

# Διαδραστικά Πολυμέσα II

2015 - 2016

Διδάσκων: Θεόδωρος Θεμιστοκλέους

# Functions

- Function (λειτουργία) είναι ένα κομμάτι κώδικα μέσα σε ένα πρόγραμμα το οποίο εκτελεί ένα συγκεκριμένο έργο. Τα functions είναι δυνατά εργαλεία προγραμματισμού που κάνουν τα προγράμματα εύκολα να διαβαστούν και να αλλάξουν.
- Ως συνήθως ένα function που δεν επιστρέφει (return) κάποιο αποτέλεσμα ξεκινά με void

```
void brick (float x, float y, float sizeX, float sizeY, color c)
{

    fill(c);
    rectMode(CORNER);
    rect(x, y, sizeX, sizeY);

    for (int i=0; i<4; ++i)
    {

        fill(0);
        rectMode(CENTER);
        rect(x+((sizeX/5)*(i+1)), y+(sizeY/4), sizeX/16, sizeY/8);
        rect(x+((sizeX/5)*(i+1)), y+((3*sizeY)/4), sizeX/16, sizeY/8);

    }

}
```

# Objects

- Το object oriented programming είναι ένας τρόπος να οργανωθεί ο κώδικας σε objects, δηλαδή κομμάτια κώδικα που περιέχουν και data και functions. Για παράδειγμα το brick function μπορεί να επεκταθεί και να αποτελέσει μέρος ενός class definition. Τα objects δημιουργούνται χρησιμοποιώντας το class ως template
- Concept of Objects

Αυτοκίνητο

## Χαρακτηριστικά(variables)

Μάρκα  
Χρώμα  
Είδος  
Μέγεθος



Honda  
Κόκκινο  
SUV  
1.5m x 5.0m

## Λειτουργίες (functions)

Κίνηση  
Ταχύτητα  
Κατεύθυνση



Μπροστά  
100Km/h  
Προς Λεμεσό

# Objects

- Το class αποτελεί την γενικότερη ιδέα του object. Για παράδειγμα στην περίπτωση του αυτοκινήτου το class αποτελεί την βάση για την δημιουργία ενός object που θα ανήκει στην κατηγορία αυτοκίνητο και θα έχει συγκεκριμένο χρώμα, μάρκα, διαστάσεις και τύπο αυτοκινήτου. Επίσης θα μπορεί να κινηθεί προς οποιαδήποτε κατεύθυνση του υποδείξουμε και να αναπτύξει συγκεκριμένη ταχύτητα.

```
class Car
{
    color col; //colour
    float x,y; //location
    float c_width, c_length; //dimensions
    String brand;
    String type;
    float speedX,speedY;
```

*Τα χαρακτηριστικά του αυτοκινήτου – τα local variables*

*Ο constructor θα πάρει τα  
συγκεκριμένα χαρακτηριστικά που θα  
ορίσουμε για να φτιάξει το object car*



```
Car(float cx, float cy, float c_l, float c_w, String cBrand, String cType, color c) //constructor
{
    x= cx;
    y=cy;
    c_width = c_w;
    c_length = c_l;
    brand = cBrand;
    type = cType;
    col = c;
}
```

*Ο constructor δίνει τις τιμές που ορίσαμε στα local variables για να  
δημιουργηθεί το object car*

# Objects

```
void draw()  
{  
    fill(col);  
    rect(x,y,c_width,c_length);  
}
```



*Τα functions αν και μπορούν να πάρουν οποιοδήποτε όνομα εντούτοις προσπαθούμε να υποδηλώνει την λειτουργία του κώδικα*

```
void move()  
{  
    x = x+speedX;  
    y = y+speedY;  
}
```



*Function για την κίνηση του object car*

```
void accelerate(int k)  
{  
    if(k==1) speedY+=1;  
    if(k==0) speedY-=1;  
    if(k==2) speedX+=1;  
    if(k==3) speedX-=1;  
}  
  
}
```



*Function για την επιτάχυνση του object car*

# Objects

- Προσδιορίζοντας ένα class δημιουργούμε τα δικά μας data types. Σε αντίθεση με τους βασικούς τύπους int, float και boolean είναι σύνθετοι τύποι όπως Pimage και Pfont που σημαίνει ότι μπορεί να περιέχουν αρκετές μεταβλητές (variables) και μεθόδους (functions) μέσα σε ένα όνομα.
- Όταν δημιουργούμε ένα class πρέπει πρώτα να σκεφτούμε προσεχτικά τι θέλουμε να κάνει ο κώδικας και έπειτα να κάνουμε μια λίστα με τις μεταβλητές που χρειάζονται
- Το όνομα του class μπορεί να είναι οτιδήποτε αλλά είναι ιδιαίτερα βοηθητικό να γράψουμε το πρώτο γράμμα με κεφαλαίο,
- Έχουμε πρόσβαση στα field και methods ενός object με το dot operator, την τελεία.
  - για να πάρουμε την τιμή μιας μεταβλητής χρησιμοποιούμε π.χ το `car.x`;
  - για να καλέσουμε μια μέθοδο του object τότε χρησιμοποιούμε π.χ το `car.draw()`;

