

PROGRAMIRANJE U INTEGRISANIM TEHNOLOGIJAMA

Oznaka predmeta: PIT

Predavanje broj: 09

Nastavna jedinica: pygame,

Nastavne teme:

Uvod: konzolna igra HANGMAN. Modul pygame za kreiranje video igara: FSM, boje, koordinate, primitivi, događaji, muzika, zvučni efekti, kontrole. 2D Igra Asteroid invasion.

Predavač: prof. dr Perica S. Štrbac, dipl. ing.

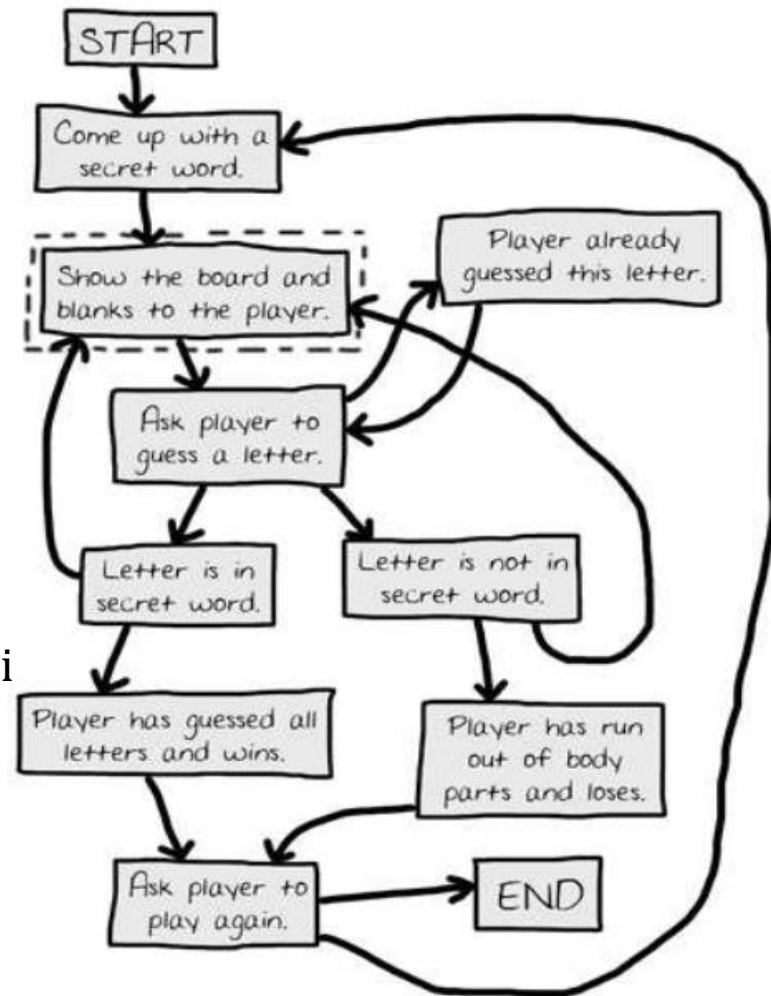
Literatura:

Al Sweigart: "Making Games with Python & Pygame", e-book, 2012.

Al Sweigart: "Invent Your Own Computer Games with Python", e-book, 3rd Edition, 2015.

Python: igra u konzolnom modu

- Daje se primer jednostavne igre u konzolnom modu bez korišćenja modula pygame.
- Razmatra se dijagram toka igre HUNGMAN:
 - daje se tekstualni prikaz vešalice i tačkicama prikrivena slova zagonetne reči
 - pogađaju se slovo koga možda sadrži zagonetna reč
 - svaka greška docrtava „čiča Glišu“ dok ne bude „obešen“.
 - pogođeno slovo se pojavljuje umesto tačkica na pozicijima u zagonetnoj reči na kojima se to slovo nalazi.



Python: igra u konzolnom modu

```
import random
HANGMANPICS = [ '''
+---+
|   |
|   |
|   |
===== ''', '''
+---+
|   |
|   |
|   |
===== ''', '''
+---+
|   |
0   |
|   |
|   |
===== ''']]
```

Python: igra u konzolnom modu

```
+----+
|
0
/|
|
|
|
===== ' ', ' '
+----+
|
0
/|\
|
|
|
===== ' ', ' '
+----+
|
0
/|\
/
|
|
|
===== ' ', ' '
```

Python: igra u konzolnom modu

```
+---+
|
o
/|\
/|\
|
=====
```

```
words = 'abcdefgh'.split() #dodajte po zelji zagonetne reci
```

```
def getRandomWord(wordList):
    # odaberi zagonetnu rec
    wordIndex = random.randint(0, len(wordList) - 1)
    return wordList[wordIndex]

def displayBoard(HANGMANPICS, missedLetters, correctLetters, secretWord):
    print(HANGMANPICS[len(missedLetters)])
    print()
    print('Missed letters:', end=' ')
    for letter in missedLetters:
        print(letter, end=' ')
    print()

    blanks = '_' * len(secretWord)
```

