώδικα που δίνεται ζητείται να προστεθούν τα εξής:**

- (α) Να ολοκληρωθεί η πάνω μπάρα (Frame self.f1) έτσι ώστε να υπάρχει το πλήκτρο 'Νέο παιχνίδι', το πλήκτρο 'Άρκετά!' με το οποίο ο χρήστης διακόπτει, το πλήκτρο αποθήκευσης σκορ και το μήνυμα ανάκτησης υψηλότερων σκορ από τις παρτίδες που έχουν παιχτεί.
- (β) Να προστεθεί στην περιοχή κάθε παίκτη ένα μήνυμα που εμφανίζει το τελικό σκορ του, ή το μήνυμα 'Δυστυχώς κάηκες' αν περάσει το 31, προσοχή το μήνυμα στην περιοχή του υπολογιστή θα πρέπει να εμφανίζεται στο τέλος
- γ) Να προστεθεί η δυνατότητα να ζητείται το όνομα του παίκτη και αυτό να εμφανίζεται στην πάνω μπάρα στο σκορ. Αν ο χρήστης δεν δώσει όνομα, να εμφανίζεται απλά το όνομα 'Παίκτης'
- (δ) Να προστεθεί η δυνατότητα αποθήκευσης του σκορ αν παιχτούν συνολικά τουλάχιστο 3 παρτίδες και εφόσον ο χρήστης έχει δώσει το όνομά του. Να χρησιμοποιήσετε τη βιβλιοθήκη pickle για την αποθήκευση του σκορ.
- (ε) Να εμφανίζεται το σκορ για τους 5 παίκτες που έχουν υψηλότερη επίδοση (έχουν το υψηλότερο ποσοστό κερδισμένων παιχνιδιών, αν δύο παίκτες έχουν το ίδιο ποσοστό, να εμφανίζει ψηλότερα αυτόν που έχει παίξει περισσότερα παιχνίδια, πχ ο παίκτης Α έχει σκορ 3-3, άρα έχει ποσοστό επιτυχίας 50% και ο παίκτης Β έχει σκορ 5-5 επίσης 50%, η σειρά εμφάνισης θα πρέπει να είναι τέτοια ώστε ο Β να εμφανίζεται πριν τον Α )

**Προγραμματίζω  
με την python**



Προγραμματίζω  
με την python



V6.3.7

Κατευθύνσεις: από δω και πέρα τι;

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

# προγραμματίζω με την python

Προγραμματίζω  
με την python 

1. Ανάλυση δεδομένων (pandas, NumPy, SciPy, matplotlib network-x)
2. Διαδικτυακές εφαρμογές (frameworks όπως το django, flask)
3. Εφαρμογές σε περιβάλλον Android, και κινητά, βιβλιοθήκη kivy

Προγραμματίζω  
με την python



# Ευχαριστούμε

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

**[hci.ece.upatras.gr/avouris](http://hci.ece.upatras.gr/avouris)**

# πηγές

Προγραμματίζω  
με την python 

<https://docs.python.org/3/library/tk.html>

<http://effbot.org/tkinterbook/>

<http://www.tkdocs.com/index.html>

[https://www.tutorialspoint.com/python3/python\\_gui\\_programming.htm](https://www.tutorialspoint.com/python3/python_gui_programming.htm)

<http://infohost.nmt.edu/tcc/help/pubs/tkinter/web/index.html>