

Python – variabile, asignari (tuplu), operatii matematice, comparatii, if, for, functii, liste, matrice, random, modul, IDE???

Mate – sistem cartezian de coordonate

Fizica – viteza, acceleratie, culori RGB

## **Pasi implementare:**

Z1 - Desenarea unui dreptunghi

Z1/2 - Desenarea unei matrice de dreptunghiuri

Z2 - Desenarea unei matrice de dreptunghiuri colorate

Z2/3 - Desenarea padului

Z3 - Miscarea padului – acceleratie + viteza

Z3/4 - Desenarea bilei si Miscarea bilei la inceput sincron cu padul

Z4/5 - Coliziunea bilei cu peretii si padul

Z5 - Coliziunea bilei cu brick-urile

Z5 - Sistemul de vietii

## **Idei de dezvoltare:**

Sistem de scor (incrementare sau variabil dupa culoarea dreptunghiului de exemplu?). Simplut

Unghi variabil la coliziunea bilei cu padul!!! Face jocul mai distractiv pentru ca ofera o mai mare implicare a jucatorului prin pozitionarea strategica a padului la fiecare coliziune.

Ceva mai greu. Frumos e cu trigonometrie, dar nicio sansa la clasa a 5-a

Poate se poate implementa cu o regula de 3 simpla? La mijloc  $45^\circ$ , la margini  $5^\circ$  ? Sau random...

Marirea vitezei mingii la 10 secunde. Se poate folosi Clock din pgz. Simplut de facut :)

Brickuri cu efecte speciale:

- marirea/micsorarea vitezei bilei

- marirea/micsorarea padului

Efectele s-ar putea distinge dupa culoare de exemplu

# Lectia 1

Demo joc

Cod tinta:

```
import pgzrun

WIDTH = 500
HEIGHT = 500

size = 100, 100
poz = 100, 100

brick = Rect(poz, size)

color = 255, 0, 0

def draw():
    screen.draw.filled_rect(brick, color)

pgzrun.go()
```

Etape:

1. **Ce este o variabila?**  
Sa faca in promptul din IDLE exercitii de declarare si operatii aritmetice  
Sa declare 2-3 variabile si sa faca operatii aritmetice
2. **Ce este un tuplu?** O variabila cu mai multe valori despartite prin virgula  
Lucrul cu tuplu din prompt  
Afisarea unui tuplu cu print  
Accesul la elemente prin **indexare [ ] - de la 0!!**  
Operatii: suma cu int (nu merge), suma cu tuplu (concat), inmultire cu int (repeat), inmultire cu tuplu (nu merge), scaderea si impartirea nu merg
3. **Ce este o functie?**  
Explicarea ideii din spatele functiei, fara declaratie/implementare  
Sa creeze un fisier de python in care sa declare variabile si sa le afiseze cu **print**
4. Sa afisam o fereasta goala – sa scrie prima si ultima linie de cod  
Sa adauge variabilele width si height dupa modelul meu  
Sa modifice valorile de la width si height pentru a intelege dimensiunea ferestrei
5. Le dau codul tinta pentru aceasta lectie  
Explicatii in mare – creeaza un dreptunghi, ii definesc culoarea  
Explicatii functia **draw** raportata la pgz. Ea e apelata pt fiecare cadru  
!! Explicatii asupra indentarii
6. **Ce este un sistem de coordonate?**  
Desenat pe flipchart exemple cu puncte  
Explicat cum e format un dreptunghi din coltul stanga sus si latime/inaltime  
Sa modifice valorile de la size/poz pentru a le intelege

## 7. Ce este o culoare RGB?

Comparatie cu sistemul de culori din pictura – 3 culori de baza – rosu, galben, albastru – discutat combinatii

Sistemul folosit nu e bazat pe vopsea, ci pe lumina si atunci culorile de baza sunt RGB – discutat combinatii si ce inseamna o valoare intre 0 si 255

## 8. Colorarea backgroundului ecranului (screen.clear(); screen.fill(color); )

# Lectia 2

Scop – desenarea unei matrice de dreptunghiuri de aceeasi culoare

Cod tinta: l2.py

Etape:

