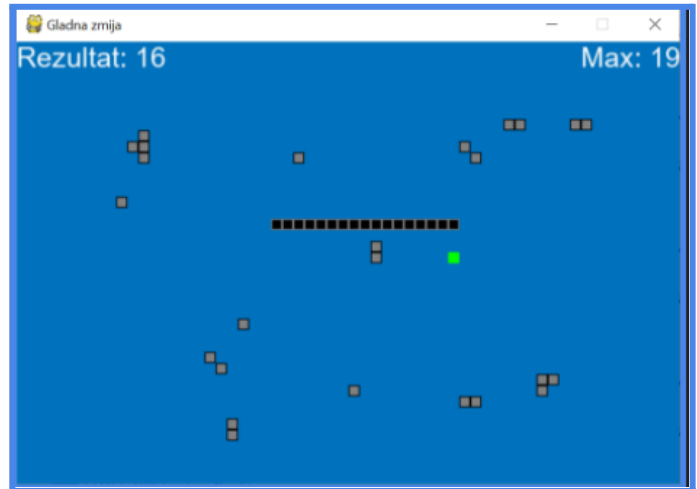

Igrica 3

OBJASNJENJE IGRICE

Osnovni play game:

- kretanje zmije u ogranicenom prostoru pygame prozora
- identifikacija kraja igre kada udari u sebe ili u ivicu zida
- cuvanje trenutnog rezultata
- cuvanje maksimalnog rezultata ili highscorea u tekstualnom fajlu sa strane
- random definisanje hrane koju zmija jede
- kretanje uz tastere tastature po zamisljenoj mrezi kvadratica
- zmija raste sto vise hrane pojede
- Uporediti ovaj gameplay sa igricom koju su ucenici igrali i prodiskutovati slicnosti i razlike i ogranicenja u implementacija sa kojima se mozemo susresti



Otezavanje igrice i fan momenat:

- 1) Dodavanje prepreka u vidu kockica koje ako zmija dodirne gubi zivot
- 2) Generisanje hrane uz ivicu prozora
- 3) Ubrzanje zmije kako puno jede hrane tj kako vreme u igrici odmice
- 4) Bilo sta sto ucenici smisle

PLAN KODIRANJA IGRICE

→ Deo 1

- ◆ Podesavanje prozora
- ◆ Boje i fontovi i funkcije za to
- ◆ Iscrtavanje zmije i hrane

→ Deo 2

- ◆ Kretanje zmije
- ◆ Detekcija inputa sa tastature

→ Deo 3

- ◆ Detekcija dodira
- ◆ Azuriranje rezultata
- ◆ Ogranicenja kretanja

→ Deo 4

- ◆ Dodavanje tela zmije

- ◆ Prepreke
- ◆ Poeni
- ◆ Highscore

DEO 1

KORAK 1: Podesavanje prozora

```
import pygame
import pygame.bg

#definisanje boja koje ce nam biti potrebne za ovu igricu
plava = (0, 113, 188)

#velicina prozora
prozor_s = 600
prozor_v = 400

#setapujemo prozor
prozor = pygame.bg.open_window(prozor_s, prozor_v, "Igra zmija")
prozor.fill(plava)

pygame.bg.wait_loop()
```

KORAK 2: Dodavanje dodatnih boja, fontova i pomocne funkcije za to

```
#definisanje boja koje ce nam biti potrebne za ovu igricu
crna = (0, 0, 0)
crvena = (241, 90, 36)
zelena = (0, 255, 0)
plava = (0, 113, 188)

...

#setapujemo prozor
prozor = pygame.bg.open_window(prozor_s, prozor_v, "Igra zmija")
prozor.fill(plava)

#implementacija fontova
font_poruka = pygame.font.SysFont("arial", 25)
font_rezultat = pygame.font.SysFont("arial", 35)
font_max_rezultat = pygame.font.SysFont("arial", 35)

#prikaz poruke na kraju
def poruka_na_kraju(tekst):
    text = font_poruka.render(tekst, True, crvena)
    prozor.blit(text, (prozor_s // 6, prozor_v // 3))
```

```

#prikaz rezultata u levom cosku
def ispisi_rezultat(score):
    rezultat = font_rezultat.render("Rezultat: {}".format(score), True,
    crvena)
    print(rezultat.get_height())
    prozor.blit(rezultat, (0, 0))

#prikaz max_rezultata u desnom cosku
def ispisi_max_rezultat():
    rez = procitajIzFajla()
    rezultat = font_rezultat.render("HighScore: {}".format(rez), True,
    crvena)
    print(rezultat.get_height())
    prozor.blit(rezultat, (prozor_s - rezultat.get_width(), 0))

def procitajIzFajla():
    fajl = open("highscore.txt", "r") # w (write), r (read)
    best_score = fajl.readline()
    fajl.close()
    return best_score

