Projektisuunnitelma - Ohjelmoinnin peruskurssi Y2

Tekninen suunnitelma

1. Ohjelman rakennesuunnitelma

Koko ohjelma tulee ”while True” -loopin sisään, joka pyörii niin kauan, kunnes pelaaja päättää lopettaa pelin. Luin netistä esimerkkejä läpi ja ymmärsin, että peli olisi järkevä koodata useampaan luokkaan.

Ohjelmaan tulisi lisätä asioita kuten grafiikat, pelaajahahmon liikuttaminen, painovoima, törmäyksien estäminen, sekä mahdolliset muut lisäominaisuudet. Nämä voitaisiin lisätä joko metodien tai luokkien kautta. Kenttäeditorille täytyy myös tehdä oma luokka.

Mietin ainakin seuraavien luokkien tekemistä peliin:

class Player:
- Sisältää pelaajan nimen
- Liikuttaa pelaajaa

class Level:
- Muodostaa halutun tason
- Mahdollisesti kutsutaan Platform -luokkaa, jossa tehdään objekteja peliin
- def Gravity:
    - Tekee tasolle painovoiman
- def CollisionDetection
    - Estää törmäämisen eri objekteihin

class Platform:
- Tehdään objekteja/tasoja peliin

class WorldEditor:
- Mahdollistaa uusienkenttien teon

while True:
- Pelin looppi, joka pyörii niin kauon kunnes pelaaja päättää lopettaa pelin.
- Kutsutaan ylläolevia luokkia

Olen hieman hukassa tämän asian kanssa ja kuulisin mielelläni kurssiassistentin mielipidettä/vinkkejä käytännön toteutukseen.

2. Käyttötapauskuvaus

Pelaajan käynnistäessä ohjelman, ohjelma kysyy pelaajan nimeä ja tämä voi syöttää tähän haluamansa merkkijonon. Tämän jälkeen pelaaja voi aloittaa pelin, jonka tavoitteena on läpäistä kyseinen taso. Pelaaja voi käyttää oikeaa ja vasenta nuolinäppäimiä liikkuakseen ja ylänuolinäppäintä pomppiakseen, ja koodi reagoi jokaiseen nuolinäppäimen painamiseen ja liikuttaa pelihahmoa tämän mukaan.

Jos pelaaja yrittää liikkua jonkin objektin läpi, yksi koodin metodeista estää tämän. Vastaavasti pelaajan ”kuollessa”, esimerkiksi tippuessa kuoppaan, koodi huomioi tämän ja pyytää pelaajaa aloittamaan pelin uudelleen. Pelaajan pomppiessa tai tippuessa painovoimasta vastaava koodinpätkä kutsutaan jokaisella pompulla.

3. Algoritmit

Liikuttaessa pelihahmoa, koodi laskee liikuttavan etäisyyden pikseleissä nuolinäppäin painalluksien mukaan. Käyttäen hyödyksi koordinaatteja, koodi huomioi mahdolliset objektit pelissä ja estää pelaajaa menemästä niiden läpi. Suuriosa pelin tekoälystä toimii koordinaattien mukaan.

4. Tietorakenteet

Tietorakenteista esimerkiksi listat tulevat hyödylliseksi varsinkin tehdessä pelikarttaa (”maata”). Lisäksi koodatessa erilaisia törmäysskenaarioita, sanakirjan käyttämisestä tulee olemaan apua. Uskon, että tietorakenteiden todellinen käyttäminen selkenee itse pelin koodauksen aikana.

5. Kirjastot

Useissa tutorialeissa on suosittu arcade ja pygame -kirjastoja. Kyseiset kirjastot mahdollistavat useiden ominaisuuksien lisäämisen peliin. Lisäksi pelin luonnista tulee huomattavasti vaivattomampaa, koska kyseisissä kirjastoissa on jo valmiit käskyt useille tasohyppely-pelissä tarvittaville käskyille.

Luin kirjastoista eri lähteistä, ja käsittääkseni molemmat soveltuvat tasohyppelypelin koodaamista varten hyvin. Itselleni tämä tulee olemaan ensimmäinen koodaamani peli, joten kuulisin mielelläni kurssiassistentin mielipidettä mahdollisesta käytettävästä kirjastosta.

6. Aikataulu

1.3. - 6.3.

- Projektin suunnitelman laatiminen
- Lähteiden läpikäymistä, videoiden katselua
- Yhteensä 10h

9.3. – 13.3

- 11.3 ohjaustapaaminen assistentin kanssa
- Palautteen reflektointi
- Koodin kirjoittamisen aloittaminen (4h)
    - Pelinäkymän luominen
    - Alkeellisen pelikentän luominen
    - Pelaajan lisääminen peliin
    - Pelihahmon liikkumisen lisääminen
- Tutorialien katselua (2h)

16.3 – 20.3

- Metodien lisääminen koodiin (6h)
    - Pelikentän muokkaaminen
    - Erilaisten objektien lisääminen pelikenttään
    - Törmäyksen estäminen
    - Pelin testaaminen

23.3 – 27.3

- Mahdollinen checkpoint aika?
- Pelikentän ja pelihahmon grafiikkojen parantaminen (6h)
- Pelikenttäeditorin tekeminen (4h)
- Pelin testaaminen (3h)

30.3 – 3.4

- Uusien kenttien luominen (5h)
- Uusien kenttien testaaminen (5h)

6.4 – 10.4

- Pelin hiominen (8h)
- Pelin viimeisteleminen projektidemoa varten

13.4 - 17.4

- Projektidemo?

7. Yksikkötestaussuunnitelma

Aion tehdä testata ohjelman niin, että eri testit käyvät ohjelman kaikki (ainakin suurimman osan) rivit läpi, jotta mahdollisilta virhetilanteilta vältyttäisiin. Aion myös testata peliä pelaamalla peliä ja kokeilemalla erilaisia strategioita, jotka voisivat johtaa virhetilanteeseen. Lisäksi, aion lukea netistä mahdollisista virhetilanteista, jotka usein jäävät huomioimatta.

8. Kirjallisuusviitteet ja -linkit

- Pygame dokumentaatiota
  https://www.pygame.org/docs/

- Python arcade kirjasto, josta löytyy vinkkejä kyseiseen kirjastoon
  https://arcade.academy

- DaFluffyPotato:n youtube kanava, josta löytyy videoita tasohyppelypelin koodaamisesta
  https://www.youtube.com/channel/UCYNrBrBOgTfHswcz2DdZQFA

- Wikipedia/Platform game
  https://en.wikipedia.org/wiki/Platform_game

- Wikipedia/Collision detection
  https://en.wikipedia.org/wiki/Collision_detection

9. Liitteet

-