Python: igra u konzolnom modu

```
for i in range(len(secretWord)): # ubaci pogodjena slova
    if secretWord[i] in correctLetters:
        blanks = blanks[:i] + secretWord[i] + blanks[i+1:]

for letter in blanks: # prikazi stanje pogodaka
    print(letter, end=' ')
print()

def getGuess(alreadyGuessed):
    # unos slova te proveriti: da li je uneto jedno slovo, te
    # da li je to slovo vec bilo uneto
    while True:
        print('Guess a letter.')
        guess = input()
        guess = guess.lower()
        if len(guess) != 1:
            print('Please enter a single letter.')
        elif guess in alreadyGuessed:
            print('You have already guessed that letter. Choose again.')
        elif guess not in 'abcdefghijklmnopqrstuvwxyz':
            print('Please enter a LETTER.')
        else:
            return guess
```

Predavanje br. 9

Python: igra u konzolnom modu

```
def playAgain():  
    # pitalica za igranje nove igre  
    print('Do you want to play again? (yes or no)')  
    return input().lower().startswith('y')  
  
print('H A N G M A N')  
missedLetters = ''  
correctLetters = ''  
secretWord = getRandomWord(words)  
gameIsDone = False  
  
while True:  
    #prikaz stanja igre  
    displayBoard(HANGMANPICS, missedLetters, correctLetters, secretWord)  
    #biraj slovo  
    guess = getGuess(missedLetters + correctLetters)  
  
    if guess in secretWord:  
        # slovo je sadržano u zagonetnoj reci  
        correctLetters = correctLetters + guess  
        # pretpostaviti da je sve pogodjeno a onda proveriti  
        foundAllLetters = True
```

Python: igra u konzolnom modu

```
for i in range(len(secretWord)):
    if secretWord[i] not in correctLetters:
        foundAllLetters = False
        break
if foundAllLetters:
    print('Yes! The secret word is "' + secretWord + '"! You
have won!')
    gameIsDone = True
else:
    # slovo nije sadržano u zagonetnoj reci, promasaj je
    missedLetters = missedLetters + guess
    # proverava da li je igrac pogadjao max dozvoljeni broj puta
    if len(missedLetters) == len(HANGMANPICS) - 1:
        displayBoard(HANGMANPICS, missedLetters, correctLetters,
secretWord)
        print('You have run out of guesses!\nAfter ' +
              str(len(missedLetters)) + ' missed guesses and ' +
str(len(correctLetters)) +
              ' correct guesses, the word was "' + secretWord +
              '"')
        # promena stanja igre
        gameIsDone = True
```


Python: igra u konzolnom modu

ako je stanje igre 'zavrsena' ponuditi pitalicu za novu igru

```
if gameIsDone:
    if playAgain():
        missedLetters = ''
        correctLetters = ''
        gameIsDone = False
        secretWord = getRandomWord(words)
    else:
        break
```

- Primer izlaza:

```
+----+
|    |
0    |
    |
    |
    |
=====
```

Missed letters: k
_ _ c _ _ _ _
Guess a letter.

```
+----+
|    |
0    |
/|\  |
/    |
=====
```

Missed letters: k l j t o
a b c d e f g _
Guess a letter.
h
Yes! The secret word is "abcdefgh"! You have won!
Do you want to play again? (yes or no)

Python: pygame

- Instalacija modula pygame:

```
pip install pygame
```

```
provera: import pygame
```

- Kreiranje pygame prozora

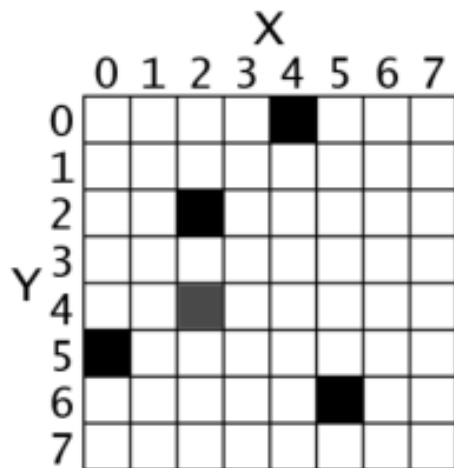
```
import pygame, sys
from pygame.locals import *

pygame.init()
DISPLAYSURF = pygame.display.set_mode((400, 300))
pygame.display.set_caption('Hello World!')
while True: # main game loop
    for event in pygame.event.get():
        if event.type == QUIT:
            pygame.quit()
            sys.exit()
    pygame.display.update()
```

Pozvana je inicijalizacija pygame-a, postavljena je veličina ekrana (400x300), naslov prozora 'Hello World!', aktivirana neprekidna petlja ispitivanja događaja, reagovanje na događaj **QUIT** te ažuriranje ekrana.

