

MILLAISIA ASIOITA OHJELMOIJA LUO?

Seuraavassa on esitelty monipuolinen mutta satunnainen valikoima oikeaa elämää sivuavia, yksinkertaistettuja esimerkkejä siitä, millaisissa yhteyksissä tarvitaan ohjelmointitaitoja – aina Harry Potter -elokuvien tekemisestä itsensä parkkeeraavaan autoon ja jätteitä lajittelevaan robottijärjestelmään.

WHATSAPP-VIESTIOHJELMA

Ohjelmoijan tehtävänanto: Luo älypuhelimelle sovellus, jonka avulla ihmiset voivat lähettää toisilleen perinteisten tekstiviestien tapaisia viestejä, mutta internetin välityksellä.

Mitä ohjelmoija tekee: Ohjelmoija pohtii esimerkiksi, miten viestit saadaan kulkemaan puhelimesta toiseen käyttäen puheliverkon sijaan nettiyhteyttä. Ohjelmoijan haaste on ohjeistaa tietokonetta siten, että viestit kulkevat mahdollisimman nopeasti, ja sovelluksen toiminta ei hidastu, vaikka sitä käyttäisivät kymmenet miljoonat ihmiset yhtä aikaa.

Huomioitavaa: WhatsAppin kaltaisessa sovelluksessa äärimmäisen tärkeää on, että sovellusta on helppo ja miellyttävää käyttää. Siksi WhatsAppin tapauksessa käyttöliittymän suunnittelijalla on valtava rooli: sovelluksen pitää näyttää ymmärrettävältä ja intuitiiviselta välittömästi.

ITSENSÄ PARKKEERAAVA AUTO

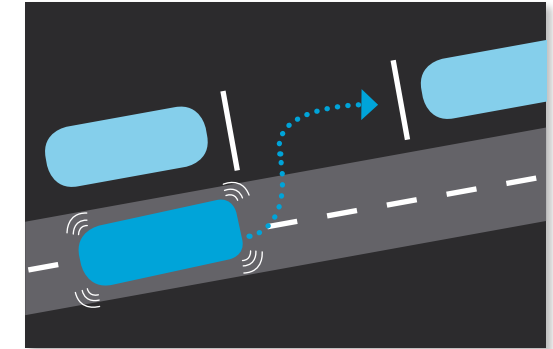
Ohjelmoijan tehtävänanto: Suunnittele autoa ohjaava ohjelmisto, joka osaa osaa taskuparkkeerata auton itsestään.

Mitä ohjelmoija tekee: Ohjelmoija saa tietoja auton etu- ja takaosissa olevilta sensoreilta. Ne kertovat esimerkiksi, kuinka kaukana auto on milläkin hetkellä katukivetyksestä, muista autoista tai ohi kulkevista ihmisistä. Parkkeerattava auto ei saa törmätä mihinkään näistä.

Ohjelmoija opettaa auton sisällä olevalle tietokoneelle sääntöjä: kun olet näin kaukana katukivetyksestä, käännä renkaita tämän verran ja paina kaasua tämän verran. Kun olet näin ja näin lähellä estettä, pysähdy heti, jotta et törmää.

Kun ohjelmoija on pohtinut kaikki mahdolliset tilanteet ja asennot joissa auto voi olla, hän antaa tietokoneelle toimintaohjeet eri tilanteisiin. Tämän jälkeen tietokone osaa automaattisesti parkkeerata auton juuri oikein ja tarkasti yhä uudestaan.

Huomioitavaa: Itsensä parkkeeraavan auton kehittäminen vaatii rutkasti tuotekehitystä, yrityksiä ja erehdyksiä. Kaikkia eri tilanteita on lähes mahdoton kuvitella ennen kuin autoa on testattu erilaisissa arkielämän tilanteissa yhä uudestaan. Vasta vuosien testien jälkeen auto on valmis kauppoihin.



Itsensä parkkeeraava auto aistii ympäristöään sensoreilla ja osaa siksi taskuparkkeerata törmäämättä.

HARRY POTTER -ELOKUVAT

Ohjelmoijan tehtävänanto: Harry Potter -elokuissa pelataan Huispausta. Huispaus on peli, jossa Tylypahkan oppilaat jahtaavat lentävillä luudilla Sieppi-nimistä lentävää esinettä. Suuri osa kohtauksista on animoitava tietokoneella, koska Sieppiä tai lentäviä luutia ei ole oikeassa elämässä. Suunnittele ohjelma, jonka avulla voidaan piirtää Siepille erilaisia lentoratoja.

