

SAVONIA



OPINNÄYTETYÖ - AMMATTIKORKEAKOULUTUTKINTO
TEKNIIKAN ALA

TEKOÄLYN VAIKUTUS OHJEL- MISTOKEHITYKSEEN

ChatGPT

TEKIJÄ Oskari Kärkkäinen

Koulutusala Tekniikan ja liikenteen ala	
Tutkinto-ohjelma Tietotekniikan tutkinto-ohjelma	
Työn tekijä Oskari Kärkkäinen	
Työn nimi Tekoälyn vaikutus ohjelmistokehitykseen, ChatGPT	
Päiväys 4.2.2025	Sivumäärä/liitteet 31
Yhteistyötaho Savonia-ammattikorkeakoulu	
Tiivistelmä <p>Työn tarkoituksena oli tutkia, kuinka tehokkaasti ChatGPT pystyy nykypäivänä teettämään koodia verrattuna ihmisen tekemään koodiin. Tätä tutkittiin antamalla tekoälylle ohjeet tehdä halutunlainen kokonaisuus kuten muistipeli. Ohjelmointikielenä käytettiin JavaScriptiä, jonka lisäksi dokumenttien suunnitteluun oli käytössä HTML:ää ja CSS-tyylejä. Ohjelmointisovelluksena oli käytössä Visual Studio Code.</p> <p>Muistipelin lisäksi selvitettiin kuinka tehokas tekoäly oli tunnistamaan virheitä sille annetusta koodin pätkästä sekä pystyykö se tarjoamaan potentiaalisen korjausehdotuksen siihen. Opinnäytetyön toimeksiantaja toimii Savonia-ammattikorkeakoulu.</p> <p>Teoriaosuudessa on puolestaan tutustuttu aiheen tiimoilta löytyviin artikkeleihin sekä tutkimuksiin. Teoriaosuudessa käydään läpi tekoälyn historiaa, kehitystä, ominaisuuksia sekä tulevaisuuden näkymää.</p> <p>Tutkimusosiosta saadut tulokset antoivat suoraan osviittaa siitä kuinka nopea, tarkka ja tehokas tekoäly on tuottamaan koodia verrattuna ihmiseen. Tekoälyn on todettu olevan oivallinen apuri koodin tuottamisessa, mutta koodari tarvitsee edelleenkin vahvan perusosaamisen koodin ymmärtämiseen ja tuottamiseen eikä tekoälyyn voida luottaa vain sokeasti. Tekoälystä on myös puhuttu, että se tulisi syrjäyttämään perinteisen koodarin työt jossain vaiheessa, mutta se milloin tekoäly siihen pystyy, on mysteeri vai pystyykö milloinkaan?</p>	
Avainsanat ChatGPT, tekoäly, ohjelmointi, JavaScript, HTML, CSS, Visual Studio Code	

SISÄLTÖ

1	JOHDANTO.....	5
2	MITÄ TEKOÄLY ON?	6
2.1	Tekoälyn historia ja kehitys	6
2.2	Tekoälyn tulevaisuus.....	7
2.3	Perustietoa ChatGPT:stä	8
2.3.1	ChatGPT:n versiot	9
2.3.2	ChatGPT:n käyttäjämäärät	10
3	CHATGPT OHJELMOIJAN SILMIN.....	11
3.1	Hyödyt ohjelmoinnissa	11
3.2	Haitat ohjelmoinnissa	11
3.3	Eettinen näkökulma.....	12
4	CHATGPT:N TESTAAMINEN	14
4.1	Muistipeli.....	14
4.1.1	HTML-rakenne (index.html)	15
4.1.2	CSS-tyylit (styles.css)	16
4.1.3	JavaScript-logiikka (scripts.js)	17
4.1.4	Muistipelin pelinäköymä.....	19
4.2	Uusien ominaisuuksien lisääminen muistipeliin	21
4.2.1	Päivitetyt koodit.....	22
4.3	Virheen löytäminen ja korjaaminen koodista.....	25
4.4	Tulosten analysoiminen.....	27
4.4.1	Muistipelin analyysi	27
5	POHDINTA.....	29
	LÄHTEET	30

