

week 5

Σύνθετα widget

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

Την εβδομάδα αυτή μελετάμε πιο σύνθετα
γραφικά αντικείμενα της tkinter

L5.1


Listbox και η εφαρμογή MyContacts

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

Στην διάλεξη 5.1 θα εξετάσουμε το σύνθετο widget Listbox και στη συνέχεια θα προχωρήσουμε σε ανάπτυξη της εφαρμογής contacts που είδαμε σε προηγούμενη ενότητα με γραφική πλέον διεπαφή.

week 5

Σύνθετα widget

Προγραμματίζω
με την python 


L5.1 Listbox και η εφαρμογή Mycontacts

L5.2 Text

L5.3 Canvas

Μάθημα L5.1

Listbox και η εφαρμογή MyContacts

Προγραμματίζω
με την python 


- V5.1.1 Listbox και Scrollbar
- V5.1.2 MyContacts: Παρουσίαση επαφών σε Listbox
- V5.1.3 MyContacts: Σύνδεση με την κλάση Contact και τη βάση δεδομένων
- V5.1.4 MyContacts: Αναζήτηση στις επαφές
- V5.1.5 MyContacts: Ανάπτυξη των μενού χειρισμού επαφών

V5.1.1

Listbox και Scrollbar

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

Listbox

Προγραμματίζω
με την python 


Σύνθετο widget.

Δημιουργεί γραφικό αντικείμενο που περιέχει πολλές γραμμές κειμένου (μία λίστα), και μας επιτρέπει να τις επιλέξουμε και να τις τροποποιήσουμε.

Συνεργάζεται με το widget **Scrollbar**

Μια ακόμη πιο σύνθετη έκδοσή του είναι το αντικείμενο **Text** που θα δούμε αργότερα.

`l = tk.Listbox(παραμ.)`

Προγραμματίζω
με την python 

Εισαγωγή στοιχείου:

`l.insert('end', κείμενο)`

διαγραφή όλων των στοιχείων:

`l.delete(0, 'end')`


εύρεση δείκτη επιλεγμένου ή επιλεγμένων στοιχείων:

`idx = l.curselection() # επιστρέφει πλειάδα`

επιλογή του στοιχείου δείκτη idx από το listbox:

`l.get(idx)`

`s = tk.Scrollbar(παραμ.)`

Προγραμματίζω
με την python 

Χρησιμοποιείται σε συνδυασμό με την Listbox αλλά και με την Text, Canvas, κλπ

Τα δύο αντικείμενα που συνδυάζονται, δημιουργούνται αρχικά και στη συνέχεια με την `config()` συνδέονται

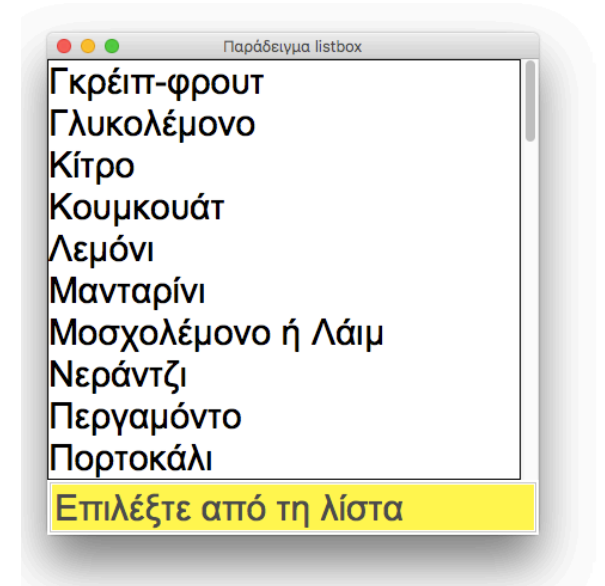
```
s.config(command=l.yview)
```

```
l.config(yscrollcommand = s.set)
```

παρόμοια μπορεί να γίνει οριζόντια κύλιση με τη μέθοδο `xview`

παράδειγμα Listbox

```
self.root.title("Παράδειγμα listbox")
self.f1 = tk.Frame(self.root)
self.f1.pack(side='top', expand=True, fill='both')
# δημιουργία listbox και συνδεδεμένης μπάρας κύλλησης
self.sbar = tk.Scrollbar(self.f1)
self.list = tk.Listbox(self.f1, width=30, height=10, font=self.fnt)
self.list.config(yscrollcommand=self.sbar.set) # σύνδεσε λίστα με sbar
self.sbar.config(command=self.list.yview) # σύνδεσε μπάρα κύλλησης με λίστα list
self.sbar.pack(side='right', expand=True, fill='y')
self.list.pack(expand=True, fill='both')
# δημιουργία entry box
self.entry = tk.Entry(self.root, width=30, bg='yellow', font = self.fnt)
self.entry['fg']='grey30'
self.entry.pack(side='bottom', expand=True, fill='both')
# επιλογή
self.list.bind('<ButtonRelease-1>', self.get_list)
```




V5.1.2

MyContacts :

Παρουσίαση επαφών σε Listbox

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

επαφές στο Listbox

Προγραμματίζω
με την python 

```
def widgets(self):
    self.f1 = tk.LabelFrame(self.root, text='επαφές')
    self.f1.pack(expand=True, fill='both', padx=2, pady=2)
    self.sbar = tk.Scrollbar(self.f1)
    self.sbar.pack(side='right', fill='y')
    self.list = tk.Listbox(self.f1, relief = 'sunken', font=self.fnt,
                           bg='lightyellow', height=30, width=30)
    self.list.pack(expand=True, fill='both')
    self.list.config(yscrollcommand=self.sbar.set)
    self.sbar.config(command=self.list.yview)
    # δημιουργία αναδυόμενου μενού
    self.context = tk.Menu(self.root, font=self.fnt)
    self.root.config(menu=self.context)
    self.context.add_command(label='Τροποποίηση ...', command=self.modify)
    self.context.add_command(label='Διαγραφή ...', command=self.delete)
    self.root.bind('<2>', self.post_menu)
```