Mitä ohjelmoija tekee: Ohjelmoija lukee ensin Harry Potter -kirjat ymmärtääkseen, miten Sieppi lentää. Hän oppii, että Sieppi pysähtyy välillä leijumaan paikoilleen ja singahtaa sitten johonkin satunnaiseen suuntaan.

Ohjelmoija opiskelee tai palauttaa mieleensä

hiukan matematiikkaa ja fysiikkaa. Niiden pohjalta hän muodostaa esimerkiksi kiihtyvyyden, painovoiman ja nosteen kaavojen tapaisia yhtälöitä. Niitä tarvitaan, jotta tietokonekin ymmärtää, miten Siepin liikeradat pitää mallintaa, jotta ne näyttävät ihmisen silmään luonnollisilta.

Ohjelmoija kokeilee, millaiset kaavat saavat yksinkertaisen palluran liikkumaan ruudulla, kuten hän haluaisi Siepin liikkuvan elokuvassa. Kun ohjelmoija on tyytyväinen, hän luovuttaa ohjelman animointiosaston ihmisille. He animoivat varsinaiseen elokuvaan yksinkertaisen palluran tilalle oikean Siepin näköisen esineen.

Huomioitavaa: Tosielämässä elokuvien erikoistehosteita suunnitellaan ohjelmilla, jotka tuntevat valmiiksi valtavan määrän fysikaalisia liikkeitä. Tämän vuoksi Harry Potterin kohdalla ei ole luultavasti tarvinnut lähteä enää alusta, vaan ohjelmoija on selvinnyt jalostamalla olemassaolevaa ohjelmakoodia.

SAIRAALAN POTILASTIETOJÄRJESTELMÄ

Ohjelmoijan tehtävänanto: Luo järjestelmä, johon sairaala tallentaa jokaisen uuden potilaan tiedot, ja josta lääkärit voivat tutkia potilaan historiaa.

Mitä ohjelmoija tekee: Ohjelmoija ottaa selvää, mitä tietoja potilaasta pitää tallentaa.

Esimerkiksi nimi, osoite, ikä, sukupuoli, pituus, paino, aiemmat sairaudet, aiemmat sairaalakäynnit, rokotukset, leikkaukset, lääkitykset ja niin edelleen.

Ohjelmoija käskee tietokonetta muodostamaan tältä pohjalta tietokannan, johon hän varaa tilaa kaikille tarvittaville tiedoille ja kymmeniletuhansille potilaille. Ohjelmoija työskentelee yhdessä käyttöliittymäsuunnittelijan kanssa luodakseen sairaalan henkilökunnan tietokoneille näkymän, jonka avulla tietoja on helppo syöttää järjestelmään ja toisaalta saada sieltä tarvittaessa ulos.

Huomioitavaa: Terveystietojärjestelmät ovat monimutkaisia hankkeita, koska vanhat ja uudet järjestelmät joutuvat usein olemaan yhteyksissä toisiinsa. Tämä luo haasteita ohjelmointiprojekteihin, sillä vanhat järjestelmät on saatettu ohjelmoida tavoilla ja kielillä, joilla ei ole paljonkaan yhteistä nykyaikaisten tekniikoiden kanssa.

ANGRY BIRDS -MOBIILIPELI

Ohjelmoijan tehtävänanto: Angry Birds -pelissä yritetään ampuja ritsalla lintuja siten, että ne osuvat vihreisiin possuihin. Possut ovat linnoitautuneet puusta, kivistä tai vaikkapa jäältä tehtyihin linnakkeisiin. Ohjelmoi peli siten, että

kun lintu osuu possun linnakkeeseen, linnake hajoaa tavalla, jolla sen voisi kuvitella hajoavan oikeassa elämässä.

Mitä ohjelmoija tekee: Ohjelmoija selvittää pelin luovalta tiimiltä, millaisia linnakkeita pelissä tarvitaan. Puu on heikompaa kuin jää, ja jää on heikompaa kuin kivi. Kaikki materiaalit hajoavat siis eri tavalla, kun lintu osuu niihin. Ohjelmoija saa peliä pyörittävältä niin kutsutulta pelimoottorilta jo valmiiksi pelin sisäistä dataa siitä, missä kulmassa ja millä nopeudella lintu tulee linnaketta kohti.