```

OBJASNJENJE:

NAPOMENA 1: Citanje iz fajla za high score - objasniti koje su druge opcije za rad sa fajlovima

NAPOMENA 2: Objasniti vrednost ovako organizovanog koda, u promenljive i u zasebne funkcije

KORAK 3: Dodavanje funkcionalnosti za iscrtavanje zmije i hrane

```

# globalna promenljiva za definisanje mreze na prozoru
zmija_sirina_px = 10

#funkcija kojom cemo iscrtavati hranu na random mestu
def nacrtaj_hranu(_x, _y):
    pygame.draw.rect(prozor, zelena, (_x, _y, zmija_sirina_px,
    zmija_sirina_px))

#random pozicija za hranu
def random_koordinate():
    _x = random.randint(0, prozor_s-1)//10*10
    _y = random.randint(0, prozor_v-1)//10*10 - zmija_sirina_px*5
    return (_x, _y)

#crtamo zmiju
def nacrtaj_zmiju(x1, y1):

```

```
pygame.draw.rect(prozor, crna, (int(x1), int(y1), zmija_sirina_px,
zmija_sirina_px))
```

OBJASNJENJA:

NAPOMENA 1: Objasniti trik da random uvek vrati broj deljiv sa 10

NAPOMENA 2: Objasniti da ne treba ukljuciti zadnji broj u rangeu za dimenzije prozora pri generisanju randoma jer postoji sansa da ce se kockica generisati na spoljasnjoj ivici prozora

NAPOMENA 3: Objasniti za iscrtavanje zmije da cemo je promeniti kad napravimo da se telo zaista uvecava

NAPOMENA 4: Dosta vremena utrositi na objasnjavaње ove mreže za random koordinate

KORAK 4: Glavni deo programa i testiranje svih ovih funkcija

```
#main game loop
def gameLoop():

    #pocetne koordinate zmije
    x1 = prozor_s // 2
    y1 = prozor_v // 2

    zmija_duzina = 1

    #random position for next thing to eat
    (hrana_x, hrana_y) = random_koordinate()
    nacrtaj_hranu(hrana_x, hrana_y)
    nacrtaj_zmiju(x1, y1)
    ispisi_rezultat(zmija_duzina - 1)
    ispisi_max_rezultat()

gameLoop()
pygamebg.wait_loop()
```

OBJASNJENJE:

NAPOMENA 1: U glavnoj petlji samo naredati napisane funkcije radi testiranja bez nekog pravila

DEO 2

KORAK 1: dogadjaji tastature i update koordinata

```
#main game loop
def gameLoop():

    kraj_igre = False
    #pocetne koordinate zmije
    x1 = prozor_s // 2
```

```

y1 = prozor_v // 2
x1_dif = 0
y1_dif = 0

zmija_duzina = 1

#random position for next thing to eat
(hrana_x, hrana_y) = random_koordinate()

while not kraj_igre:

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            kraj_igre = True
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_LEFT:
                x1_dif = -zmija_sirina_px
                y1_dif = 0
            elif event.key == pygame.K_RIGHT:
                x1_dif = zmija_sirina_px
                y1_dif = 0
            elif event.key == pygame.K_UP:
                y1_dif = -zmija_sirina_px
                x1_dif = 0
            elif event.key == pygame.K_DOWN:
                y1_dif = zmija_sirina_px
                x1_dif = 0

    # update coordinates, unutar while petlje
    x1 += x1_dif
    y1 += y1_dif
    # sve ostale funkcije pomeriti unutar while petlje
    nacrtaj_hranu(hrana_x, hrana_y)
    nacrtaj_zmiju(zmija_sirina_px, x1, y1)
    ispisi_rezultat(zmija_duzina - 1)
    ispisi_max_rezultat(0)