V5.1.3

MyContacts: σύνδεση με την κλάση Contact και τη βάση δεδομένων


Νίκος Αβούρης, Πανεπιστήμιο Πατρών

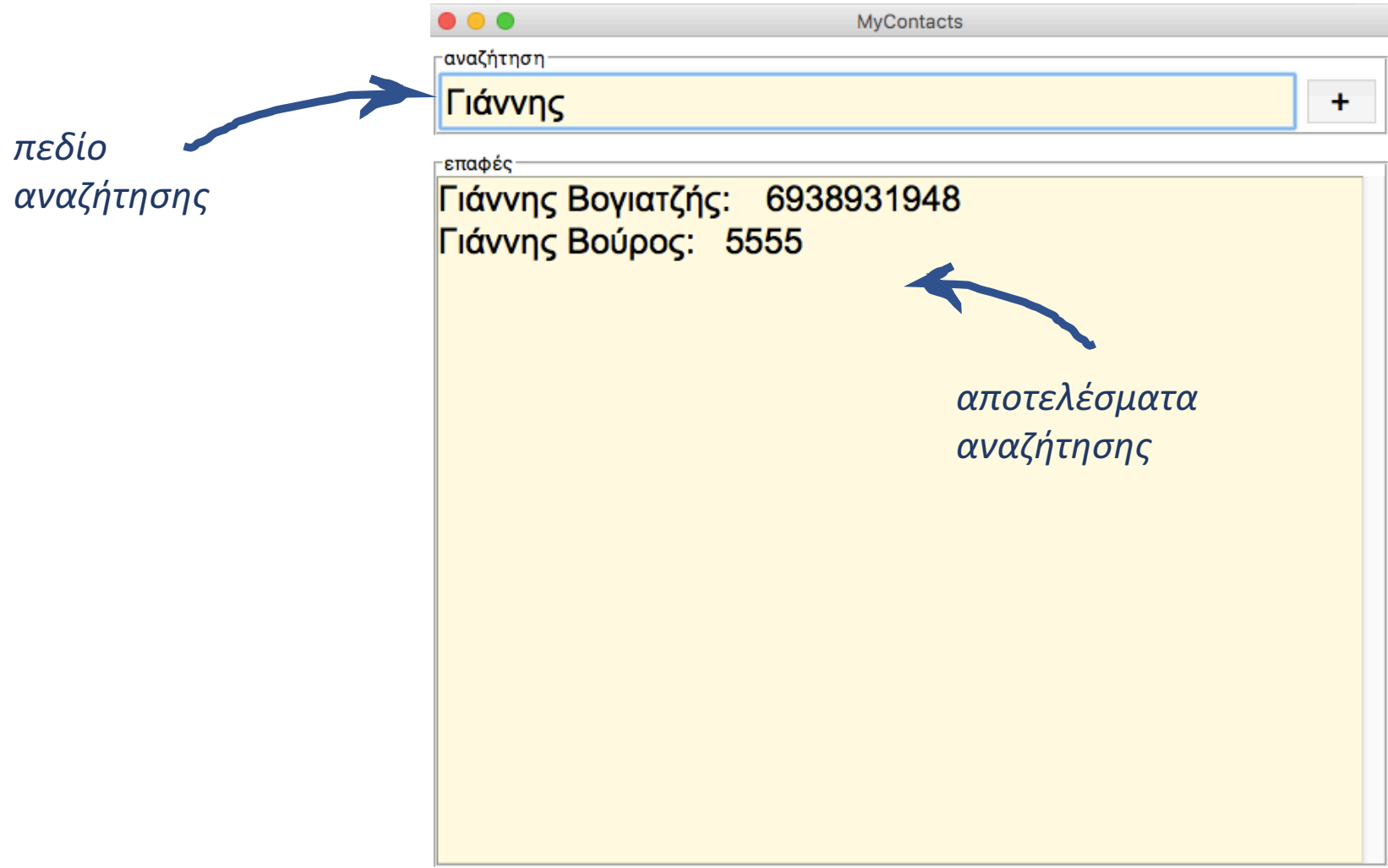
V5.1.4

MyContacts: Αναζήτηση στις επαφές

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

αναζήτηση στις επαφές

Προγραμματίζω
με την python 




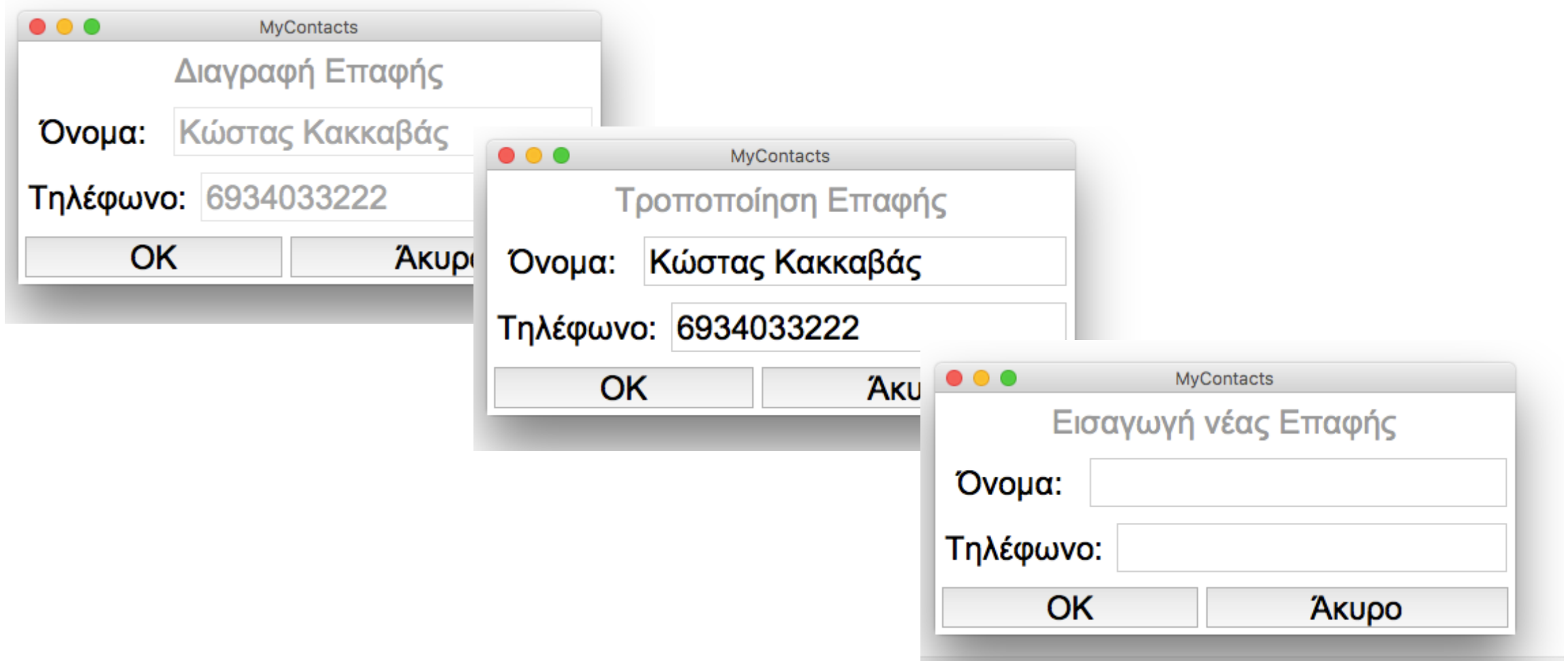
V5.1.5

MyContacts: Τροποποίηση και διαγραφή επαφών

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

διαγραφή /τροποποίηση/εισαγωγή επαφής

Προγραμματίζω
με την python 



The image displays three overlapping graphical user interface (GUI) windows for a contact management application titled "MyContacts". Each window has a standard macOS-style title bar with red, yellow, and green window control buttons.

- Top-left window: Διαγραφή Επαφής (Delete Contact)**
Contains two text input fields: "Όνομα:" (Name) with the value "Κώστας Κακκαβάς" and "Τηλέφωνο:" (Phone) with the value "6934033222". At the bottom are two buttons: "OK" and "Άκυρο" (Cancel).
- Middle window: Τροποποίηση Επαφής (Edit Contact)**
Contains two text input fields: "Όνομα:" (Name) with the value "Κώστας Κακκαβάς" and "Τηλέφωνο:" (Phone) with the value "6934033222". At the bottom are two buttons: "OK" and "Άκυρο" (Cancel).
- Bottom-right window: Εισαγωγή νέας Επαφής (Add New Contact)**
Contains two empty text input fields: "Όνομα:" (Name) and "Τηλέφωνο:" (Phone). At the bottom are two buttons: "OK" and "Άκυρο" (Cancel).


V5.1.6

MyContacts:

Εισαγωγή νέων επαφών

Νίκος Αβούρης, Πανεπιστήμιο Πατρών