Hänen tehtävänä on opettaa tietokoneelle, miten linnakkeen pitää sortua, kun lintu osuu sen eri kohtiin.

Ohjelmoija opettaa tietokoneelle luonnonlakeihin liittyviä matemaattisia kaavoja: painovoiman, kitkan, liikemäärän ja niin edelleen. Kun ne on opetettu tietokoneelle oikein, linnakkeet alkavat sortua kuten ne sortuisivat oikeassa elämässä.

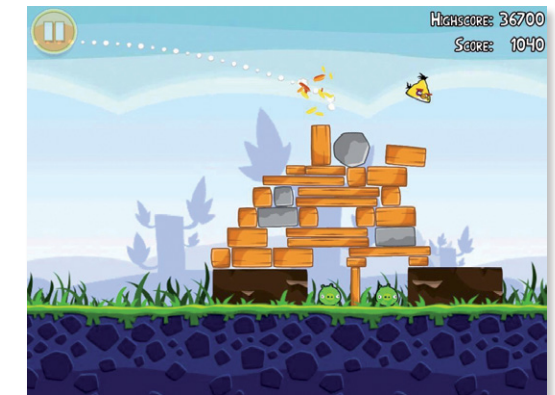
Tämän jälkeen ohjelmoija voi tehdä muutoksia esimerkiksi siten, että kivi ei olekaan ihan yhtä kestävää kuin oikeassa elämässä, tai että sininen lintu särkee jäätä tehokkaammin kuin punainen lintu.

Huomioitavaa: Esimerkki on kirjoittajien kuvitteleva. Pelien kehittäessä paukut käytetään usein aluksi siihen, että luodaan niin sanottu työkalukokoelma, jonka avulla pelin kehittäminen ja

testaaminen on helpompaa.

Esimerkiksi Angry Birds -tapauksessa voidaan aluksi aloittaa ohjelmoimalla työkalu, jolla pelin kehittäjä voi rakentaa ruudulle erilaisia linnakkeita yksinkertaisesti raahaamalla näytöllä puu- ja kivipalikoita erilaisiksi kasoiksi. Kun tuo työkalu on valmis, yhä uusien pelin maailmojen ja tasojen luominen on helppoa, sillä pohjatyö on tehty kunnolla.

Asiaa voisi verrata siihen, että puita sahaava firma kehittää moottorisahan: tavallisella sahalla pääsee nopeammin alkuun, mutta kun moottorisaha on lopulta saatu kehitettyä, sillä tekeminen on paljon tehokkaampaa. Se ottaa äkkiä tavallisella sahalla saadun etumatkan kiinni ja menee pian heittämällä ohi.



Angry Birds -pelissä pommitetaan linnuilla vihreiden possujen linnakkeita. Kuva: Rovio.

JÄTTEITÄ LAJITTELEVA ROBOTTI

Ohjelmoijan tehtävänanto: Käytössämme on robottikoura (ks. oikean palstan kuva), joka osaa tarttua liukuhihnalla kulkeviin asioihin. Lisäksi meillä on digitaalinen videokamera, joka kuvaa liukuhihnaa. Liukuhihnalla kulkee sekaisin kiviä sekä roskia. Opeta robottikouraa poimimaan hihnalta kivet roskien seasta.

Mitä ohjelmoija tekee: Ohjelmoija laittaa videokameran ottamaan tuhansia kuvia liukuhihnalla kulkevasta materiaalista. Ohjelmoija näyttää kuvia tietokoneelle ja kertoo sille, missä kohden kuvissa on kivi.

Ohjelmoija käskee konetta tallentamaan muistiin kaikki kuvat ja tiedon siitä, missä kohden niissä näkyy kivi. Kun tätä on jatkettu jonkin aikaa, kone alkaa itsenäisesti hahmottaa, millaiset asiat kuvassa ovat luultavasti kiviä.

Seuraavaksi ohjelmoija antaa koneen yrittää kiven tunnistamista itse. Hän näyttää koneelle kuvia, ja kone yrittää arvata, missä kohden kivi on.

