

Juhana Kalastuspeli V.1

Ohjelmoinnin peruskurssin harjoitustyö

Kerttulin Lukio 25.1.2010

Tekijä: Juhana Moilanen 19D

Ohjelman kuvaus ja laajentaminen

Ohjelma on kalastuspeli. Pelaajalta kysytään miten pitkälle soudetaan rannasta ja mikä uistin valitaan. Matkasta lasketaan syvyys ja uistimesta lasketaan miten syvälle se uppoaa vedessä. Näiden kahden perusteella katsotaan uppoaako uistin pohjaan, jääkö uistin liian korkealle veteen vai saako pelaaja saalista. Jos pelaaja saa saalista, saalis arvotaan sen syvyyden kalojen listasta, saaliille lasketaan paino ja pelaajaa onnitellaan.

Ohjelma muodostuu kahdesta tiedostosta: **KalastuspeliListat.py**, johon on talletettu kaikki pelin listat mm. uistimet ja kalat. **Kalastuspeli.py**, jossa on kaikki muu koodi, aliohjelmat ja dialogi.

Ohjelma ei talleta saaliita ja ohjelma pitää suorittaa uudestaan jokaista kierrosta varten, mutta sitä voisi laajentaa ja siitä voisi luoda kokonaisen pelin: mm. tallentamalla saadut kalat, lisäämällä saaliita, lisäämällä eri systeemeitä (kuten värit) uistimille.

Ohjelmaa voisi myös parantaa luomalla kokonaan oma tiedosto kaikille pelissä käytetyille apuohjelmille. Myös koodia voisi parantaa ottamalla uistimen paino suoraan uistinListasta sen sijaan että luodaan kokonaan uusi lista.

Ratkaisuperiaate: (+kuva)

Pelaajalta kysytään, haluaako hän ohjeita (joo/ei). Jos syöte on jotain muuta kuin ”joo”, ohjeita ei anneta ja jatketaan. Ohjeissa kerrotaan syvyyden selvityksessä käytetty matematiikka.

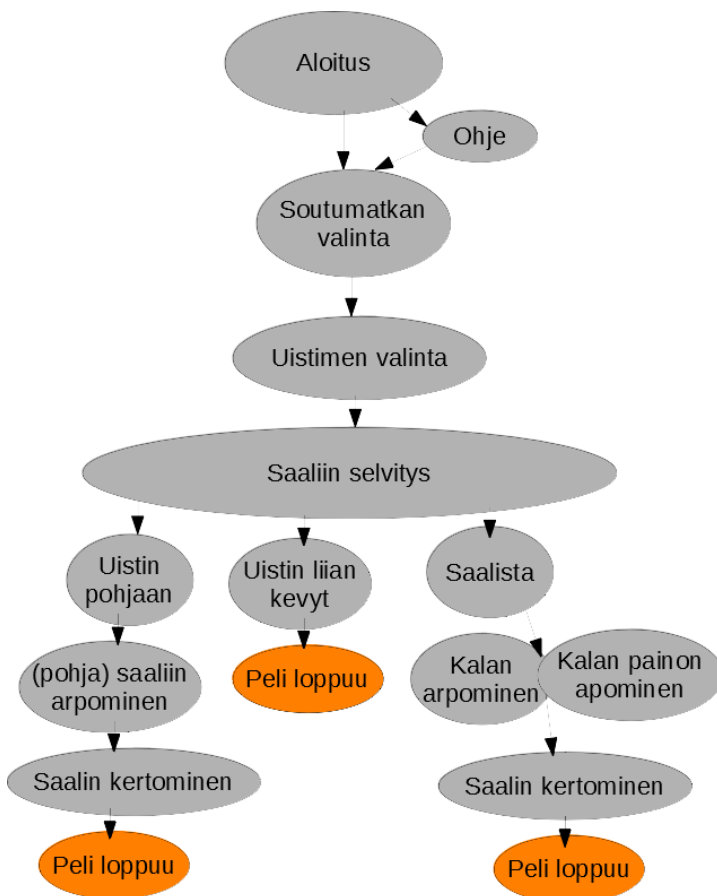
Ohjeiden jälkeen pelaajalta pyydetään, miten pitkälle soudetaan rannasta. Vastauksesta lasketaan veden syvyys jakamalla syöte muutettuna luvuksi kahdella. Syötettä kysytään while-silmukalla ja try-except rakenteella; jos syötettä ei voida muuttaa luvuksi, annetaan virheilmoitus ja kysytään uudestaan.