```

KORAK 2: animacija kretanja i sat

#dodamo promenljive

```

zmija_sirina_px = 10
sat = pygame.time.Clock()
zmija_brzina = 15

```

..

dodamo unutar while poziv updatea i otkucaj sata

```

ispisi_rezultat(zmija_duzina - 1)

```

```
ispisi_max_rezultat(0)
pygame.display.update()
sat.tick(zmija_brzina)
```

KORAK 3: redosled iscrtavanja nakon implementacije sata, mora da bude i fill tu

```
# update coordinates
x1 += x1_dif
y1 += y1_dif

prozor.fill(plava)
nacrtaj_hranu(hrana_x, hrana_y)
nacrtaj_zmiju(zmija_sirina_px, x1, y1)
ispisi_rezultat(1)
ispisi_max_rezultat()

pygame.display.update()
sat.tick(zmija_brzina)
```

KORAK 4: Obrada kraja igre i udarac u zid prozora

```
elif event.key == pygame.K_DOWN:
    y1_dif = zmija_sirina_px
    x1_dif = 0

# Ako se dodirne ivica prozora zmijom onda je kraj igre
if x1 >= prozor_s or x1 < 0 or y1 >= prozor_v or y1 < 0:
    kraj_igre = True

# update coordinates
x1 += x1_dif
y1 += y1_dif

prozor.fill(plava)
nacrtaj_hranu(hrana_x, hrana_y)
nacrtaj_zmiju(zmija_sirina_px, x1, y1)
ispisi_rezultat(1)
ispisi_max_rezultat()

pygame.display.update()
sat.tick(zmija_brzina)

prozor.fill(plava)
poruka_na_kraju("Izgubio si")
pygamebg.wait_loop()

gameLoop()
```

~~pygamebg.wait_loop()~~

OBJASNJENJE SVIH RESENJA:

NAPOMENA 1: Objasnjenje za petlju while sa boolom i opcije za nastanak beskonacne petlje

NAPOMENA 2: Objasniti kako rade eventi sa tastaturom, paralela sa Scratch eventima

NAPOMENA 3: Bitno razumeti kako se koordinate menjaju po strelicama da bi se sacavala mreza i kretanje po 10 piksela

NAPOMENA 4: Bitno razumeti iscrtavanje u jednom ticku i jednoj iteraciji while petlje, sta se sve mora iscrtati, sta se updatuje a sta ne

DEO 3

KORAK 1: detekcija dodira izmedju zmiije i hrane prilikom kretanja zmiije

```
x1 += x1_dif
y1 += y1_dif
# sve ostale funkcije pomeriti unutar while petlje
```

```
if x1 == hrana_x and y1 == hrana_y:
    (hrana_x, hrana_y) = random_koordinate()
```

```
prozor.fill(plava)
nacrtaj_hranu(hrana_x, hrana_y)
```

OBJASNJENJE:

- Ne postoji funkcija za dodir izmedju dva drawn objekta (moze ako su sprites)
- Kada to nemamo onda mora preko koordinata
- Ovde je lakse nego kod skupljanja loptica, ovde su dva kvadrata istih dimenzija
- Pustiti da ucenik sam dodje do ovog resenja

KORAK 2: brojanje poena, tj duzine zmiije

```
x1 += x1_dif
y1 += y1_dif
# sve ostale funkcije pomeriti unutar while petlje
```

```
if x1 == hrana_x and y1 == hrana_y:
    (hrana_x, hrana_y) = random_koordinate()
    zmija_duzina += 1
```

```
prozor.fill(plava)
nacrtaj_hranu(hrana_x, hrana_y)
```

OBJASNJENJE:

- Zmija duzina ce se koristiti kao promenljiva i za momenat iscrtavanja zmije
- Iscrtavanje zmije ce se uraditi kasnije

KORAK 3: upis novog high scorea

```
def upisiUFajl(rezultat):  
    f = open("highscore.txt", "w")  
    f.write(str(rezultat))  
    f.close()
```

OBJASNJENJE:

- Funkcija write mora da primi string kao parametar

KORAK 4: poziv te funkcije nakon završene while petlje

```
prozor.fill(plava)  
poruka_na_kraju("Izgubio si")  
if (zmija_duzina-1) > int(procitajIzFajla()):  
    upisiUFajl(zmija_duzina-1)  
pygamebg.wait_loop()
```

