Ohjelmointi II -kurssin lopputyö

RISTINOLLA

Heljä Hietala

31.12.18

Tehtävän kuvaus ja analysointi

Tehtävänäni oli ohjelmoida toimiva ristinolla-peli, jota käyttäjä pelaa tietokonetta vastaan. Ristinolla-peli toimii 3 x 3 ruudukolla, johon pelaajat vuorotellen merkkaavat oman merkkinsä, joko ristin tai nollan. Pelin voittaa se, joka ensiksi saa merkattua kolme peräkkäistä ruudukon ruutua omalla merkillään, kolmesta omasta merkistä muodostuva rivi voi tulla pystysuoraan, vaakasuoraan tai vinottain.

Lähtökohtana oli, että pelin tulee olla ennen kaikkea toimiva, käyttöliittymän ulkoasuun ei tarvinnut käyttää aikaa. Tietokone aloittaa pelin aina sijoittamalla keskelle oman merkkinsä (ristin) ja tietokoneen toiminnan ei tarvinnut olla älykästä. Peli ei tule mennä rikki pienistä käyttäjän testauksista ja sitä tulee voida pelata useita pelejä kerrallaan.

Ratkaisuperiaate

Käytännössä pelin tuli toimia niin, että pelaajan laittaessa oman merkkinsä, tietokone laittaa myös omansa ennen kuin vuoro siirtyy taas pelaajalle, mikäli peli ei keskeytynyt voittoon tai siihen, että kaikki paikat on jo käytetty. Tämän kaiken tulee siis tapahtua yhden klikkauksen jälkeen. Lisäksi pelaaja voi aloittaa milloin tahansa uuden pelin lataamalla sivun uudestaan. Ohjelmani perustuu siis kahteen HTML-koodista tulevaan tapahtumaan, sivun lataamiseen ja pelaajan klikkaukseen pelialueella.

Sivun lataamisen aikana valmistelen pelilaudan niin, että taulukko on muuten tyhjä, mutta tietokoneen aloitusvuorona keskimmäisessä ruudussa on risti.

Kun pelaaja klikkaa pelitaulukon ruutua, HTML kutsuu javascriptin funktiota, jolla on useita toimintoja. Toiminnat ovat seuraavat:

1. Ensin tarkastetaan, että klikattu paikka on tyhjä. Mikäli näin ei ole, funktion muihin toimintoihin ei mennä ja klikkaajaa pyydetään valitsemaan tyhjä paikka.
2. Tämän jälkeen tarkistetaan, onko peli edelleen käynnissä, eli onko jompikumpi pelaaja voittanut (tai onko pelilauta täynnä, tämä toiminto ei vielä ole lisätty koodiin). Mikäli on, ilmoitetaan siitä pelaajalle ja pyydetään aloittamaan uusi peli.
3. Mikäli yllä olevien vaiheiden jälkeen peli on siis käynnissä ja klikattu paikka on tyhjä, vaihdetaan pelimatriisiin ja näytölle klikattuun paikkaan pelaajan merkki ’nolla’. Välittömästi tämän jälkeen tarkistetaan, toiko siirto voittoa pelaajalle ja mikäli tuli, ilmoitetaan tästä pelaajalle ja lopetetaan funktion läpikäynti tähän.
4. Mikäli siirto ei tuonut voittoa pelaajalle, etsitään tietokoneen merkille vapaa paikka ja merkataan se siihen. Tarkistetaan taas välittömästä toiko uusi siirto voittoa tietokoneelle ja ilmoitetaan tarvittaessa pelaajalle voitosta ja pelin päättymisestä. Mikäli voittoa ei tullut, jäädään odottamaan pelaajan uutta klikkausta.

Ohjelman käyttöohje

Ohjelman käyttö sujuu käytännössä näin: Pelaaja valitsee vapaan paikan omalle merkilleen ja klikkaa sitä. Varatun paikan klikkaamisesta ei tapahdu mitään, vaan pelaaja saa ilmoituksen valita vapaa paikka. Mikäli peli on päättynyt pelaajan tai tietokoneen voittoon, uusia siirtoja ei enää sallita, vaan pelaaja kehotetaan aloittamaan uusi peli lataamalla sivu uudelleen. Ohjelman tämänhetkinen puute on, ettei ohjelma ilmoita erikseen pelin päättymisestä silloin, kun pelilauta täyttyy ilman voittoa.

Ohjelman ja sen osien kuvaaminen

Ohjelmassa on kuusi funktiota, joista kaksi käynnistyy pelaajan toiminnan vaikutuksesta ja neljä toisen funktion kutsusta. Pelaajan kutsusta käynnistyy sivun lataamisessa pelilaudan alkujärjestely ja klikkauksesta peliä eteenpäin vievä ja johtava funktio. Tämä funktio on nimeltään Change(). Change() funktio kutsuu tarvittaessa kolmea funktiota, joista yksi (Computer()) kutsuu tarvittaessa vielä neljättä funktiota.

Change() funktion toiminta on jo ilmaistu melko seikkaperäisesti luvussa Ratkaisuperiaate.