### 1. Ce este o lista?

Lucru in prompt

Sa declare un tuplu si sa incerce sa ii atribui o alta valoare => **imutabilitate**

Exemplificarea declaratiei unei liste cu [ ]

Accesul tot prin []

Sa mai incerce o data modificarea unei valori si sa o afiseze => **mutabilitate**

Operatii aritmetice cu lista: suma cu int (nu merge), suma cu lista (concat), inmultire cu int (repeat), inmultire cu lista (nu merge), scaderea si impartirea nu merg

Functia **list** pentru a converti un tuplu

Metoda **append**

### 2. Functia **range** pentru a genera o lista de la 0 la stop

### 3. Ciclu repetitiv **for**

Pentru o lista cu multe elemente (100 de exemplu) sa se adune un alt numar. Sa faca asta manual, apoi o sa le arat cu un for

!! Explicatii asupra indentarii

Range-based for – i e copie

```
l = [1, 2, 3]
```

```
for i in l:
```

```
    i = 5
```

```
    print(i)
```

C-style for

```
l = [1, 2, 3, 4]
```

```
for i in range(4):
```

```
    l[i] = l[i] + 1
```

Merge si pe list, si pe tuplu

Exemplu de creare de lista cu for si append

Exercitiu: sa contruiasca o lista cu patratele iteratorului

### 4. Ce este o matrice? Lista in lista

Versatilitatea limbajului de a defini orice in orice variabila oricand

Declaratie de lista in lista hardcodata

Conceptele de linie si coloana

Accesul prin [ ][ ]

Exercitiu: sa scrie ei o lista cu 2 linii si 3 coloane; pe prima linie sa fie nr pare si pe a 2 a impare

- Parcurgerea unei matrice cu 2 for-uri
- Crearea unei matrice cu append
- 5. Discutie introductiva asupra unei matrice de dreptunghiuri si pozitionarea lor
- Necesitatea definirii unei latimi/inaltimi a dreptunghiului, a spatiului despartitor dintre ele si a distantei fata de tavan si peretele stang
- 6. Le dau codul tinta daca nu mai e timp – altfel poate incerc o implementare cu ei... macar un schelet, chiar daca nu merge- poate pseudocod like
- 7. Discutie pe formulele din cod

## Lectia 3

Sa genereze procedural culorile dreptunghiurilor

**Exercitiu:** sa generam o culoare aleatorie pentru background la fiecare rulare  
Prezentarea bibliotecii de random

Sa genereze 3 variabile in intervalul [0, 255], sa le puna in tuplu si sa il paseze in draw  
Codul nu o sa fie intr-o functie la momentul actual

**Exercitiu:** in baza lectiei anterioare, vom lua in discutie cum putem face astfel incat fiecare dreptunghi sa aiba o culoare proprie.

Reluarea discutiei asupra functiei **draw** si asupra metodei de a desena un dreptunghi

Copiii trebuie sa realizeze ca trebuie sa salvam cate o culoare pt fiecare dreptunghi si sa fie pasata metodei in mod corespunzator.

O sa fie nevoie de o matrice de culori

Dupa modelul generarii de dreptunghiuri, fie scriu copiii, fie scriu eu impreuna cu ei codul. Sau daca nu, le dau codul si discutam pe el

Vor introduce in matrice tuplu-uri RGB

Extra exercitiu: sa transformam cele 3 generari in functii

## Lectia 4

**Exercitiu:** Sa desenam padul pe centru ferestrei in partea de jos

Copii vor trebui sa defineasca latimea si inaltimea dreptunghiului, culoarea, sa instantieze un Rect si sa apeleze metoda screen.draw.filled\_rect - asemanator cu lectia 2

Ii pot ghida prin intrebari de tipul “prin ce am definit pana acum un dreptunghi”, “unde vrem sa definim padul”

Ii pot ghida cu coordonatele facand desene pe un sistem cartezian pe flipchart

La inceput o sa le spun sa hardcodeze valorile

Apoi o sa le spun sa schimbe dimensiunea ferestrei si o sa vada ca nu mai e centrat padul

O sa incercam sa calculam pozitia centrala a padului (pot renunta la pasul acesta daca vad ca nu se descurca sau il pot face extra/bonus)