Python: pygame: koordinate, boje

- Ekranske koordinate su određene kao na slici:



- U tabeli pored, prikazan je naziv boje koju odgovara vrednosti parametara RGB komponenti.
Prikazane vrednosti komponente boje su u opsegu [0,255].
- Dodatno boji je pridružen alfa kanal, neprovidnost, tako da se dobija RGBA vrednost pixel-a.
A=255 je neprovidan
A=0 je providan

Color	RGB Values
Aqua	(0, 255, 255)
Black	(0, 0, 0)
Blue	(0, 0, 255)
Fuchsia	(255, 0, 255)
Gray	(128, 128, 128)
Green	(0, 128, 0)
Lime	(0, 255, 0)
Maroon	(128, 0, 0)
Navy Blue	(0, 0, 128)
Olive	(128, 128, 0)
Purple	(128, 0, 128)
Red	(255, 0, 0)
Silver	(192, 192, 192)
Teal	(0, 128, 128)
White	(255, 255, 255)
Yellow	(255, 255, 0)

Python: pygame: crtanje

- Primer programa koji crta neke osnovne geometrijske primitive:

```
import pygame, sys
from pygame.locals import *
pygame.init()
# set up the window
DISPLAYSURF = pygame.display.set_mode((500, 400), 0, 32)
pygame.display.set_caption('Drawing')
# set up the colors
BLACK = ( 0, 0, 0)
WHITE = (255, 255, 255)
RED = (255, 0, 0)
GREEN = ( 0, 255, 0)
BLUE = ( 0, 0, 255)
# draw on the surface object
DISPLAYSURF.fill(WHITE)
pygame.draw.polygon(DISPLAYSURF, GREEN, ((146, 0), (291, 106), (236,
277), (56, 277), (0, 106))) # debljina bi znacila konturu
pygame.draw.line (DISPLAYSURF, BLUE, ( 60, 60), (120, 60), 4)
pygame.draw.line (DISPLAYSURF, BLUE, (120, 60), ( 60, 120) )
pygame.draw.line (DISPLAYSURF, BLUE, ( 60, 120), (120, 120), 4)
pygame.draw.circle(DISPLAYSURF, BLUE, (300, 50), 20, 0) # krug
```

Python: pygame: crtanje

```
pygame.draw.ellipse(DISPLAYSURF, RED, (300, 250, 40, 80), 1)
pygame.draw.rect(DISPLAYSURF, RED, (200, 150, 100, 50))
```

za brzi rad sa pojedinacnim pikselima

```
pixObj = pygame.PixelArray(DISPLAYSURF)
```

```
pixObj[480][380] = BLACK
```

```
pixObj[482][382] = BLACK
```

```
pixObj[484][384] = BLACK
```

```
pixObj[486][386] = BLACK
```

```
pixObj[488][388] = BLACK
```

```
del pixObj # kraj postavljanja pojedinacnih piksela
```

Surface.get_at((x, y)): return Color

run the game loop

```
while True:
```

```
    for event in pygame.event.get():
```

```
        if event.type == QUIT:
```

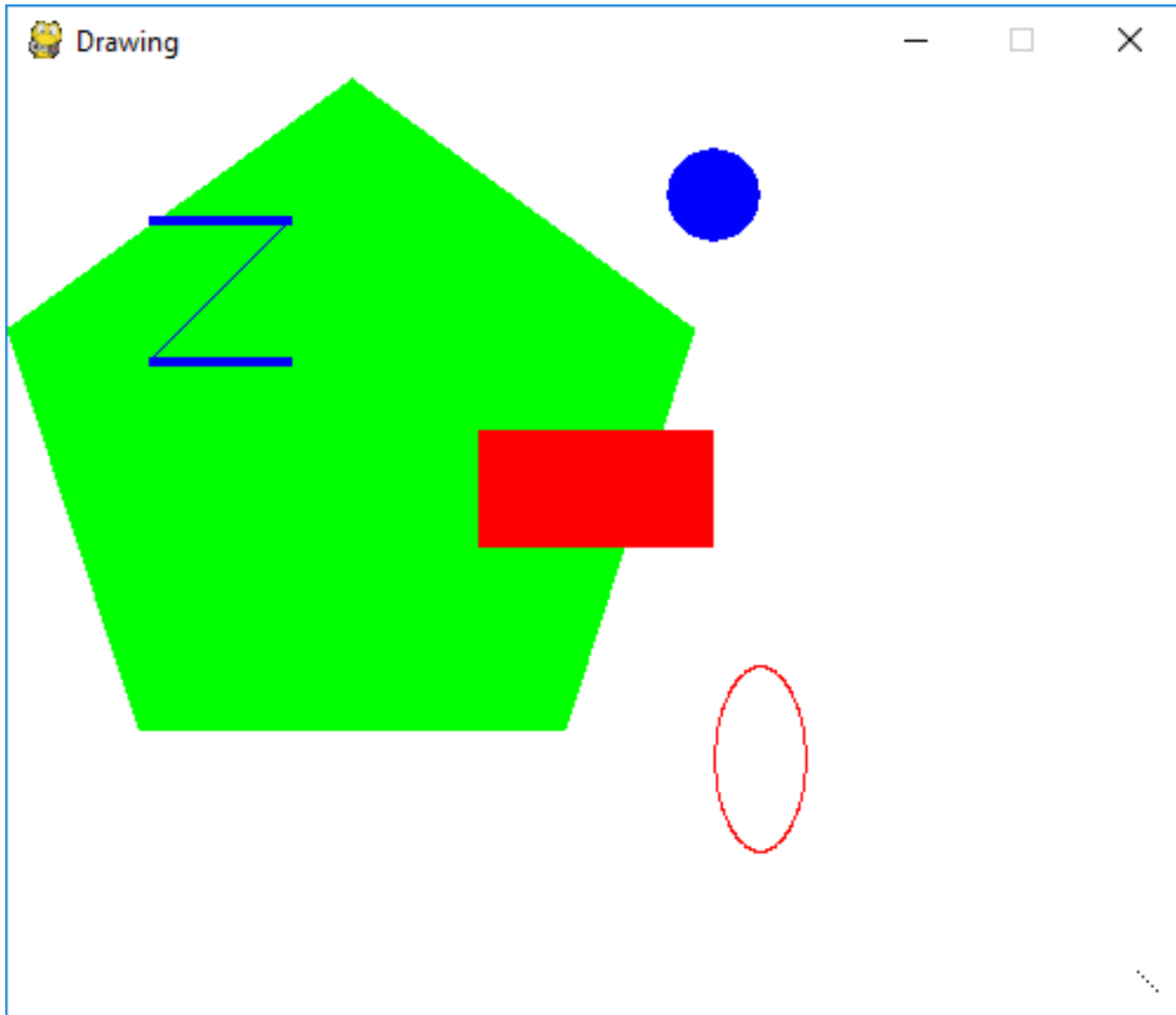
```
            pygame.quit()
```

```
            sys.exit()
```

```
    pygame.display.update()
```