Voiton tarkistamiseksi on oma funktio, joka käy läpi kaikki mahdolliset kolmen suorat ja tarkistaa, onko jommallakummalla merkillä oma suora. Tätä funktiota kutsutaan jokaisen siirron jälkeen ja lisäksi ennen siirtoja, jotta varmistutaan siitä, että peli päättyy voittoon eivätkä uudet siirrot ole enää voiton jälkeen mahdollisia.

Computer() -funktio arpoo tietokoneen risti-merkille satunnaisen paikan, ja sijoittaa sen siihen, mikäli se on vapaa. Jos paikka ei ole vapaa, ohjelma etsii automaattisesti seuraavan vapaan paikan matriisista ja laittaa merkin siihen.

Kaksi funktiota vaihtaa HTML-koodin aloituskuvan ristiksi tai nollaksi, niin että pelaaja voi seurata pelin tilannetta laitteensa ruudulta.

Testausjärjestely

Pelin toimivuuden testaamiseksi pelikoodiin on merkitty joitakin javascript-konsolin merkintöjä, joiden avulla voidaan seurata pelin toimimista oikein. Merkinnät kirjoittavan konsoliin pelin aikana tapahtuneita tapahtumia. Merkinnöistä voi tarkistaa pelimatriisin tilanteen ennen klikkausta, pelaajan siirron jälkeen ja tietokoneen siirron jälkeen. Lisäksi merkinnöistä käy ilmi, kuinka tietokone löysi vapaan ruudun ja mitä kohtaa pelaaja klikkasi.

Ohjelmaa on testattu seuraavasti:

1. Testattiin, ilmoittaako tietokone voiton kaikilla mahdollisilla suorilla sekä tietokoneelle että pelaajalle.
2. Testattiin, mitä käy, kun klikataan jo merkattua paikkaa.
3. Testattiin, loppuuko peli voittoon vai voiko uusia siirtoja tehdä vielä voitonkin jälkeen.
4. Testattiin, aloittaako sivun lataaminen uuden pelin.

Itsearviointi

Harjoitustyö tuntui aluksi todella vaikealta. Tuntui, etten osannut mitään ja koko ohjelman laatiminen vaati pelkkää kopioimista opettajan koodista. Mitä pidemmälle koodaus eteni, sitä enemmän koodiin alkoi ilmestyä myös sitä omaa kädenjälkeä ja omia ratkaisuja. Lopulta koodi tuntui ja näytti omalta, olen siitä ylpeä.

Harjoitustyössä helppoa oli jotkut pienet koodin osaset, jotka olivat jo hallussa. Esimerkiksi alert-toiminto, if-ketjutukset ja for-loopit. Loppuvaiheessa tajusin pystyväni ilmaisemaan itseäni jo aika hyvin pelkillä if-ketjuilla ja for-loopeilla. Kokonaisuudessaan kuitenkaan koodissa ei ollut kovinkaan montaa helppo kohtaa, vaan koodaus tuntui melko vaikealta joka kohdassa.

Ongelmia tuli monessa kohtaa ja monella tasolla. Ensinnäkin koko ohjelman järkevä toistojärjestys vaati paljon pohdintaa ja miettimistä. Mikä tapahtuu milloinkin ja missä välissä mitäkin funktiota kutsutaan ja mistä osista edes kannattaa tehdä funktio? Toinen päänvaiva oli monet koodin osat, joita en vaan osannut vielä tehdä. Opin esimerkiksi seuraavat toiminnot: miten HTML-koodista voidaan hakea elementti ja määrittää sen tyyppi joksikin tietyksi; miten if-ketjutuksesta voi hypätä pois kesken ketjun tai vastaavasti for-loopista, silloinkin kun for-loopeja on kaksi sisäkkäin; mistä kannattaa etsiä tietoa silloin, kun w3schools ei selvennä asiaa; miten aikaa vievä ja turhautumista aiheuttava joku pieni koodi voi olla.

Koodi jäi vielä vähän kesken, tulen varmaankin täydentämään sitä vielä ajan kanssa. Ominaisuuksia, joita en ehtinyt ajanpuutteen vuoksi lisäämään, on esimerkiksi ilmoitus pelin loppumisesta kun pelilauta on täynnä ja voittoa ei ole tullut (tiedän jo, miten sen teen, mutten vain enää ehdi). Lisäksi haluaisin kehittää tietokoneen logiikkaa tyhjän paikan etsimisessä. Tekoälyä tuskin pystyn siihen lähiaikoina lisäämään, mutta jokin vähän parempi toiminta-algoritmi ei varmasti ole mahdoton tehtävä.

Koodini toimii sujuvasti enkä ainakaan itse löydä siitä isompia virheitä tällä hetkellä. Itselleni antaisin 5/5 arvosanan siitä, että sain koodin toimimaan enkä luovuttanut kesken sekä siitä, että opin niin paljon uutta. Toisaalta koodissani ei ole kuitenkaan mitään erityistä jujua tai toimintoa, mikä erottaisi sen muista ja koen, että koodini voisi olla vielä 100 kertaa parempi itsenikin tekemänä. Näin ollen arvosanani tästä työstä on reilu 3 tai varovainen 4.