Haaga-Helia
University of Applied Sciences

Ohjelmistokehityksen teknologioita - *S*eminaarityö

Possu jump peli

Seminaari 6 Python

Max Bowen

26.4.2022

Sisältö

Tiivistelmä *1*

1 Johdanto/Ylätason esittely *1*

2 Käytetyt tekniikat *1*

2.1 Pelin aloittaminen *1*

2.2 Hahmon luominen ja sen liikuttaminen *2*

2.3 Painovoiman ja muuttujien luominen *4*

2.4 Alustojen kehittäminen *5*

2.5 Pelin scrollaus ja pelin loppuminen *7*

2.6 Pisteitten tallennus *9*

3 Yhteenveto *10*

Lähdeluettelo *11*

Tiivistelmä

Seminaarityön aiheena on pythonilla PyGame kirjaston avulla kehitetty peli, missä hahmo hyppii alustoja ylöspäin loputtomasti ennen kuin tippuu. Peli loppuu tippumiseen ja käyttäjä voi halutessaan aloittaa peli uudestaan. Käyttäjän piste-ennätykset tallentuvat erilliseen tekstitiedostoon. Motivaationa työhön on minun oma henkilökohtainen kiinnostuksenkohteeni pelejä kohtaan.

Tavoitteena oli oppia PyGamen käyttöä ja toteuttaa sillä peli. Pelin rakentamisessa käytin pythonin funktioita, luokkia, tiedoston avaamista ja siihen kirjoittamista. Peli rakentui askel kerrallaan alkaen peli-ikkunasta pelin pisteytykseen. Tuloksena syntyi melko hauska hyppely peli, jota voi vielä jatkokehittää lisäämällä siihen vielä musiikkia ja muita äänitehosteita.

1 Johdanto/Ylätason esittely

Ohjelmistokehityksen teknologiat kurssilla on päätteeksi seminaarityö vaihe, jossa jokainen tekee oman projektin itseään kiinnostavasta aiheesta. Minun kohdallani tämä oli python, mutta en vielä tiennyt mitä sillä haluaisin tehdä. Valitsin kuitenkin tehdä jonkun pelin PyGame kirjastoa hyödyntäen. Aloitin työn tutkimalla PyGame kirjastoa ja eri materiaaleja sen käytöstä. PyGame kirjasto oli siitä hyvä, että löytyi paljon eri opetusvideoita sen käytöstä ja soveltamisesta.

Työn tarkoituksena oli tutustua ja syventyä PyGame:n eri mahdollisuuksiin. Mietin muutaman päivän, minkälaisen pelin haluaisin kehittää. Lopuksi päädyin kehittämään tasohyppely pelin, missä pelinhahmo eli possu hyppii loputtomasti alustoja ylöspäin. Peli loppuu, jos possu tippuu ruudulta. Lisäsin tähän pisteytys systeemin ja vaikeutin pelinkulkua lisäämällä liikkuvia alustoja, kun pelaaja pääsee tarpeeksi korkealle.