Άσκηση

Προγραμματίζω
με την python 

Να συγκρίνετε την εφαρμογή contacts που αναπτύξαμε στο πλαίσιο της ενότητας μόνιμη αποθήκευση με την εφαρμογή mycontacts που αναπτύξαμε στο πλαίσιο της ενότητας γραφικών διεπαφών. Τι διαφορές παρατηρείτε στον κώδικα (αριθμός γραμμών, συνθετότητα), τι διαφορές παρατηρείτε ως προς την εμπειρία του χρήστη και τις λειτουργίες των εφαρμογών αυτών.

Μάθημα L5.2

To widget Text

Προγραμματίζω
με την python 


- V5.2.1 Το γραφικό αντικείμενο Text
- V5.2.2 Η βιβλιοθήκη Scrolltext
- V5.2.3 Παράδειγμα: Το Ελληνικό Ημερολόγιο
- V5.2.4 Παράδειγμα: Ενδιαφέροντα λήμματα

V5.2.1

Το γραφικό αντικείμενο Text

Νίκος Αβούρης, Πανεπιστήμιο Πατρών


Text : υποδοχέας
μορφοποιημένου κειμένου

Προγραμματίζω
με την python 

Επεξεργαστής κειμένου, το οποίο μπορούμε να
μορφοποιήσουμε και να επεξεργαστούμε

```
t = tk.Text(παράμετροι)
t.delete('1.0', 'end')
t.insert('end', astring)
something = t.get('1.0', 'end')
```

Text : υποδοχέας
μορφοποιημένου κειμένου

Προγραμματίζω
με την python 


```
t = tk.Text(παράμετροι)
t.search(pattern, '1.0', stop='5.end')
t.see('100.0') # κύλησε ώστε να φαίνεται
η γραμμή 100
t.image_create('1.0', image = photo)
```

V5.2.2

Διαχείριση κειμένου με ετικέτες

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

Ετικέτες της tk.Text()

Προγραμματίζω
με την python 

→ ορίζονται κατά την εισαγωγή του κειμένου

```
t.insert('1.0', astring, 'title', 'section')
```


tags

→ ή εκ των υστέρων

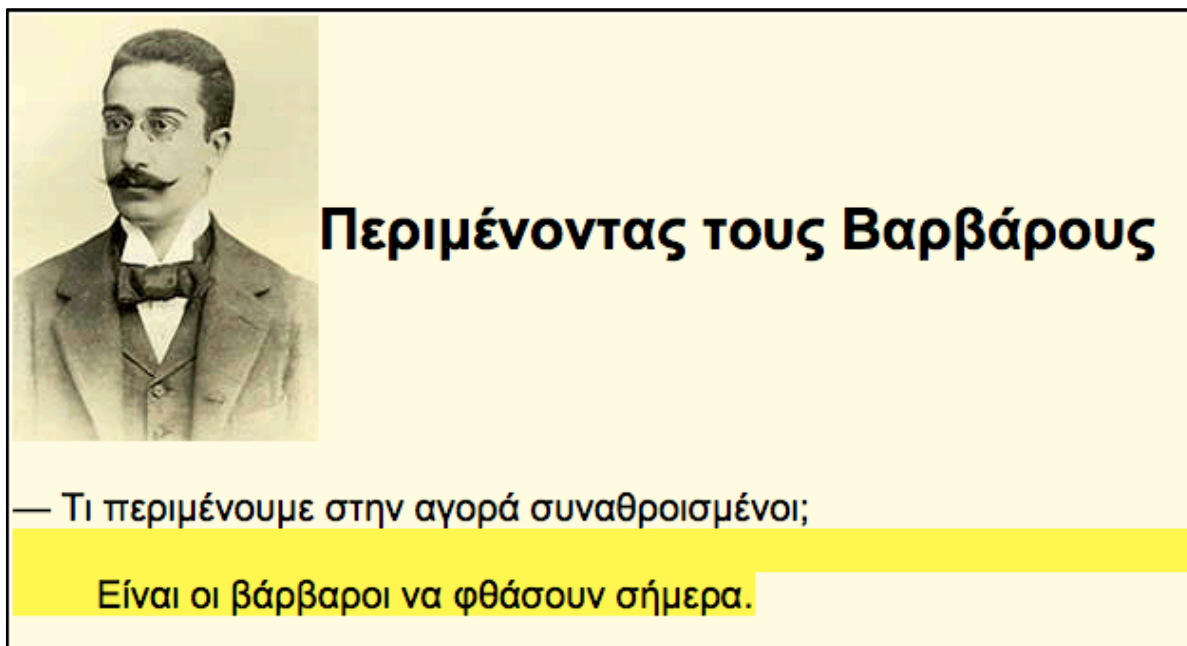
```
t.tag_add('title', '1.0', '2.end')
```