Syvyyden selvityksen jälkeen pelaajalle esitetään uistimet, joista pelaaja valitsee syöttämällä sen numeron. Taas käytetään while-silmukassa try-except rakennetta, jolla saadaan annettua virheilmoitus väärästä syötteestä. Kysytään taas kunnes pelaaja antaa oikeanlaisen syötteen. Valitun uistimen uppoaminen lasketaan uistinten painojen listasta.

Syvyydellä ja uistimen uppoamisella voidaan viimein arpoa mitä saalista tuli; jos saalista saa. Apuohjelma 'mikaSaalis()' saa parametreina syvyyden ja uistimen uppoamisen. Se tarkistaa onko uistimen uppoaminen suurempi kuin syvyys, jolloin uistin osuu pohjaan; kutsutaan apuohjelma 'osuiPohjaan()' ja peli loppuu.

Jos uistin ei uponnut pohjaan, katsotaan onko syvyys ja uistimen paino suurempia kuin tietyt arvot, joilla muodostetaan eri vyöhykkeet; joissa on erilaisia kaloja. Löydettyään oikean vyöhykkeen apuohjelma arpoo kalan ja määrittää kalalle maksimi- ja minipainot; joiden välistä arvotaan kalan paino toista aliohjelmaa käyttäen. Saalis (kala +saatu paino) talletetaan ja palautetaan.

Saalis kerrotaan pelaajalle käyttäen apuohjelmaa 'saiKalan()' ja peli loppuu.



Ohjelman ja sen osien kuvaaminen

KalastuspeliListat.py

Listat:

listat 'kalaListaXtoY' sisältävät erilaiset mahdolliset kalat joita voi saada x-y syvyydeltä
kalat ovat pelkkiä merkkijonoja

lista 'pohjaLoot' sisältää erilaiset mahdolliset saaliit joita voit saada kun uistin osuu pohjaan
merkkijonoja

lista 'uistinLista' sisältää numeroittain uistimet ja niiden painon (eri yksiköissä g-kg)
merkkijonoja

lista 'uistinPainot' sisältää uistinten painot grammoina
alkiot jaetaan kuudella, jotta saadaan uistimen uppoaminen

Kalastuspeli.py

Ensin tuodaan 'KalastuspeliListat.py' listat ja
random-kirjastosta funktiot choice ja randint

Aliohjelmat:

proseduuri aloitus()

Esittelee pelin nopeasti pelaajalle ja kysyy pelaajalta haluaako tämä nähdä ohjeita.

proseduuri ohje()

Antaa pelaajalle tiedon, miten syvyys ja uistimen uppoaminen lasketaan, ja muuta pitäisi varoa.

funktio valitseSyvyys()

Kysyy pelaajalta miten pitkälle tämä haluaa soutaa rannasta kunnes syöte on lukuarvo. Syötteestä lasketaan sen kohdan veden syvyys, joka palautetaan.

funktio valitseUistin()

Esittelee pelaajalle uistimet ja pyytää pelaajaa valitsemaan niistä yhden. Kysytään taas kunnes syöte on lukuarvo. Valitun uistimen indeksillä etsitään uistinten paino-listalta uistimen paino, joka palautetaan.

funktio mikaSaalis(syvyys, uistinPaino)

Laskee osuiko uistin pohjaan, vai onko syvyys ja uistinPaino molemmat samalla tasolla. Jos ovat, asetetaan minimi- ja maksimipainot ja arvotaan saatu kala. Kalan painoa varten kutsutaan arvoKalanPaino(minPaino, maxPaino) johon annetaan minimi- ja maksimipaino, joka palauttaa saadulle kalalle arvotun painon. Saadun kalan nimi ja paino talletetaan merkkijonoksi, kutsutaan proseduuri saiKalan(loot).

Jos uistin osuu pohjaan, kutsutaan proseduuri osuiPohjaan().

Jos syvyys ja uistinPaino eivät ole molemmat sopivia mihinkään vyöhykkeeseen, kerrotaan pelaajalle, että uistin oli liian kevyt eikä saalista tullut.

proseduuri osuiPohjaan()

Kertoo pelaajalle, että tämä osui pohjaan ja arpoo hänen saaman pohja-saaliin.

proseduuri saiKalan(loot)

Kertoo pelaajalle saamansa saaliin, ja onnittelee häntä.

Lopuksi ohjelmassa kutsutaan apuohjelmat ja peli alkaa.

Ohjelman testaus:

Testasin ohjelmaa syöttämällä sekalaisia merkkejä ja kirjaimia, mutta ohjelma toimi toivottuun tapaan ja heitti virheilmoituksen – ja kysyi syötettä uudestaan. Alussa ohjeiden valinnassa syöttää muuta kuin ”joo”/”ei”, peli jatkaa antamatta ohjeita, joka on täysin suunniteltua.

En onnistunut rikkomaan peliä.