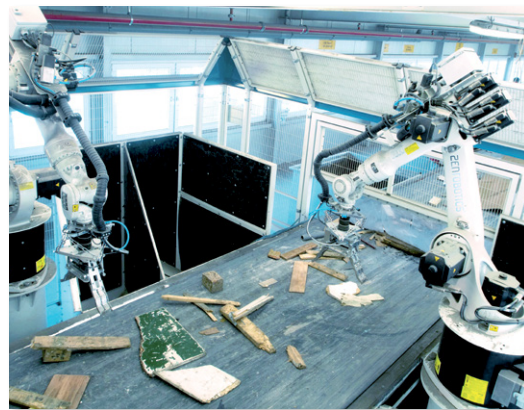
Kun kone onnistuu, ohjelmoija kertoo sille, että kone oli oikeassa. Kun kone epäonnistuu, ohjelmoija kertoo, ettei kohdassa ollut kiveä.

Näin tietokone oppii yhä lisää, ja ohjelmoija säästää kiven osoittamiseen kulunutta omaa aikaansa.

Kun tätä on jatkettu tarpeeksi kauan, kone alkaa tunnistaa kiven esimerkiksi 9 kertaa 10:stä. Tässä vaiheessa prosentti saattaa – käytötarkoituksesta riippuen – olla jo riittävän hyvä siihen, että koneen voi laittaa ohjaamaan robottikouraa, jonka tehtävä on poimia kiviä.

Huomioitavaa: Suomalainen ZenRobotics-yritys valmistaa jätteenlajittelujärjestelmää, jonka yhtenä osana on yllä kuvaillun kaltainen robottikoura. Kouran ja sitä ohjaavan tietokoneen tarkoitus on lajitella erilaisia käyttökelpoisia asioita talteen rakennusjätteen seasta.

Yllä kuvattu työ on tarkemmin sanottuna tekoälyn kehittämistä, jossa yhtenä osa-alueena tarvitaan ohjelmointitaitoa.



Suomalaisen ZenRobotics-yrityksen tekoälyllä toimiva jätteenlajittelujärjestelmä. Kuva: ZenRobotics.

MUSIIKKITAIDETEOS KIASMAAN

Ohjelmoijan tehtävänanto: Luo Kiasmaan taide-teos, joka soittaa tyhjässä valkoisessa huoneessa erilaista äänimaisemaa sen perusteella, kuinka monta ihmistä huoneessa on.

Mitä ohjelmoija tekee: Ohjelmoija tai hänen apurinsa asentaa huoneen kattoon nettikameran ja kytkee sen tietokoneeseen. Tietokoneen hän kytkee myös kaiuttimiin. Ohjelmoija haluaa, että huoneessa olevat ihmiset vaikuttavat siellä kuuluviin ääniin – toisin sanoen äänten pitää olla erilaisia riippuen siitä, millaista kuvaa verkkokamera välittää huoneesta.

Ohjelmoija avaa valmiin musiikinteko-ohjelman, jonka avulla voi soittaa tietokoneversiota erilaisista instrumenteista. Seuraavaksi ohjelmoija kirjoittaa koodia, joka yhdistää verkkokamerasta tulevan kuvan musiikkiohjelmaan. Kun kuvassa näkyy pelkkää valkoista, kuvassa ei selvästikään ole ketään. Tällöin huoneessa voi soida esimerkiksi pelkkä tasainen pianon pimputus, joka tulee ohjelmistosta.

Kun joku kävelee kuvaan, ohjelmoija voi opettaa tietokoneelle esimerkiksi, että vasemmassa laidassa näkyvä liike käynnistää musiikkiohjelmasta viidakkorumpuääniä. Jokainen oikeassa laidassa näkyvä liike puolestaan soittaa jonkin sävelen viulun ääntä.

Kun ihmisiä nyt tulee kuvaan oikeasta laidasta, huoneessa alkaa soida rumpukomppi, jonka päälle jokainen liike kuvan vasemmassa laidassa alkaa säästää pianoa viulun avulla. Taideteos on valmis.

Huomioitavaa: Uudenaikaisilla pelikonsoleilla toimivat, pelaajan liikkeen tunnistavat urheilu- ja tanssipelit perustuvat pohjimmiltaan samankaltaiseen ajatteluun kuin yllä on esitetty: digitaalinen kamera välittää liikkeen tietokoneelle, joka analysoi, mitä sen edessä tapahtuu.

Kun ihmisiä saapuu huoneeseen, kone alkaa soittaa rumpukomppia.