tags

`t.tag_config(ετικέτα, επιλογές)`

Προγραμματίζω
με την python 

`t.tag_config('sel', background= 'yellow')`



ετικέτα 'sel'
μπαίνει αυτόματα
από το Text στο
κείμενο που
επιλέγει ο χρήστης

δεικτοδότηση κειμένου

'1.c' line.char, πχ '5.9' (5^η γραμμή χαρακτήρας 10)


- `indx+n chars`, πχ '5.9+5c' (5 χαρακτήρες πιο δεξιά από την προηγούμενη θέση)
- `indx+n lines`, πχ '5.9+5l' (5 γραμμές πιο κάτω από την προηγούμενη θέση)
- `linestart, lineend,`
- `wordstart, wordend`

V5.2.3

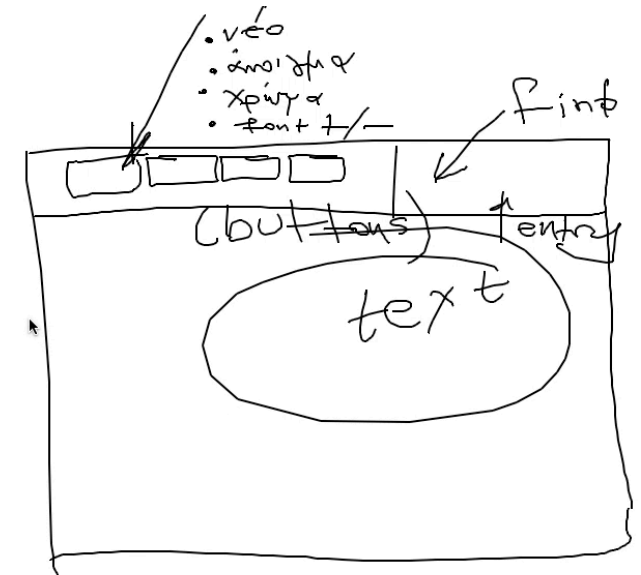
Παράδειγμα: MyEdit

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

MyEdit

Προγραμματίζω
με την python 

Να σχεδιάσετε ένα πρόγραμμα επεξεργασίας αρχείων κειμένου. Να περιέχει στο πάνω μέρος ένα πλαίσιο με πλήκτρα (Νέο.., Άνοιγμα.., Χρώμα, Αύξηση/μείωση μεγέθους χαρακτήρων, πλαίσιο αναζήτησης)
Να περιλαμβάνει επίσης μπάρα κύλισης




V5.2.4

Παράδειγμα: MyEdit (μέρος Β)

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

Μάθημα L5.3

To widget Canvas

Προγραμματίζω
με την python 


- V5.3.1 Μέθοδοι και ταυτότητες αντικειμένων
- V5.3.2 Canvas: παραδείγματα
- V5.3.3 Canvas: το γεγονός <B1-Motion>
- V5.3.4 Παράδειγμα: ελαστικό ορθογώνιο
- V5.3.5 Παράδειγμα: Το παιχνίδι "Γέμισε το κουτί"

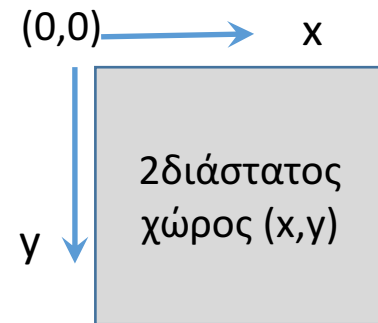
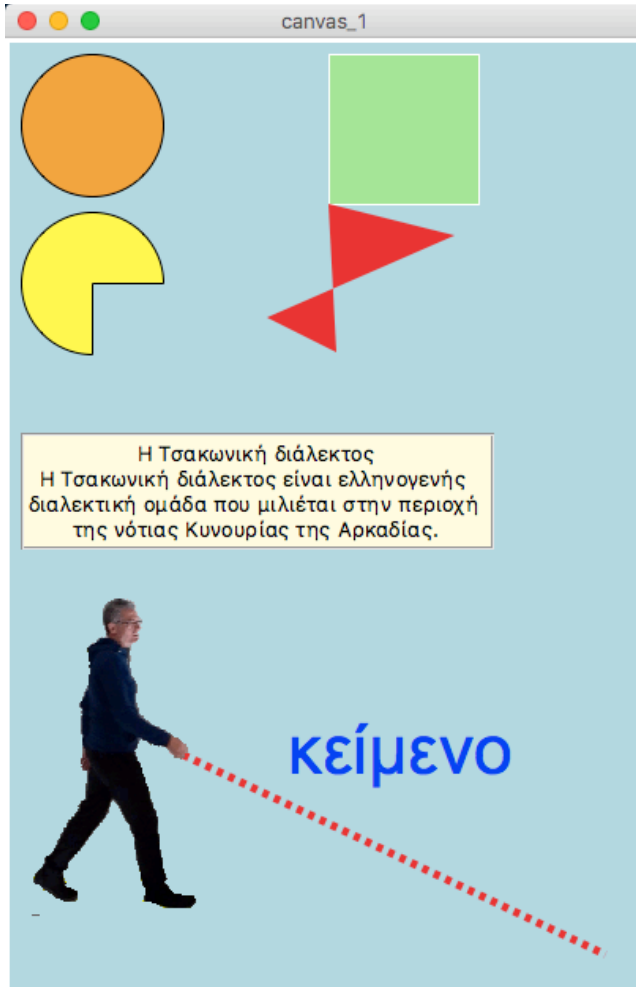
V5.3.1