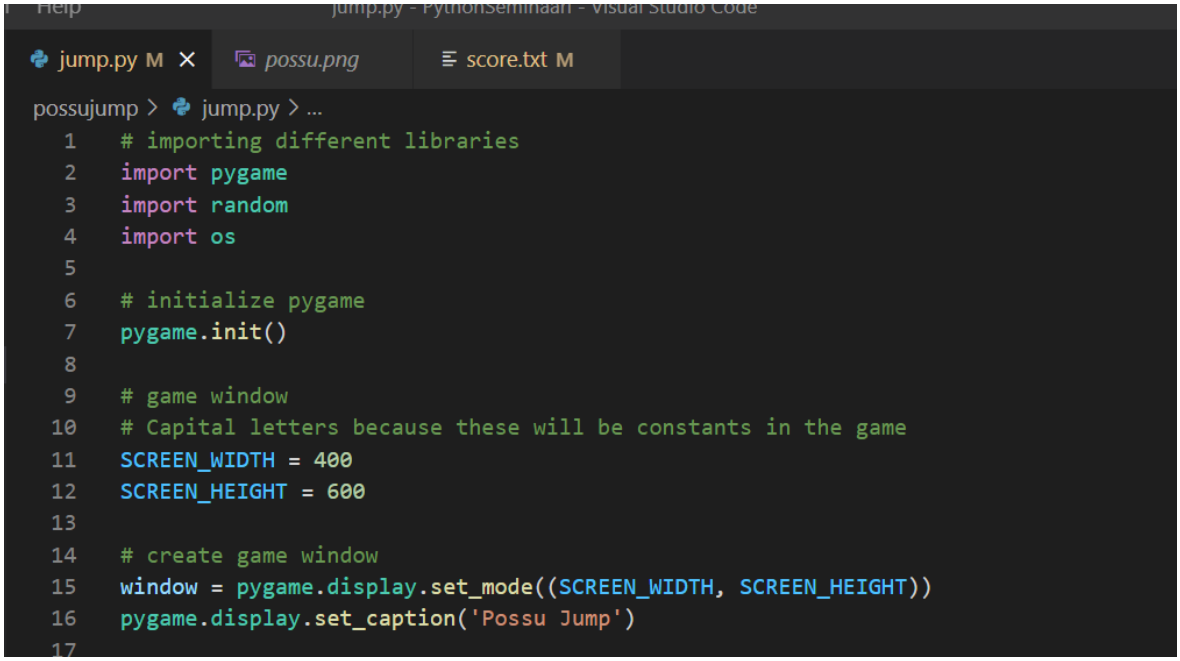
2 Käytetyt tekniikat

Tässä kohdassa käyn läpi pelin eri kehitysvaiheita.

2.1 Pelin aloittaminen

Ensimmäisenä piti tehdä pelille oma ikkuna missä peli pystyy pyöriä. Tämä tapahtui helposti. Oheisessa kuvassa näkyy koodia pelin aloittamisesta. Tärkeintä tässä oli antaa peli-

ikkunalle mitat ja valitsin tässä 400 x 600 koon. Importeista sen verran, että ainoa tärkeä importti tässä vaiheessa oli tietenkin import pygame. Randomia ja os importteja käytin vasta pelin kehityksen loppuvaiheessa.



Kuvasta näkee, miten pelin rakentaminen alkaa PyGame kirjastolla.

Tässä kohdassa loin pelille taustakuvan, hahmolle kuvan ja alusta kuvan.

2.2 Hahmon luominen ja sen liikuttaminen

Kun olin saanut peli-ikkunan luotua ja siihen asetettu taustakuvan aloin luomaan pelin hahmoa, eli possua. Loin pelaaja luokan, johon tarvitsi hahmon eri aloitus koordinaatit ja hahmon koon. Self.flip rivillä 78 käytetään pelaajan liikkumisen yhteydessä. Kun pelaaja menee vasemmalle, hahmo katsoo siihen suuntaan ja flip komennon avulla voidaan hahmo kääntää katsomaan toiseen suuntaan.

```
69  # player character
70  class Player():
71      def __init__(self, x, y):
72          self.image = pygame.transform.scale(possu_img, (80, 80))
73          self.width = 40
74          self.height = 40
75          self.rect = pygame.Rect(0, 0, self.width, self.height)
76          self.rect.center = (x, y)
77          self.vel_y = 0
78          self.flip = False
79
```

Player luokan jälkeen pystyin renderöimään hahmon peliin. Ensin loin hahmolle muuttujan ja sen jälkeen lisäsin hahmon PyGame:n peli looppiin.

```
179   # Keeps game running
180   run = True
181
182   while run:
183
```

Peli loopin luominen

```
169   # The character possu is an instance of the Player() class
170   possu = Player(SCREEN_WIDTH // 2, SCREEN_HEIGHT - 150)
171
```

Pelin hahmo muuttujan luominen

```
262
263       # pygame event handler
264       for event in pygame.event.get():
265           if event.type == pygame.QUIT:
266               run = False
267
268       # update the game window
269       pygame.display.update()
270
271   pygame.quit()
```

Update komennolla saadaan hahmo näkyviin ruudulle ja jos käyttäjä painaa ruksia peli-ikkunassa event handlerilla saadaan peli lopetettua.