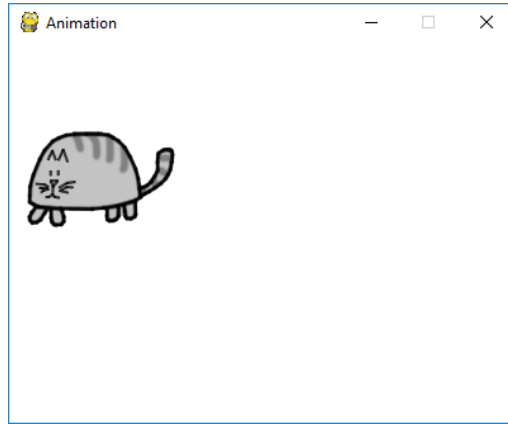
Python: pygame: crtanje

- Izlaz:



Python: pygame: baratanje slikom

- Primer: pomeranje png slike mačke po ivicama ekrana, uz kontrolu brzine prikaza.



```
import pygame, sys
from pygame.locals import *
pygame.init()
FPS = 60 # frames per second setting
fpsClock = pygame.time.Clock()
# set up the window
DISPLAYSURF = pygame.display.set_mode((400, 300), 0, 32)
pygame.display.set_caption('Animation')
WHITE = (255, 255, 255)
catImg = pygame.image.load('cat.png')
catx = 10
caty = 10
direction = 'right'
```

Python: pygame: baratanje slikom

```
while True: # the main game loop
    DISPLAYSURF.fill(WHITE)
    if direction == 'right':
        catx += 5
        if catx == 280: direction = 'down'
    elif direction == 'down':
        caty += 5
        if caty == 220: direction = 'left'
    elif direction == 'left':
        catx -= 5
        if catx == 10: direction = 'up'
    elif direction == 'up':
        caty -= 5
        if caty == 10: direction = 'right'
    DISPLAYSURF.blit(catImg, (catx, caty)) # GL ugao slike na catx,caty
    for event in pygame.event.get():
        if event.type == QUIT:
            pygame.quit()
            sys.exit()

    pygame.display.update()
    fpsClock.tick(FPS)
```


Python: pygame: font

- Postavljanje teksta na ekranu

```
import pygame, sys
from pygame.locals import *
pygame.init()
DISPLAYSURF = pygame.display.set_mode((400, 300))
pygame.display.set_caption('Hello World!')
WHITE = (255, 255, 255)
GREEN = (0, 255, 0)
BLUE = (0, 0, 128)
fontObj = pygame.font.Font('freesansbold.ttf', 32)
textSurfaceObj = fontObj.render('Hello world!', True, GREEN, BLUE)
# anti-aliasing = True

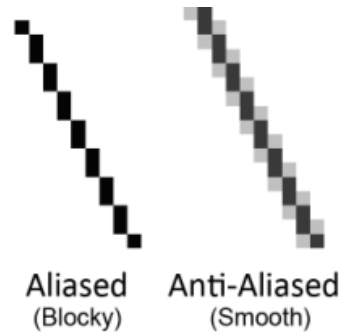
textRectObj = textSurfaceObj.get_rect()
textRectObj.center = (200, 150)

while True: # main game loop
    DISPLAYSURF.fill(WHITE)
    DISPLAYSURF.blit(textSurfaceObj, textRectObj)
    for event in pygame.event.get():
        if event.type == QUIT:
            pygame.quit()
            sys.exit()
    pygame.display.update()
```



Python: pygame: anti-aliasing, sound

- Anti-aliasing crtanje omogućuje da kosa linije „ne bude“ nazubljena.