Canvas: Μέθοδοι και ταυτότητες αντικειμένων

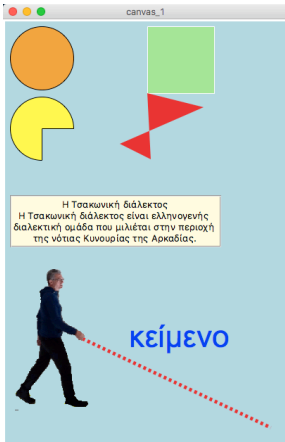
Νίκος Αβούρης, Πανεπιστήμιο Πατρών

Canvas : υποδοχέας γραφικών αντικειμένων, εικόνων κλπ.


Προγραμματίζω
με την python 



```
self.canvas.create_oval( ) # κύκλος  
self.canvas.create_rectangle() # ορθογώνιο  
self.canvas.create_arc() # τόξο κύκλου  
self.canvas.create_polygon() # πολυγωνική γραμμή  
self.canvas.create_window()#παράθυρο με widget  
self.canvas.create_image() #εικόνα PhotoImage  
self.canvas.create_text( ) #κείμενο  
self.canvas.create_line( ) # γραμμή
```

Canvas: παράδειγμα εισαγωγής αντικειμένων

Προγραμματίζω
με την python 

```
self.canvas = tk.Canvas(self.root, width=400, height=600, bg='lightblue')
self.canvas.pack()
self.canvas.create_oval(10, 10, 100, 100, fill='orange') # πορτοκαλί κύκλος
self.canvas.create_line(105, 450, 380, 580, fill="red", dash=(4, 4), width=5) # κόκκινη γραμμή
self.canvas.create_rectangle(205, 10, 300, 105, outline='white', fill='lightgreen') # πράσινο κουτί
xy = 10, 110, 100, 200
self.canvas.create_arc(xy, start=0, extent=270, fill='yellow') # τόξο κύκλου
self.canvas.create_polygon(205, 105, 285, 125, 166, 177, 210, 199, 205, 105, fill='red') # πολύγωνο
self.canvas.create_text(250, 450, text='κείμενο', fill='blue', font='Arial 44') #κείμενο
self.canvas.img = tk.PhotoImage(file='walking.gif')
self.canvas.create_image(10, 350, image=self.canvas.img, anchor='nw') #εικόνα
frm = tk.Frame(self.canvas, relief='groove', borderwidth=2)
tk.Label(frm, text= self.insert_quote(), bg='lightyellow').pack()
self.canvas.create_window(10, 250, window=frm, anchor='nw') #παράθυρο μέσα στον καμβά
```

Canvas : ταυτότητα αντικειμένων

Κάθε αντικείμενο που δημιουργείται στον καμβά αποκτά μια μοναδική ταυτότητα.

Μπορούμε να βρούμε τις ταυτότητες όλων των αντικειμένων με τη μέθοδο `canvas.find_all()`.

Επίσης μπορούμε να επισυνάψουμε ετικέτες (tag) σε αντικείμενα με τη μέθοδο `canvas.addtag()` και να τις διαγράψουμε με τη μέθοδο `canvas.dtag()`


Με τη μέθοδο `find_with_tag()` μπορούμε να βρούμε όλα τα αντικείμενα που έχουν μια ετικέτα.

Η ειδική ετικέτα `'current'` δίνεται στο αντικείμενο που βρίσκεται στη δεικτική συσκευή του χρήστη (αν υπάρχει).