Tässä on kuva pelin hahmosta. Valkoinen laatikko possun ympärillä kuvastaa sen sijaintia ja mihin hahmo osuu. Tätä hyödynnettiin alustojen luomisessa. Poistin pelin kehityksen lopussa valkoisen laatikon.

Hahmon luomisen jälkeen piti kehittää hahmon liikuttaminen. PyGame kirjasto pystyy tunnistamaan näppäimen paineluja, joten hahmoa liikutetaan A ja D näppäimillä. Tässä myös käytetään aiemmin mainittua flip komentoa.

```
81      def move(self):
82          # reset variables
83          # changes in the delta variables of the x and y coordinates
84          scroll = 0
85          dx = 0
86          dy = 0
87
88          # keyboard input for moving the possu
89          key = pygame.key.get_pressed()
90          if key[pygame.K_a]:
91              dx = -10
92              self.flip = True
93          if key[pygame.K_d]:
94              dx = 10
95              self.flip = False
96
```

2.3 Painovoiman ja muuttujien luominen

Seuraavaksi piti luoda hahmolle painovoimaa, että hahmo käytännössä putoaa koko ajan ja myöhemmin luodut alustat nostavat hahmoa ylös.

```
17
18  # setting framerate of the game
19  clock = pygame.time.Clock()
20  FPS = 60
21
22  # game variables
23
24  SCROLL_START = 200
25  GRAVITY = 1
26  MAX_PLATFORMS = 10
27  scroll = 0
28  bg_scroll = 0
29  game_over = False
30  score = 0
31
```

Tässä on luotu eri muuttujia, joita käytetään pelissä. Alkuvaiheessa oli vain painovoima muuttuja ja loput tulivat pelin edetessä.

```
96
97      # gravity that pulls the character down increases gravity by 1 each game loop
98      self.vel_y += GRAVITY
99      dy += self.vel_y
100
101     # collision to make sure the character doesn't go off the screen left and right
102     if self.rect.left + dx  < 0:
103         dx = 0 - self.rect.left
104
105     if self.rect.right + dx > SCREEN_WIDTH:
106         dx = SCREEN_WIDTH - self.rect.right
107
```

Tässä kohdassa luodaan painovoima hahmolle, jonka avulla hahmo tippuu alaspäin ja tarkastetaan, että hahmo ei lähde peliruudulta pois vasemmalle tai oikealle.

2.4 Alustojen kehittäminen

Aluksi loin saman tyylisen luokan alustoille, kun mitä käytin hahmon luomiseen.

```python
# Making the platforms for the game
class Platform(pygame.sprite.Sprite):
    def __init__(self, x, y, width, moving):
        pygame.sprite.Sprite.__init__(self)
        self.image = pygame.transform.scale(platform_img, (width, 20))
        self.moving = moving
        self.move_counter = random.randint(0, 50)
        self.direction = random.choice([-1, 1])
        self.rect = self.image.get_rect()
        self.rect.x = x
        self.rect.y = y
```

Alussa ei ollut alustojen liikkuvuutta ollenkaan. Alussa oli vain yksittäiset alustat, joiden x ja y koordinaatit määrittävät mihin alustat ilmestyvät.

```python
172  # create platform groups
173  platform_group = pygame.sprite.Group()
174
175  # create starting platforms
176  platform = Platform(SCREEN_WIDTH // 2 - 40, SCREEN_HEIGHT - 50, 100, False)
177  platform_group.add(platform)
178
```

Tässä luodaan alusta ryhmä, johon lisätään yksittäiset alustat pelin loopissa.

```python
199          # generate platforms
200          # to generate platforms you need x , y and width
201          if len(platform_group) < MAX_PLATFORMS:
202              # platform width between 40 and 60 pixels
203              p_width = random.randint(40, 60)
204              # platform x coordinate
205              p_x = random.randint(0, SCREEN_WIDTH - p_width)
206              # platform y coordinate
207              p_y = platform.rect.y - random.randint(80, 120)
208              p_type = random.randint(1, 2)
209              if p_type == 1 and score > 400:
210                  p_moving = True
211              else:
212                  p_moving = False
213              platform = Platform(p_x, p_y, p_width, p_moving)
214              platform_group.add(platform)
215
```

Tässä oli aluksi vain alustojen leveys, x ja y koordinaatit. Rivistä 208 eteenpäin tuli vasta myöhemmin, kun tein liikkuvia alustoja.