OBJASNJENJE:

- Paziti da uporedjivanje bude po intu, jer funkcija vraca str. Ostaviti da bude greska da bi ucenik skapirao da treba da se koristi int() funkcija
- Ucenik treba sam da detektuje gde treba dodati ovaj if
- Ucenik treba sam da napravi uslov

KORAK 5: Kreiranje hrane uz ivicu prozora kako bi igra bila teza

```
def random_koordinate():  
    # zelimo na 100 piksela oko ivice prozora da se generise hrana  
    randX = random.randint(0, 1)  
    _x = randX*random.randint(0, 100)//10*10 if randX == 0 else  
random.randint(prozor_s-100, prozor_s-1)//10*10  
    randY = random.randint(0, 1)  
    _y = randY*random.randint(0, 100)//10*10 if randY == 0 else  
random.randint(prozor_v-100, prozor_v-1)//10*10 - zmija_sirina_px*5  
    print(_x, _y) # testiranje  
    return (_x, _y)
```

OBJASNJENJE:

- Ucenik da smisli algoritam kako da se bira random mesto po ivici od 100 piksela sa bilo koje strane prozora (sem odozgo, gde je isto 100 piksela ali ispod slova highscora)
- U algoritmu iznad bira se random broj 0 ili 1 i u odnosu na to se bira da li po horizontali ili vertikali se uzima bliza ili dalja ivica
- Navesti ucenika na takav algoritam
- Koristen drugi kraci zapis ifa, objasniti (nesto kao inline grananje, kao ternarni operator)

KORAK 6: Telo zmije - dodavanje promenljive koja je niz

```
zmija_sirina_px = 10
sat = pygame.time.Clock()
zmija_brzina = 15
zmija_telo = []
```

OBJASNJENJE:

- Objasniti da se u listi mora pamtiti svaka adresa zmije kako bi se pri dodavanju novog dela repa znalo gde da se doda

KORAK 7: Telo zmije - azuriranje novog polja na kome je zmija bila

```
if x1 == hrana_x and y1 == hrana_y:
    (hrana_x, hrana_y) = random_koordinate()
    zmija_duzina += 1

#zmija:
zmija_glava = []
zmija_glava.append(x1)
zmija_glava.append(y1) # zmija_glava = [x1, y1]
zmija_telo.append(zmija_glava)

prozor.fill(plava)
nacrtaj_hranu(hrana_x, hrana_y)
```

OBJASNJENJE:

- Objasniti da se u pamti jedno polje kao lista od x i y koordinate
- Jedno polje je element velike liste
- Propitati ucenike u komentarima da li razumeju indeksiranje takve strukture
- Zmija[[0,0], [10,0], [10,10],[10,20]], koji je indeks broja 20, itd...
- Kako se appenduje element u ovakvu strukturu

KORAK 8: Telo zmije - iscrtavanje - editovanje funkcije za iscrtavanje

```
#crtamo zmiju
def nacrtaj_zmiju(zmija): # zmija je lista
    for element in zmija:
        pygame.draw.rect(prozor, crna, (int(element[0]),
int(element[1]), zmija_sirina_px, zmija_sirina_px))
```

OBJASNJENJE:

- Pretraga liste po objektu, a ne po indeksima objekata

KORAK 9: Poziv te funkcije na pravi nacin za iscrtavanje tela zmije

```
prozor.fill(plava)
nacrtaj_hranu(hrana_x, hrana_y)
nacrtaj_zmiju(zmija_telo)
ispisi_rezultat(zmija_duzina - 1)
ispisi_max_rezultat()
pygame.display.update()
sat.tick(zmija_brzina)
```

KORAK 10: Testirati. Sta ne valja, zasto je telo zmije tako veliko? Propitati ucenike. (jer je telo lista kvadrata na adresama u prozoru, i puni se svakim poljem koje zmija dodirne. Lista se nikada ne prazni, a trebalo bi i to da se izbacuje najstariji element)

```
#zmija:
zmija_glava = []
zmija_glava.append(x1)
zmija_glava.append(y1) # zmija_glava = [x1, y1]
zmija_telo.append(zmija_glava)

if len(zmija_telo) > zmija_duzina:
    del zmija_telo[0]

prozor.fill(plava)
nacrtaj_hranu(hrana_x, hrana_y)
```