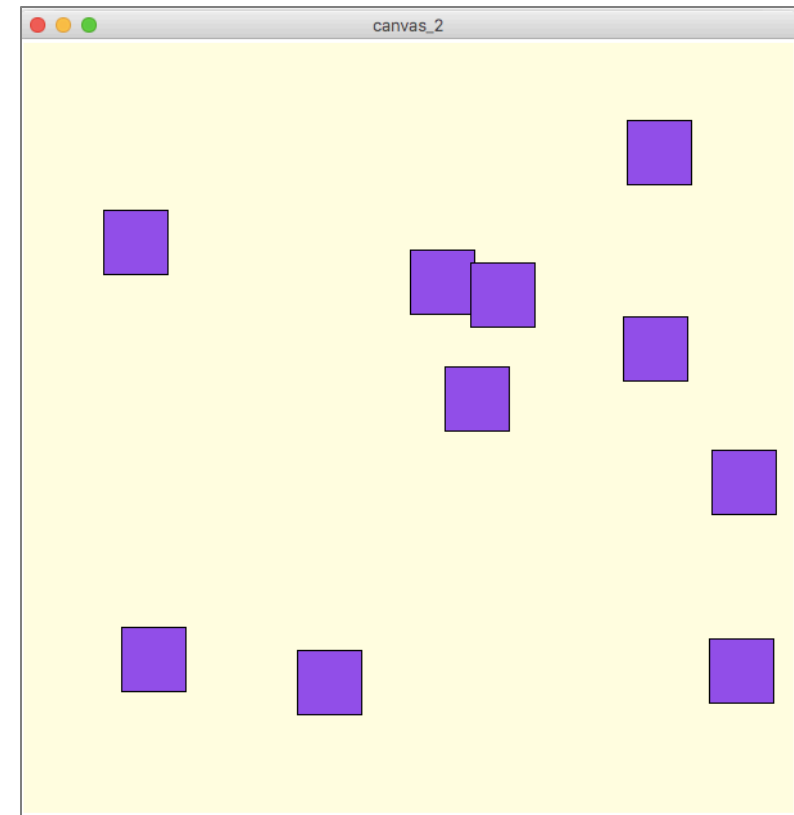
Canvas : μέθοδοι καμβά

- .coords(id, x0, y0, x1, y1)** οι συντεταγμένες του αντικείμενο id
- .delete(id)** διαγράφεται το αντικείμενο id
- .itemcget(id, option)** η τρέχουσα τιμή του γνωρίσματος
- .bbox(id)** bounding box το ορθογώνιο πλαίσιο μέσα στο οποίο είναι το αντικείμενο id
- .itemconfig(id, **options)** τροποποιεί τα γνωρίσματα του αντικειμένου id
- .find_closest(x, y)** επιστρέφει το πλησιέστερο αντικείμενο στη θέση x,y
- .find_enclosed(x1, y1, x2, y2)** τα αντικείμενα που περιέχονται πλήρως στο ορθογώνιο, παραλαγή: **find_overlapping(x1, y1, x2, y2)**
- .move(item, dx, dy)** μετακινεί το αντικείμενο κατά dx, dy
- .tag_bind(item, event, callback)** χειριστής για γεγονότα

Canvas : Άσκηση moverandomsquares

Προγραμματίζω
με την python 

Ζωγραφήστε ένα καμβά 600x600 και γεμίστε τον με 10 τυχαία τετράγωνα όπως στο σχήμα. Ορίσετε την εξής συμπεριφορά:
Όταν ο χρήστης πατάει το αριστερό πλήκτρο του ποντικιού, όλα τα τετράγωνα κινούνται προς τα κάτω κατά 20 pixel, ενώ όταν πατάει το δεξί πλήκτρο, προς τα πάνω αντίστοιχα.




V5.3.2

Canvas: παραδείγματα

Νίκος Αβούρης, Πανεπιστήμιο Πατρών


παράδειγμα : πώς μετακινούμε όλα
τα αντικείμενα κατά dx με κάθε κλικ;

Προγραμματίζω
με την python 

```
self.canvas.bind('<1>', self.move_all)
```

```
def move_all(self, event):  
    for i in self.canvas.find_all():  
        self.canvas.move(i, 10, 0)
```


παράδειγμα : πώς δημιουργούμε ένα νέο κύκλο μετά από κάθε κλικ;

Προγραμματίζω
με την python 

```
self.canvas.bind('<1>', create_circle)
def create_circle(self, event):
    self.c = self.circle(event.x, event.y, 50, fill='orange')
def circle(self, x,y,r, **kwargs):
    return self.canvas.create_oval(x-r, y-r, x+r, y+r, **kwargs)
```

ερώτηση: πώς θα μπορούσε να παραληφθεί η μέθοδος create_circle με χρήση ανώνυμης συνάρτησης;


παράδειγμα : πώς επιλέγουμε ένα κύκλο αλλάζοντας προσωρινά το χρώμα του ;

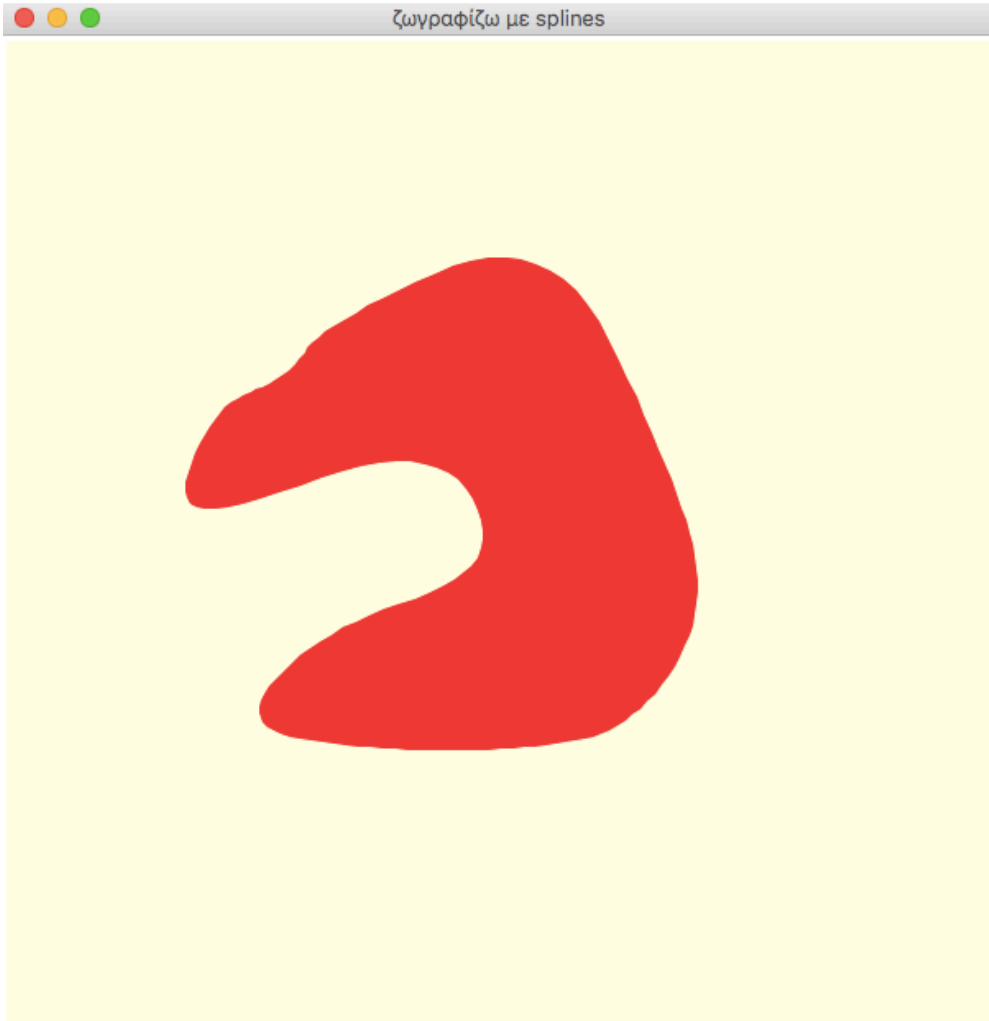
Προγραμματίζω
με την python 

```
self.canvas.bind('<1>', self.click)
self.canvas.bind('<ButtonRelease-1>', self.back)

def click(self, event):
    if self.canvas.find_withtag('current'):
        self.canvas.itemconfig('current', fill="blue")
def back(self, event):
    if self.canvas.find_withtag('current'):
        self.canvas.itemconfig('current', fill="red")
```