Tässä kuva aloitus alustasta ja siitä seuraavista.

2.5 Pelin scrollaus ja pelin loppuminen

Pelin pitää jatkua loputtomasti ylöspäin siihen asti, kunnes possu putoaa ruudulta alas. Tätä tarkastellaan aiemmin luodulla scroll muuttujalla. Kun hahmo nousee ylöspäin muut komponentit kuten alustat ja taustakuva laskevat. Tämän takia pitää renderöidä taustakuva kaksi kertaa peräkkäin ja käydä sitä läpi loop tyylisesti.

```
118
119         # check if the player is at the top of the screen
120         # if the player moves up the screen everything else moves down with -dy variable
121         if self.rect.top <= SCROLL_START:
122             # if player is jumping
123             if self.vel_y < 0:
124                 scroll = -dy
125
126         # updates the position of the character.
127         # controls the vertical position of the player
128         self.rect.x += dx
129         # will freeze the players position on the screen dy - dy
130         self.rect.y += dy + scroll
131
132         return scroll
133
```

```
if game_over == False:

    scroll = possu.move()

    # checking if the game recognizes the scroll use
    # print(scroll)

    # draw background
    # and loop the background while scrolling
    bg_scroll += scroll
    if bg_scroll >= 600:
        bg_scroll = 0
    draw_bg(bg_scroll)
```

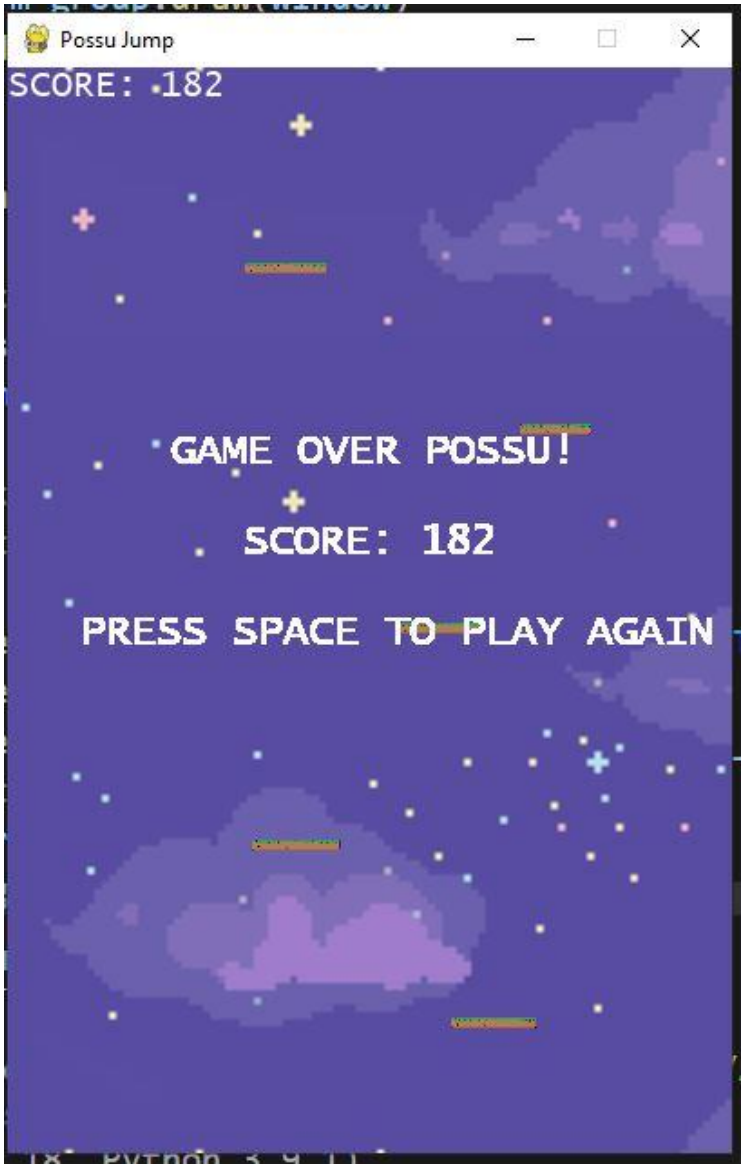
Tässä, jos hahmo menee ruudun ylärajaan eli 600 pixeliä peli aloittaa scrollaamisen.

```
    # check if the platforms are still in the game window
    # gets rid of platforms that leave the screen
    if self.rect.top > SCREEN_HEIGHT:
        self.kill()
```