## Lectia 5

### Miscarea padului in mod accelerat

Etape:

1. **Ce este viteza?**

Discutie asupra masurarii timpului si a distantei

Discutie asupra cunostintelor lor despre viteza

Pot sa le fac un tabel in care sa pun variatia distantei in functie de timp pt o anumita viteza

Copii trebuie sa inteleaga ca este o simpla adunare a vitezei la distanta pentru fiecare moment de timp

Aducerea discutiei in spatiul sistemului de coordonate

Cum s-ar misca un punct pe axa orizontala cu viteza  $v$ ? Din nou tabel cu exemplu

2. Definirea unei variabile de viteza pentru pad. Nenumita

3. Functia **update** din pgz – apelata de 60 de ori pe secunda pentru logica jocului

Initial putem creste componenta  $x$  a padului si copiii o sa vada cum padul pleaca la dreapta.

Apoi sa incerce sa scada

Noi vrem sa se miste padul doar cand apasam tastatura si atunci trebuie sa verificam daca s-a apasat o tasta sau nu

Intrerupere pentru a explica if-ul

4. **Expresii booleene**

In prompt sa exerseze operatorii de comparatie cu numere si cu variabile

O sa observe existenta valorilor de adevar care pot fi salvate in variabile

O sa le prezint apoi cele 3 operatii logice cu tabelele de valori

Sa le testeze in prompt

5. **Structura alternativa if**

O sa le dau cateva exemple in care facem lucruri diferite in functie de anumite conditii

O sa le prezint if-ul simplu si cu else

Exercitiu: in fisier, pt 2 numere date, sa il afiseze pe cel care e mai mare

Exercitiu 2: pentru 2 numere date, sa se afiseze suma daca primul e mai mare, numarul daca sunt egale si produsul daca primul e mai mic – acest exercitiu il fac impreuna cu ei pentru a le arata ca un if poate contine un alt if in el

Rescrierea exercitiului folosind **elif**

6. Reluarea functiei update – o sa le arat pe cod cum verificam daca o tasta a fost apasata sau nu

Discutie asupra ce trebuie facut daca pentru fiecare tasta

Implementare initiala – 2 if-uri pentru cand sunt apasate tastele, fara verificari de iesire din fereastra

Copii ar trebui sa observe faptul ca padul iese din ecran daca tin apasate butoanele de

stanga/dreapta – sa adaugam verificari in functie de pozitia padului daca il mai miscam sau nu

7. **Extra(?) Ce este acceleratia?**

S-ar putea ca miscarea padului sa fie destul de plictisitoare pentru ei si atunci le pot atrage atentia ca se poate face mai interesanta

Putem aplica viteza drept concept de variatie si asupra vitezei in sine => acceleratie

Comparatie cu sentimentul pe care il au copii cand accelereaza/decelereaza o masina sau un autobuz

Exemplificare, pt o acceleratie data si  $v=0$ , o sa fac un tabel cu variatia vitezei si cea a distantei in timp

Copii ar trebui sa inteleaga ca totul se reduce la niste simple adunari progresive

Exemplificare pe coordonatele unui punct – tot cu tabel si desen

8. Adaugarea unei variabile pentru a defini acceleratia si viteza=0
9. Adaugarea la cele 2 ifuri pt taste a unei insumari in plus (la un e scadere)  
Explicarea membrilor **left, right, top, bottom** pe un **Rect**  
Conceptul de **global variable**
10. O sa se observe dupa cateva apasari de buton ca padul se misca cu o viteza super mare  
Trebuie resetata viteza cand nu mai e apasat niciun buton

Extra 1: O verificare in plus sa nu mai iasa putin padul din ecran (nu e asa important)

Extra 2: O verificare in plus daca sunt apasate ambele taste stanga dreapta sa nu mai miscam padul de exemplu

## Lectia 6

Desenarea bilei

Etape:

1. **Ce este un modul?**

O sa le explic ca uneori dorim sa separem anumite bucati de cod de alte bucati de cod pentru a le putea refolosi sau pentru un mai bun management

Impreuna cu ei, o sa creem un fisier nou numit ball.py si vom incepe discutia despre ce ne trebuie pentru a desena un cerc.

