**Aplicatie in limbajul de programare**

**Python: „Snake Game”**

**Realizat de Rosu Andrei**

In timpul acestui curs de programare python am invatat multe concepte de baza. Acestea sunt: citirea,afisarea unor valori,liste,tuple,functii, proiectarea orientata pe obiecte,baza de date etc.

Aici am realizat o aplicatie importanta in python:”Snake-game”. Acesta este un joc in care sarpele consuma „bila”.Astfel creste scorul,acumuleaza puncte,sarpele isi mareste lungimea. In cazul in care sarpele se loveste de peretii chenarului sau de el insusi se opreste jocul.

In primul rand, am folosit „import from tkinter” ,la fel si modulul random. Apoi am definit diferite functii si clase. Am creat clasele Snake si Food. De asemenea, am rea-lizat si functiile next\_turn, change\_direction, check\_collisions si game\_over.

Inainte de acestea, am pus niste constante.Constantele sunt variabile utilizate in mai multe setari. Ele sunt scrise cu litere mari: GAME\_WIDTH latimea de 700, GAME\_HEIGHT inaltimea de 700, SPEED viteza de 100, SPACE\_SIZE marimea spa tiului 50, BODY\_PARTS partile corpupui 3,SNAKE\_COLOR culoarea sarpelui verde valoarea in hexazecimal #00FF00, FOOD\_COLOR culoarea mancarii in rosu #FF0000 , BACK-GROUND\_COLOR culoarea de fundal in negru #000000. Astfel sar pele e verde,mancarea in rosu, iar fundalul e in negru.

In al doilea rand, am creat o fereastra pe care am numit-o window si am folosit functiile Tk() si mainloop. Si functia title a fost folosita pentru a defini titlul:”Snake ga-me”. Pentru ca fereastra sa nu fie redimensionata ,am utilizat window.resizable (Fal-se , False). Pentru a defini scorul , i-am dat o variabila score=0, si o directie initiala. Mai tarziu, am realizat o eticheta, utilizand functia Label() la fereastra, in care avem textul cu scorul , formatul si fontul.

In al treilea rand, se adauga functia Canvas pentru a seta fundalul jocului con-stant: latimea si inaltimea chenarului.La actualizarea ferestrei se foloseste window. update(). Apoi, pentru a putea gasi anumite dimensiuni , am implementat functiile window.winfo\_width(),winfo\_screenheight() etc. De asemenea, pentru a vedea mai mult, am reglat pozitia ferestrei si am definit x si y.La geometria ferestrei se face cu window.geometry() . In interior sunt dimensiunile.

In al patrulea rand, s-au definit obiectele snake si food. Acestea folosesc clasele Snake,respectiv Food. In clasa Food , am creat metoda init , am definit functia init . Pentru a pune obiectul food la intamplare am folosit x=random.randint(). Sunt 14 posibilitati: de la 0 la 14. Acelasi lucru e facut si pentru y. La setarea coordonatelor se practica self.coordinates ce cuprinde o lista formata din x si y. La realizarea mancarii/ bilei am folosit canvas.create\_oval(). Astfel, la fiecare rulare se schimba pozitia bilei.

In clasa Snake, la fel ca la Food, am pus functia def init. Aici am setat marimea sarpelui,o lista a coordonatelor . De asemenea, cele doua bucle repetitive for sunt pentru coordonatele sarpelui,respectiv sa aiba o forma rectangulara.

In functia next\_turn(), avem doi parametrii snake si food. Se foloseste snake. coordinates[0] pentru a stoca coordonatele in x si y. x este setat pentru stanga sau dreapta, iar y pentru sus sau jos. Cand sarpele atinge cercul,scorul creste cu o va-loare,sarpele isi mareste lungimea.Bila dispare prin canvas.delete si apare intr-o noua pozitie. Pentru randul urmator am pus window.after(). De asemenea, pentru parametrul snake am inserat un nou set de coordonate, iar pentru a crea un nou grafic am utilizat canvas.create\_rectangle(). La final am apelat functia next\_turn.

In al cincilea rand, pentru controlul sarpelui se foloseste window.bind().Astfel, el isi schimba directia in sus,in jos, la stanga, la dreapta. Apoi,in functia change\_directi-on am utilizat global direction pentru a putea accesa directia. Astfel, putem sa schim-bam directia sarpelui ori de cate ori vrem.

Functia check\_collisions() este pentru a detecta coliziunile. Are un parametru in paranteza:snake. Pentru x si y am setat coordonatele astfel incat sa fie sarpele doar in parametrii chenarului. Daca sarpele atinge peretii chenarului,se va afisa mesajul: „Game over”. Apoi, ultima functie implementata game\_over() este pentru finalul jocu-lui.Astfel,cand sarpele atinge peretii chenarului sau pe el insusi,se sterge tot prin canvas.delete(ALL),si apare in chenar mesajul „GAME OVER” prin canvas.create \_ text().

In concluzie, la final se apeleaza clasele si functiile se apeleaza. Apoi se apasa pe butonul de run si apare chenarul cu jocul. Astfel acesta este un exemplu de dez-voltare de jocuri in python.