Jos alusta katoaa ruudulta peli poistaa sen ja luo uuden alustan ruudulle

```
52
53  # function to output gameover text with
54  def draw_text(text, font, text_col, x, y):
55      img = font.render(text, True, text_col)
56      window.blit(img, (x, y))
57
58
59  # function for drawing game score
60  def draw_panel():
61      draw_text('SCORE: ' + str(score), font_small, WHITE, 0, 0)
62
```

Näillä funktioilla näytetään pelaajan pisteet. Kun peli päättyy käyttäjä näkee pelin lopetus-näkymän.



Tässä on pelin lopetusnäkymä

2.6 Pisteitten tallennus

Pisteet tallennetaan erilliseen teksti dokumenttiin, jonka avulla peli tarkistaa onko pelaaja saavuttanut uuden piste-ennätyksen.

```
30
31  if os.path.exists('score.txt'):
32      with open('score.txt', 'r') as file:
33          high_score = int(file.read())
34  else:
35      high_score = 0
36
```

```
245         if score > high_score:
246             high_score = score
247             with open('score.txt', 'w') as file:
248                 file.write(str(high_score))
```

Peliin on rakennettu viiva, joka merkitsee piste-ennätys kohdan

```
224         # draw a line at the previous high score spot
225         pygame.draw.line(window, WHITE, (0, score - high_score + SCROLL_START), (SCREEN_WIDTH, score - hig
226         draw_text('HIGH SCORE', font_small, WHITE, SCREEN_WIDTH - 130, score - high_score + SCROLL_START)
```

Pelissä viiva näyttää tältä:

3 Yhteenveto

Mielestäni tämä oli erittäin hyvä oppimiskokemus. Pythonia olen käyttänyt tällä kurssilla ja basics of python kurssilla. Pygame kirjastoon tutustuminen oli myös erittäin hauskaa ja uskon, että käytän sitä vielä uudestaankin. Peliin voisi vielä lisätä haluttaessa vihollisia, jotka lentävät ruudulla tai tehdä pelistä erillisen .exe tiedoston, mutta päätin rajata työnteon siihen suositeltuun 20 tuntiin. Teemun vinkkaamista Helsingin Yliopiston mooc kurssin lähteistä sai paljon tietoa pelinrakentamiseen. Katsoin myös youtubesta eri videoita pygamen käytöstä.

Tällä hetkellä peli pyörii vscodessa kun käyttäjä laittaa ohjelman pyörimään, mutta kuten aikasemmin mainitsin pelistä voisi tehdä erillinen tiedosto. Mielestäni työ oli hyödyllistä

pythonin oppimisen kannalta, koska tässä työssä tuli käytettyä monta eri käytännön asiaa kuten esimerkiksi erilliset funktiot, luokat ja tiedoston lukeminen sekä siihen kirjoittaminen.

Lähdeluettelo

Helsingin Yliopisto (2022). Ohjelmoinnin perusteet ja jatkokurssi osa 13 & 14. Luettavissa:

https://ohjelmointi-22.mooc.fi/osa-13

https://ohjelmointi-22.mooc.fi/osa-14

Pygame. Pygame documentation. Luettavissa:

https://www.pygame.org/docs/

Clear code (2021) The ultimate introduction to Pygame katsottavissa:

https://www.youtube.com/watch?v=AY9MnQ4x3zk

Real Python. PyGame: A Primer on Game Programming in Python. Luettavissa:

https://realpython.com/pygame-a-primer/