- Metode koje se koriste za anti-aliasing su oblika:

```
pygame.draw.aaline()  
pygame.draw.aalines()
```

sa istim parametrima kao što je bilo za `line()` i `lines()`.

- Zvučni efekti se postižu korišćenjem objekta `pygame.mixer.Sound()`, koji podržava formate: WAV, MP3 i OGG.
 - Koriste se metode `play()` i `stop()` za pokretanje i zaustavljanje zvuka.

```
soundObj = pygame.mixer.Sound('beeps.wav')  
soundObj.play()  
import time  
time.sleep(1)  
soundObj.stop()
```

Python: pygame: anti-aliasing, sound

- Za pozadinsku muzike koristi se:
`pygame.mixer.music.load('pozadinska_muzika.mp3')`
- Pygame može učitati samo jedan muzički fajl za pozadinsku muziku u jednom trenutku.
- Podržani formati su:
 - WAV,
 - MP3,
 - MIDI.
- Puštanje pozadinske muzike se postiže sa:
`pygame.mixer.music.play(-1, 0.0)`
 - prvi parametar -1 znači da muzika svira neprekidno, a ako je ovaj parametar veći onda određuje koliko puta će se uzastopno puštati pozadinska muzika.
 - drugi parametar znači od koje sekunde, gledano od početka, počinje puštanje pozadinske muzike.
- Za trenutno zaustavljanje pozadinske muzike pozvati:
`pygame.mixer.music.stop()`

Python: pygame, događaji

- Obrada događaja koji potiču sa tastature, miša:

```
for event in pygame.event.get(): # uzmi sve dogadjaje koji se desise
    if (event.type==KEYUP and event.key==K_ESCAPE):
        pygame.quit()
        sys.exit()
    elif event.type == MOUSEMOTION:
        mousex, mousey = event.pos
    elif event.type == MOUSEBUTTONUP: # and event.button==LEFT ili ==1
        mousex, mousey = event.pos
        mouseClicked = True
```

- U prethodnom primeru proverava se da li je:
 - na tastaturi kliknuto na taster Esc pri čemu se izlazi iz programa;
 - pomeren miš, pri čemu se uzimaju trenutne koordinate miša;
 - kliknuto na dugme miša pri čemu se opet uzimaju trenutne koordinate miša i postavlja logička promenljiva da se desio klik tastera miša.

- Za ispitivanje da li je bilo događaja datog tipa koristi se poziv:

```
for event in pygame.event.get(QUIT): izlaz()
for event in pygame.event.get(KEYUP): # uzmi sve KEYUP dogadjaje
    if event.key == K_ESCAPE: izlaz()
    pygame.event.post(event) #prosledi ostalim KEYUP objekt
```

Python: pygame, događaji

- Vremenske funkcije:

```
pygame.time.get_ticks() #ms od poziva pygame.init()
pygame.time.wait(1000) #pauza 1000ms, NE koristi procesor
pygame.time.delay(1000) #pauza 1000ms, koristi procesor
pygame.time.set_timer(eventid, ms) # nakon ms aktiviraj event
```

- Klasa `pygame.time.Clock`

```
pygame.time.Clock() #kreiranje objekta za vreme
```

metode su:

- `tick(framerate=0)` , koliko je ms prošlo od prethodnog poziva
parametar `framerate` omogućuje da se limitira brzina igre npr. poziv `tick(40)` po frejmu ograničava brzinu do max 40 FPS. Neprecizna je i zavisi od platforme.
- `tick_busy_loop(framerate=0)` , precizniji od `tick`-a.
- `get_time()` , broj ms između dva poziva `tick`-a
- `get_rawtime()` , kao prethodni osim što ne uključuje vreme odlaganja za *framerate*
- `get_fps()` , vraća *float* broj frejmova u sekundi FPS (prosečno prema 10 poziva `tick`-a)

Primer: Asteroid invasion, **modifikovati za domaći**

- Sledi opis 2D igre Asteroid invasion :
 - na ekranu 700x600 pri 60 frejmova u sekundi potrebno je izbegavati sudar sa asteroidima uz istovremeno pucanje u asteroide različite veličine.
 - pogodak u asteroid uništava asteroid.
 - igrač raspolaže sa 3 života i tri power-up-a (povećanje broja života, dvostruko pucanje, uništenje svih asteroida na ekranu – strelica gore pri čemu se resetuje traka za aktivaciju ovog power up-a)
 - pucanje, pogodak astereoida ili svemirskog broda prate zvučni efekti
- Neki elementi programa:
 - `WIDTH = 700; HEIGHT = 600; FPS = 60`, postavljanje veličine ekrana i brzine igre
 - `HS_file`, datoteka najboljih rezultata
 - `random`, generisanje slučajnih brojeva
 - `os.path.join(path.dirname(__file__), 'folder')`,
 - za kreiranje staze od korenskog foldera do foldera resursa
 - `pygame.image.load('assets/images/galgadas.png')`,
 - učitavanje slike
 - `BLACK = (0, 0, 0)`, n-torka boje