# Objects

```
Car car;
```



Πρώτα δηλώνω το όνομα του *object* και ποιο *class* να χρησιμοποιηθεί για να το δημιουργήσω

```
void setup()
{
  size(800, 800);
  car = new Car(width/2,height/2,100,20,"Honda","SUV",color(200,100,170));
}
```

Δίνω τα χαρακτηριστικά στον *constructor* του *class* για να δημιουργήσει το *object*

```
void draw()
{
  background(0);
  car.draw();
  car.move();
  text(car.speedX,20,20);
  text(car.speedY,20,40);
}
```



Καλώ το *function* που θα εμφανίσει το *object* και το *function* για να κινηθεί

```
void keyPressed()
{
  if (key == CODED)
  {
    if (keyCode== UP) car.accelerate(0);
    if (keyCode== DOWN) car.accelerate(1);
    if (keyCode== LEFT) car.accelerate(3);
    if (keyCode== RIGHT) car.accelerate(2);
  }
}
```



Καλώ το *function* που θα επιταχύνει το αυτοκίνητο ανάλογα με τα πλήκτρα που πιέζω

# Array of Objects

- Δουλεύοντας με τα array of objects είναι παρόμοιο με το να χρησιμοποιούμε arrays για άλλους τύπους δεδομένων

```
Car [] car;
int n;
void setup()
{
    size(800, 800);
    car = new Car[10];
    for(int i=0; i<10; ++i)
    {
        car[i] = new Car (random(100,width-100),random(100,height-100),random(40,100),random(5,20),"Honda"+i,
        "SUV",color(random(0,255),random(0,255),random(0,255)));
    }
}

void draw()
{
    background(0);
    for(int i=0; i<10; ++i)
    {
        car[i].draw();
        car[i].move();
    }

    text(car[n].speedX,20,20);
    text(car[n].speedY,20,40);
}
```



# Array of Objects

```
void keyPressed()
{
  if(key == 'a' && n < 9)
  {
    n += 1;
  }
  if(key == 'z' && n > 0)
  {
    n -= 1;
  }
  if (key == CODED)
  {
    if (keyCode == UP) car[n].accelerate(0);
    if (keyCode == DOWN) car[n].accelerate(1);
    if (keyCode == LEFT) car[n].accelerate(3);
    if (keyCode == RIGHT) car[n].accelerate(2);
  }
}
```

# References

Coding for this session:

- <https://processing.org/reference/>
- REAS, C., & FRY, B. (2007). ***Processing: a programming handbook for visual designers and artists***. Cambridge, Mass, MIT Press. **PAGES 395-411**

Βιβλία:

- GREENBERG, I., & KEITH, P. (2007). ***Processing creative coding and computational art***. Berkeley, CA, Friends of Ed, an Apress Co.
- REAS, C., & FRY, B. (2007). ***Processing: a programming handbook for visual designers and artists***. Cambridge, Mass, MIT Press.
- SHIFFMAN D. (2009), ***Learning Processing A Beginner's Guide to Programming Images, Animation, and Interaction***, Morgan Kaufmann, 2009

# Άσκηση για το σπίτι 7

Να αναπτύξετε περαιτέρω το sketch arcanoid.pde προσθέτοντας βαθμό δυσκολίας σε σχέση με την πρόοδο του χρήστη και παράλληλα να δημιουργήσετε γραφικές απεικονίσεις που να ειδοποιούν τον χρήστη για τον βαθμό δυσκολίας. Ζητούμενο της άσκηση είναι να προσθέσετε τουλάχιστον δυο από τα πιο κάτω:

- να μικραίνει το slider για κάποιο χρονικό διάστημα και τα bricks να αλλάζουν χρώμα (προσωρινή παραμόρφωση)
- η ταχύτητα της μπάλας να αυξάνεται για κάποιο χρονικό διάστημα όπως και το μέγεθος της και παράλληλα να αλλάζει και το χρώμα της (προσωρινή παραμόρφωση)
- τα Bricks να μετακινούνται προς τα κάτω σε σχέση με τον χρόνο που περνά ή με τον αριθμό των Bricks που καταστράφηκαν
- Το slider να μικραίνει σε σχέση με το χρόνο (μόνιμη παραμόρφωση)
- Να δημιουργηθεί σύστημα βαθμολογίας
- Όταν η μπάλα βγει από την οθόνη να τελειώνει το παιχνίδι