παράδειγμα : ζωγραφίζω με καμπύλες spline

Προγραμματίζω
με την python 



Χρησιμοποιούμε την
`create_polygon` με
ενεργοποιημένη την παράμετρο
`smooth=True`


```
self.canvas.create_polygon(*points, smooth=1,  
fill = 'red', tags='spline')
```

V5.3.3

Canvas: το γεγονός <B1-Motion>

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

`c.bind("<B1-Motion>", συνάρτηση)`

Προγραμματίζω
με την python 

Χειρισμός κίνησης με πατημένο πλήκτρο. παράδειγμα:
σχεδιάζουμε ελεύθερες γραμμές με το ποντίκι (ή το χέρι)

```
self.canvas.bind("<Button-1>", self.xy)
self.canvas.bind("<B1-Motion>", self.add_line)
self.lastx, self.lasty = 0,0
```

```
def xy(self, event):
    self.lastx, self.lasty = event.x, event.y
```


```
def add_line(self, event):
    self.canvas.create_line((self.lastx, self.lasty, event.x,
event.y))
    self.lastx, self.lasty = event.x, event.y
```

V5.3.4

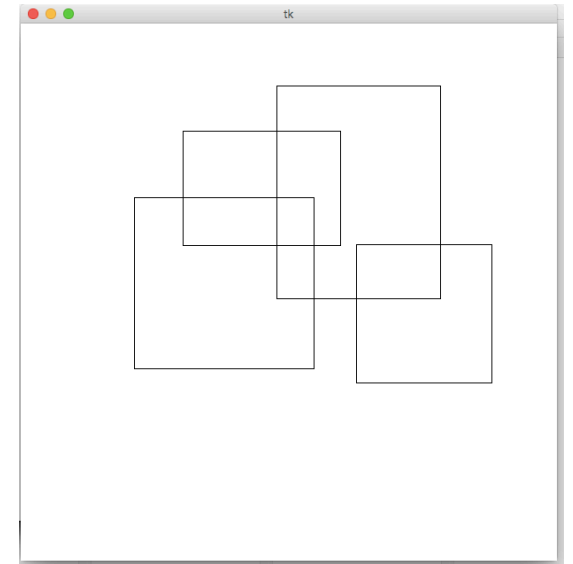
Παράδειγμα: ελαστικό ορθογώνιο

Νίκος Αβούρης, Πανεπιστήμιο Πατρών

παράδειγμα : ελαστικό ορθογώνιο

Προγραμματίζω
με την python 

```
class MyApp():
    def __init__(self, root):
        self.root = root
        self.canvas = tk.Canvas(self.root, width=600, height=600)
        self.canvas.pack()
        self.rubberbandBox = None
        self.canvas.bind("<Button-1>", self.mouse_down)
        self.canvas.bind( "<Button1-Motion>", self.mouse_motion)
        self.canvas.bind( "<ButtonRelease-1>", self.mouse_up)
    def mouse_down(self, event):
        self.startx = event.x
        self.starty = event.y
        self.rubberbandBox = self.canvas.create_rectangle(self.startx, self.starty, event.x, event.y)
    def mouse_motion(self, event):
        self.canvas.coords(self.rubberbandBox, self.startx, self.starty, event.x, event.y)
    def mouse_up(self, event):
        self.canvas.create_rectangle(*self.canvas.coords(self.rubberbandBox), width=2)
        self.canvas.delete(self.rubberbandBox)
```