Primer:Asteroid invasion

```
import pygame
import random
from os import path
WIDTH = 700; HEIGHT = 600; FPS = 60 # ekran WIDTHxHEIGHT, brzina FrPerSe
HS_FILE='highscore.txt' # datoteka najboljih rezultata
hdir=path.dirname(__file__) # root folder igre
img_dir = path.join(hdir, 'meteors') # folder meteora
img_dir2= path.join(hdir, 'assets/images/')# folder slika
bg=pygame.image.load('assets/images/galgadas.png') # zvezdano nebo
WHITE = (255, 255, 255) # n-torke boja
BLACK = (0, 0, 0)
RED = (255, 0, 0)
GREEN = (0, 255, 0)
BLUE = (0, 0, 255)
YELLOW = (255, 255, 0)
icon=pygame.image.load('assets/images/asteroid invasion.png') # prva slika
life=pygame.image.load('assets/images/life.png') # slika zivota
meteor_images = [] # lista objekata slika meteora
meteor_list = ['meteorBrown_big1.png', 'meteorBrown_med1.png', # sl. meteora
               'meteorBrown_big2.png', 'meteorBrown_big3.png',
               'meteorBrown_med3.png',
               'meteorBrown_small1.png', 'meteorBrown_small2.png',
               'meteorBrown_tiny1.png', 'meteorBrown_tiny2.png']
```

Primer:Asteroid invasion

```
powerup_images = {} # recnik power up -ova
powerup_images['gun'] = pygame.image.load(path.join(img_dir2,
'powerpowerup.png'))
powerup_images['life'] = pygame.image.load(path.join(img_dir2,
'life.png'))
pygame.display.set_icon(icon) # postavljanje ikonice igre
pygame.init() # inicijalizacija pygame i kreiranje prozora igre
pygame.mixer.init() # inicijalizacija miksera zvuka
pygame.mixer.music.load('assets/sounds/bgspaceshooter.mp3') # muzika
pygame.mixer.music.play(-1) # sviraj neprekidno
screen = pygame.display.set_mode((WIDTH, HEIGHT)) # ekran WIDTHxHEIGHT
pygame.display.set_caption("asteroid invasion") # naslov prozora
speed=8      # brzina
bspeed=12    # brzina bullet-a
sspeed=14    # brzina sonic-a
powerc=RED   # boja power bloka
pew=pygame.mixer.Sound('assets/sounds/pew.wav') # zvuk pucanja
astrexpl=pygame.mixer.Sound('assets/sounds/astrexpl.wav')#zvuk eks. ast.
for meteors in meteor_list:
    meteor_images.append(pygame.image.load(path.join(img_dir,
meteors)).convert()) # ucitavanje slika uz povecanu brzinu prikaza
clock = pygame.time.Clock() # otkucaji od ovog trenutka
```


Primer:Asteroid invasion

klasa Player izvedena iz klase Sprite

```
class Player(pygame.sprite.Sprite):
    def __init__(self):
        pygame.sprite.Sprite.__init__(self)
        self.image = pygame.image.load('assets/images/player.png')
        self.rect = self.image.get_rect()
        self.radius=29
        self.rect.centerx = WIDTH / 2
        self.rect.bottom = HEIGHT - 10
        self.speedx = 0
        self.hidden = False
        self.lives=3
        self.soniclaser=0
        self.power=1
        self.hide_timer = pygame.time.get_ticks()

    def update(self):
        if self.hidden and pygame.time.get_ticks()-self.hide_timer>1000:
            self.hidden = False
            self.rect.centerx = WIDTH / 2
            self.rect.bottom = HEIGHT - 10
        self.speedx = 0
```

Primer:Asteroid invasion

```
keystate = pygame.key.get_pressed() # pritisnut taster
if keystate[pygame.K_LEFT]:
    self.speedx = -speed
if keystate[pygame.K_RIGHT]:
    self.speedx = speed
self.rect.x += self.speedx
if self.rect.right > WIDTH:
    self.rect.right = WIDTH
if self.rect.left < 0:
    self.rect.left = 0
```

```
def powerup(self): self.power += 1
```

```
def shoot(self):
    if self.power == 1:
        bullet = Bullet(self.rect.centerx, self.rect.top)
        all_sprites.add(bullet)
        bullets.add(bullet)
        pew.play()
    if self.power >= 2:
        bullet1 = Bullet(self.rect.left, self.rect.centery)
        bullet2 = Bullet(self.rect.right, self.rect.centery)
```