Vom defini cateva constante pentru coordonatele centrului, raza si culoare

Vom implementa functia de draw cu screen.draw.filled\_circle

**!!! Daca nu au facut pana acum la extra-ul de la lectia 3, se le explic cum se defineste o functie**

2. Importul moduluilui in joc si explicatii
3. Apelul functiei ball.draw in **draw**
4. Vom defini o functie pentru a centra bila pe pad si sa fie deasupra padului  
Voi lua pe flipchart exemplu cu padul si bila si ii voi ghida sa isi dea seama de formule
5. Adaugarea unui bool **ball\_is\_moving** pentru miscarea bilei (optional)  
Explicatii/recapitulare variabile booleene
6. Apelul metodei de centrare in if-urile pentru miscarea padului stanga/dreapta

## Lectia 7

Coliziunea bilei cu peretii si padul

Etape:

1. Definirea a doua variabile pentru miscarea bilei, **delta\_x** si **delta\_y**  
Exemplificarea miscarii bilei pe grafic  
Cele 2 fiind egale se obtine o miscare la 45°  
Dupa implementare ei se pot juca cu aceste valori si pot obtine si alte unghiuri
2. Definirea functiei de **move**
3. **Optional:** definirea functiilor left/right/top/bottom

4. Discutie cand se doreste punerea in miscare a bilei – tasta spatiu  
Cum detecteaza ca o tasta a fost apasata in pgz – metode **on\_key\_down**  
Discutie asupra parametrului functiei  
Implementarea ei prin setarea lui **ball\_is\_moving** pe True
5. Adaugarea implementarii necesare in functia **update**  
Daca bila se misca, apelam functia de a misca bila  
Copii o sa observe ca bila trece prin peretele din dreapta  
Discutie cand vrem sa rikozeze bazat pe coordonate, adica la ce valoare a lui x  
Discutie ce ar trebui sa se intample cand rikozeze- cum **delta\_x** trebuie sa isi schimbe semnul pentru ca directia bilei se inverseaza de la dreapta spre stanga  
Acelasi pattern pentru peretele de sus  
Acelasi pattern pentru peretele din dreapta  
Apoi copii o sa observe ca bila trece prin pad. O sa desenez pe grafic cum dorim sa facem coliziunea si cum vom modifica pozitia bilei  
**!! Implementarea o voi face impreuna cu ei pentru ca trebuie sa le explic metoda **collidepoint****  
Apoi ar trebui sa observe ca bila trece prin peretele de jos si ca atunci dorim o resetare a pozitiei ei si a miscarii ei (optional resetarea delta-urilor cu modul sau hardcodat)

## Lectia 8

Coliziunea bilei cu brickurile

Etape:

1. Discutie asupra unui brick vazut ca dreptunghi  
Recapitulare coordonate colturi, centru si laturi  
Discutie asupra bilei vazuta ca patrat  
Desenarea cazurilor in care se poate ciocni bila de un brick  
4 cazuri: sus, jos, stanga si dreapta  
Definirea celor 4 puncte pentru bila  
Discutie asupra celor 2/4 cazuri de coliziune cu desenarea cazurilor si prezentarea punctelor – ce se intampla cu brickul si schimbarea directiei bilei  
Mentionarea tipului **None**  
Pana acum totul e pe flipchart
2. Implementarea functiei **check\_bricks\_collision**  
O sa implementez deodata cu ei  
Parcurgerea matricei de brickuri  
Definirea celor 4 puncte de pe bila  
Scrierea cazurilor  
Fara iful in care verificam daca un brick e **None** o sa le crape programul atat in check, cat si in **draw!!**  
Explicatii asupra tipului **None**

## Lectia 9

Sistemul de viet

Etape:

1. Definirea unei variabile pentru a tine minte cate vieti avem
2. Afisarea numarului de vieti la o pozitie hardcodata sau cu o variabila  
Explicarea pe scurt a stringurilor
3. Discutie cand vrem sa scada numarul de vieti si ce se intampla la 0  
Implementarea scaderii  
Implementarea if-ului pentru 0 vieti => resetarea brickurilor si (optional) a culorilor  
!! Daca nu au facut-o deja, generarea brickurilor trebuie facuta functie