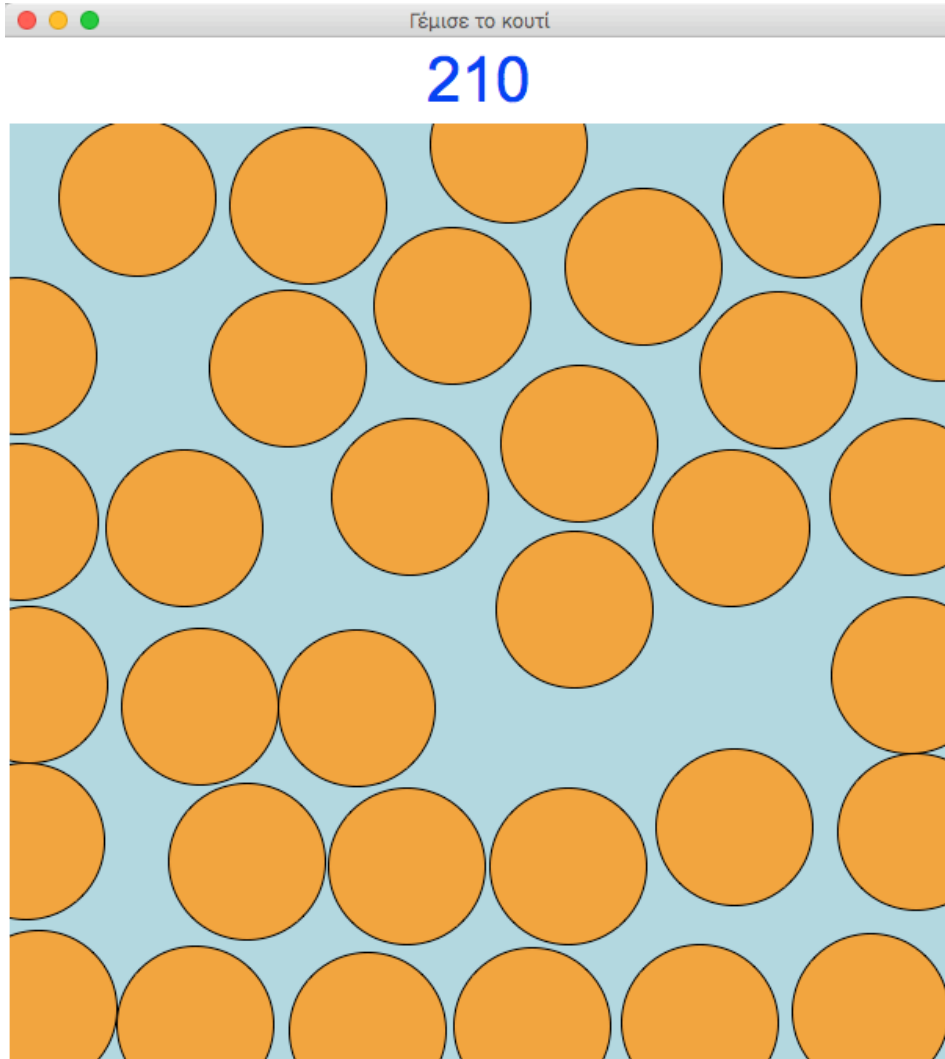
V5.3.5

Παράδειγμα: Το παιχνίδι "Γέμισε το κουτί"

Νίκος Αβούρης, Πανεπιστήμιο Πατρών


παιχνίδι: γέμισε το κουτί

Προγραμματίζω
με την python 



Γέμισε με όσο περισσότερες μπάλες μπορείς το κουτί, αν ακουμπήσεις μια μπάλα που υπάρχει ήδη χάνεις 5 πόντους. Κάθε νέα μπάλα που προσθέτεις σου δίνει 10 πόντους

παράδειγμα : μετακίνηση αντικείμενου στη θέση του κλικ


Προγραμματίζω
με την python 

```
def __init__(self, root):
    self.root = root
    root.title("moving blue box")
    root.geometry("600x400+300+300")
    self.create_canvas()
    self.rect = None
    self.size=100

def create_canvas(self):
    self.canvas = tk.Canvas(self.root, bg='lightyellow')
    self.canvas.pack(fill='both', expand=1)
    self.canvas.bind("<Button-1>", self.new_box)

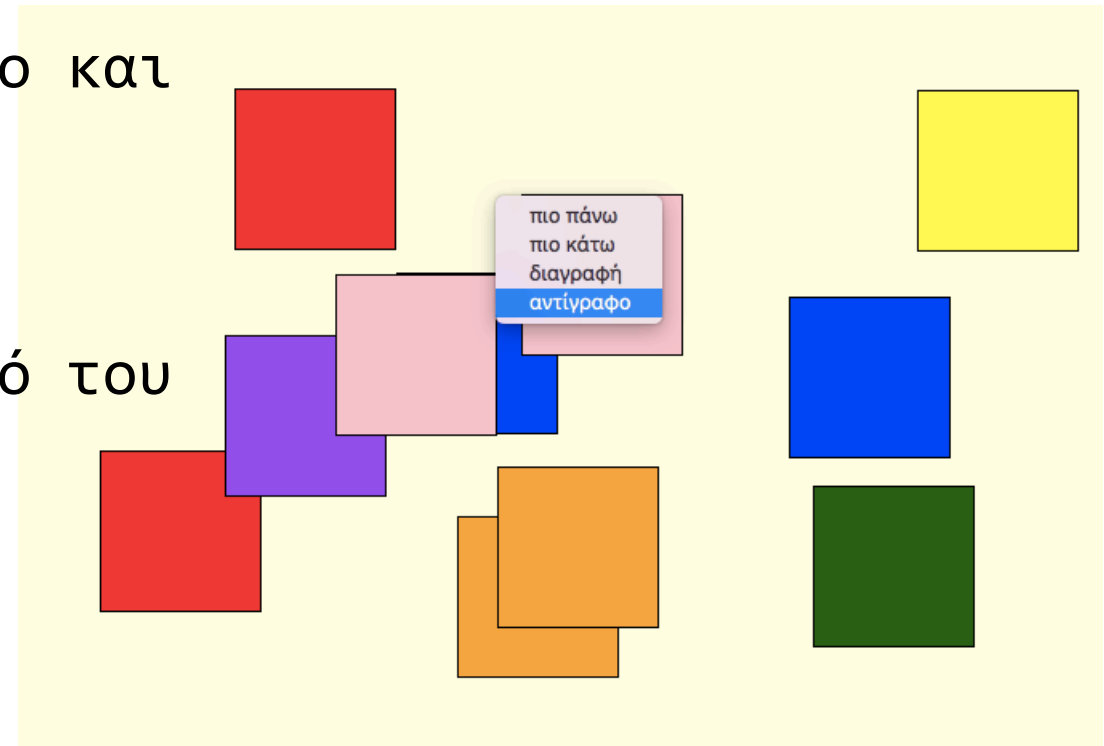
def new_box(self, ev):
    if not self.rect:
        self.rect = self.canvas.create_rectangle(ev.x, ev.y,
                                                    ev.x+self.size, ev.y+self.size, fill='blue')
    else:
        xy = self.canvas.coords(self.rect)
        dx,dy = ev.x-xy[0], ev.y-xy[1]
        self.canvas.move(self.rect, dx, dy)
```


Τελική εργασία εβδομάδας 5

Προγραμματίζω
με την python 

Να κατασκευαστεί εφαρμογή που έχει την εξής συμπεριφορά:

1. με διπλό κλικ στον καμβά δημιουργείται τετράγωνο τυχαίου χρώματος
2. μπορεί να επιλεγεί κάποιο τετράγωνο και να συρθεί σε νέα θέση
3. με δεξί κλικ το τετράγωνο που επιλέχτηκε μπορεί να αναδυθεί ή να κρυφτεί ή να δημιουργηθεί αντίγραφό του




Άσκηση

(για συζήτηση)

Να κατασκευάσετε μια αριθμομηχανή 4 πράξεων, με μια θέση μνήμης.


Η αριθμομηχανή θα πρέπει να έχει λειτουργίες όπως στο σχήμα: **+m**

(αποθήκευση στη μνήμη), **mr** (ανάκτηση από τη μνήμη), **mc** (καθαρισμός μνήμης), **c** (καθαρισμός οθόνης).

Προγραμματίζω
με την python 



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

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