Primer:Asteroid invasion

```
all_sprites.add(bullet1)
all_sprites.add(bullet2)
bullets.add(bullet1)
bullets.add(bullet2)
pew.play()
```

```
def sonicshot(self):
    sonicbull=Sonicshot(self.rect.centerx,self.rect.top)
    all_sprites.add(sonicbull)
    sonicshot.add(sonicbull)
```

```
def hide(self):
    self.hidden = True
    self.hide_timer = pygame.time.get_ticks()
    self.rect.center = (WIDTH / 2, HEIGHT + 200)
```

Primer:Asteroid invasion

```
class Mob(pygame.sprite.Sprite):
    def __init__(self):
        pygame.sprite.Sprite.__init__(self)
        self.image_orig = random.choice(meteor_images)
        self.image_orig.set_colorkey(BLACK)
        self.image = self.image_orig.copy()
        self.rect = self.image.get_rect()
        self.radius = int(self.rect.width * .85 / 2)
        self.rect.x = random.randrange(WIDTH - self.rect.width)
        self.rect.y = random.randrange(-100, -40)
        self.speedy = random.randrange(1, 8)
        self.speedx = random.randrange(-3, 3)

    def update(self):
        self.rect.x += self.speedx
        self.rect.y += self.speedy
        if self.rect.top > HEIGHT + 10 or self.rect.left < -25 or
self.rect.right > WIDTH+ 20:
            self.rect.x = random.randrange(WIDTH - self.rect.width)
            self.rect.y = random.randrange(-100, -40)
            self.speedy = random.randrange(1, 8)
```

Primer:Asteroid invasion

```
class Bullet(pygame.sprite.Sprite):
    def __init__(self, x, y):
        pygame.sprite.Sprite.__init__(self)
        self.image = pygame.image.load('assets/images/laserRed.png')
        self.rect = self.image.get_rect()
        self.rect.bottom = y
        self.rect.centerx = x
        self.speedy = -bspeed

    def update(self):
        self.rect.y += self.speedy
        # kill if it moves off the top of the screen
        if self.rect.bottom < 0:
            self.kill()

class Sonicshot(pygame.sprite.Sprite):
    def __init__(self, x, y):
        pygame.sprite.Sprite.__init__(self)
        self.image = pygame.Surface((WIDTH*2,10))
        self.rect = self.image.get_rect()
        self.rect.bottom = y
        self.rect.centerx = x
        self.speedy = -sspeed
```

Primer:Asteroid invasion

```
def update(self):  
    self.rect.y += self.speedy  
    if self.rect.bottom < 0:  
        self.kill()
```

```
class Explosion(pygame.sprite.Sprite):  
    def __init__(self, center, size): # size moze biti: lg, player  
        pygame.sprite.Sprite.__init__(self)  
        self.size = size  
        self.image = explosion_anim[self.size][0]  
        self.rect = self.image.get_rect()  
        self.rect.center = center  
        self.frame = 0  
        self.last_update = pygame.time.get_ticks()  
        self.frame_rate = 50
```

```
def update(self):  
    now = pygame.time.get_ticks()  
    if now - self.last_update > self.frame_rate:  
        self.last_update = now  
        self.frame += 1  
        if self.frame == len(explosion_anim[self.size]):  
            self.kill()
```

Primer:Asteroid invasion

else:

```
center = self.rect.center
self.image = explosion_anim[self.size][self.frame]
self.rect = self.image.get_rect()
self.rect.center = center # moze i krace
```

class Pow(pygame.sprite.Sprite):

```
def __init__(self, center):
    pygame.sprite.Sprite.__init__(self)
    self.type = random.choice(['life', 'gun'])
    self.image = powerup_images[self.type]
    self.image.set_colorkey(BLACK)
    self.rect = self.image.get_rect()
    self.rect.center = center
    self.speedy = 2
```

```
def update(self):
    self.rect.y += self.speedy
    if self.rect.top > HEIGHT:
        self.kill()
```

Primer:Asteroid invasion

```
font_name = pygame.font.match_font('InsaneHours2')
def draw_text(surf, text, size, x, y):
    font = pygame.font.Font(font_name, size)
    text_surface = font.render(text, True, RED)
    text_rect = text_surface.get_rect()
    text_rect.midtop = (x, y)
    surf.blit(text_surface, text_rect)

with open(path.join(hdir, HS_FILE), 'w+') as f:
    try:
        highscore=int(f.read())
    except:
        highscore=200

def BG():
    screen.fill(WHITE)
    screen.blit(bg,(0,0))

def newmob():
    m = Mob()
    all_sprites.add(m)
    mobs.add(m)
```


Primer:Asteroid invasion

```
def draw_shield_bar(surf, x, y, pct):
    if pct < 0:
        pct = 0
    BAR_LENGTH = 100
    BAR_HEIGHT = 10
    fill = (pct / 100) * BAR_LENGTH
    outline_rect = pygame.Rect(x, y, BAR_LENGTH, BAR_HEIGHT)
    fill_rect = pygame.Rect(x, y, fill, BAR_HEIGHT)
    pygame.draw.rect(surf, powerc, fill_rect)
    pygame.draw.rect(surf, WHITE, outline_rect, 2)

def showscreen():
    BG()
    draw_text(screen, "asteroid invasion", 50, WIDTH/2, HEIGHT/4)
    draw_text(screen, "left and right arrow to move", 20, WIDTH/2, HEIGHT
/ 2-12.5)
    draw_text(screen, "press space to shoot", 20, WIDTH/2, HEIGHT/2-50)
    draw_text(screen, "if red bar turns green ", 20, WIDTH / 2, HEIGHT /
2+22)
    draw_text(screen, "press arrow up to use sonicshot", 20, WIDTH/2,
HEIGHT / 2+44)
    draw_text(screen, 'highscore: '+str(highscore), 18, WIDTH/2, 400)
    draw_text(screen, "Press up arrow to start", 18, WIDTH/2, HEIGHT*3/4)
```

Primer:Asteroid invasion

```
pygame.display.flip()
waiting = True
while waiting:
    clock.tick(FPS)
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            exit()
        pygame.exit()
    if event.type == pygame.KEYUP:
        if event.key == pygame.K_UP:
            waiting = False
```

Primer:Asteroid invasion

```
explosion_anim = {}
explosion_anim['lg'] = []
explosion_anim['player'] = []
for i in range(9):
    filename = 'assets/images/regularExplosion0{}.png'.format(i)
    img = pygame.image.load(filename).convert()
    img.set_colorkey(BLACK)
    img_lg = pygame.transform.scale(img, (75, 75))
    explosion_anim['lg'].append(img_lg)
    filename = 'sonicExplosion0{}.png'.format(i)
    img = pygame.image.load(path.join(img_dir2, filename))
    explosion_anim['player'].append(img)
# glavna petlja igre
running = True
game_over=True
while running:
    if game_over:
        showscreen()
        game_over=False
    all_sprites = pygame.sprite.Group()
    mobs        = pygame.sprite.Group()
    bullets     = pygame.sprite.Group()
```

Primer:Asteroid invasion

```
sonicshot    = pygame.sprite.Group()
powerups     = pygame.sprite.Group()
player=Player()
all_sprites.add(player)
score=0
powerc=RED
for i in range(9):
    newmob()
# kontrola brzine igre
clock.tick(FPS)
# ulazni dogadjaji
for event in pygame.event.get():
    if event.type == pygame.QUIT:
        running = False
    elif event.type == pygame.KEYDOWN:
        if event.key == pygame.K_SPACE and player.hidden==False:
            player.shoot()
        if event.key == pygame.K_UP and player.soniclaser==100:
            powerc=RED
            player.soniclaser=0
            player.sonicshot()
# azuriranje sprajtova
all_sprites.update()
```

Primer:Asteroid invasion

```
# asteroida je pogodio bullet?
```

```
hits = pygame.sprite.groupcollide(mobs, bullets, True, True)
```

```
for hit in hits:
```

```
    expl = Explosion(hit.rect.center, 'lg')
```

```
    all_sprites.add(expl)
```

```
    newmob()
```

```
    score+=50
```

```
    player.sonicclaser+=2
```

```
    astrexp1.play()
```

```
    if random.random() > 0.98:
```

```
        pow = Pow(hit.rect.center)
```

```
        all_sprites.add(pow)
```

```
        powerups.add(pow)
```

```
# asteroida je pogodio sonic ?
```

```
hits = pygame.sprite.groupcollide(mobs, sonicshot, True, False)
```

```
for hit in hits:
```

```
    expl = Explosion(hit.rect.center, 'lg')
```

```
    all_sprites.add(expl)
```

```
    newmob()
```

```
    score+=10
```

```
    astrexp1.play()
```

Primer:Asteroid invasion

```
# igraca je pogodio asteroid ?
hits = pygame.sprite.spritecollide(player, mobs, False,
pygame.sprite.collide_circle)
if hits :
    death_explosion = Explosion(player.rect.center, 'player')
    astrexp1.play()
    all_sprites.add(death_explosion)
    player.hide()
    player.lives -= 1
    player.power=1
    powerc=RED
    player.soniclaser=0
if player.lives==0 and not death_explosion.alive():
    game_over = True
# igraca je pogodio power_up ?
hits = pygame.sprite.spritecollide(player, powerups, True)
for hit in hits:
    if hit.type == 'gun':
        player.powerup()
        score+=100
    if hit.type=='life':
        player.lives+=1
```

Primer:Asteroid invasion

```
if player.soniclaser>=100:
    player.soniclaser=100
    powerc=GREEN
# renderovanje
BG()
draw_text(screen,str(player.lives)+'x',18,648,10)
draw_text(screen, str(score), 18, WIDTH / 2, 10)
screen.blit(life,(665,0))
all_sprites.draw(screen)
draw_shield_bar(screen, 5, 5, player.soniclaser)
# koriscenje dvostrukog bafera
pygame.display.flip()
pygame.quit()
```