KUVALUETTELO

Kuva 1. ChatGPT:n aloitus näyttö (ChatGPT, OpenAI, Kärkkäinen 2024)	8
Kuva 2. ChatGPT yksinkertainen kysymys (ChatGPT, OpenAI, Kärkkäinen 2024).....	9
Kuva 3. Miljoonan käyttäjän rajapyykki (ExplodingTopics, 2024).....	10
Kuva 4. ChatGPT:n hyödyt ja haitat kehittäjän näkökulmasta (Medium, 2023)	12
Kuva 5. Ohjeet muistipeliin (ChatGPT, OpenAI, Kärkkäinen 2024)	15
Kuva 6. Ohjeet projektin luomiseen (ChatGPT, OpenAI, Kärkkäinen 2024)	15
Kuva 7. HTML-koodi (Visual Studio Code, Kärkkäinen 2024).....	16
Kuva 8. CSS-koodi (Visual Studio Code, Kärkkäinen 2024)	16
Kuva 9. Scripts-tiedoston koodi osa 1 (Visual Studio Code, Kärkkäinen 2024).....	18
Kuva 10. Scripts-tiedoston koodi osa 2 (Visual Studio Code, Kärkkäinen 2024).....	19
Kuva 11. Muistipelin "pelilauta" (Kärkkäinen, 2024)	20
Kuva 12. Voitto ilmoitus (Kärkkäinen, 2024).....	20
Kuva 13. Häviö ilmoitus (Kärkkäinen, 2024).....	20
Kuva 14. Jatkokehitysidea (ChatGPT, OpenAI, Kärkkäinen 2024)	21
Kuva 15. Pelaaja määrän valinta ja nimien syöttäminen (Kärkkäinen 2024)	21
Kuva 16. Moninpelin voittaja ilmoitus (Kärkkäinen 2024).....	21
Kuva 17. "Pelilauta" monipelissä (Kärkkäinen 2024)	22
Kuva 18. Päivitetty HTML-koodi (Visual Studio Code, Kärkkäinen 2024)	22
Kuva 19. CSS-tiedostoon tehdyt päivitykset (Visual Studio Code, Kärkkäinen 2024)	23
Kuva 20. Päivitetty Scripts-tiedosto osa 1 (Visual Studio Code, Kärkkäinen 2024).....	23
Kuva 21. Päivitetty Scripts-tiedosto osa 2 (Visual Studio Code, Kärkkäinen 2024).....	24
Kuva 22. Päivitetty Scripts-tiedosto osa 3 (Visual Studio Code, Kärkkäinen 2024).....	24
Kuva 23. Päivitetty Scripts-tiedosto osa 4 (Visual Studio Code, Kärkkäinen 2024).....	25
Kuva 24. Päivitetty Scripts-tiedosto osa 5 (Visual Studio Code, Kärkkäinen 2024).....	25
Kuva 25. Virheet koodissa (Visual Studio Code, Kärkkäinen 2024)	26
Kuva 26. Virheen etsiminen (ChatGPT, OpenAI, Kärkkäinen 2024).....	26
Kuva 27. Korjaus ehdotukset (ChatGPT, OpenAI, Kärkkäinen 2024).....	27
Kuva 28. Muistipelin bugi (Kärkkäinen 2024)	28

1 JOHDANTO

Tekoäly on viime vuosina kehittynyt merkittävästi, ja sen soveltaminen eri toimialoilla on herättänyt laajaa kiinnostusta sekä huolta. Yksi keskeisistä osa-alueista, jossa tekoälyn vaikutukset ovat tulleet esille on ohjelmistokehitys.

Tässä opinnäytetyössä on tarkoituksena tutkia ChatGPT nimisen tekoälytyökalun kykyä tuottaa koodia, löytää koodista virheitä sekä selvittää kuinka hyvin tekoäly kykenee korjaamaan koodia. Opinnäytetyössä perehdytään tutkimaan tekoälyn (AI = artificial intelligence) tehokkuutta ohjelmoinnin näkökulmasta. Tutkimuksissa käytetään ChatGPT:n tarjoamaa ilmaisversiota, joka on avoin kaikille.

Tutkimusosion pääpainona on selvittää tekoälyn tuomat mahdolliset hyödyt sekä haitat ohjelmistokehitykselle. Tutkimusosion saadut tulokset antavat meille suoraan osviittaa siitä kuinka nopea, tarkka ja tehokas tekoäly on tuottamaan koodia verrattuna ihmiseen.

ChatGPT:n tarjoamia palveluja tutkimalla pyritään arvioimaan sen tuomia vaikutuksia ohjelmistotuotannon tuottavuuteen, tuotannon laatuun sekä tehokkuuteen. Näitä asioita selvittämällä saadaan lisää tietoisuutta tekoälyn asemasta ohjelmistotuotannossa nykytilanteessa sekä sen tarjoamista mahdollisuuksista tulevaisuudessa ohjelmistotyöntekijöille.

ChatGPT:n tehokkuutta testaavilla tehtävillä analysoidaan tekoälyn kyvykkyyttä pärjätä ohjelmoinnissa. Testauksia on kolmea erilaista, nämä ovat koodin tuottaminen pyydetyn tehtävänannon mukaisesti, virheiden tunnistaminen- ja korjaaminen koodista sekä jatkokehittäminen. Lopuksi analysoidaan tuloksia kuten nopeus ja täsmällisyys.

Tutkimuksessa halutaan saada vastauksia pariin kysymykseen, jotka ovat

1. Kuinka tehokas ja hyödyllinen työkalu ChatGPT on ohjelmatuotannossa?
2. Miten ChatGPT vaikuttaa ja tulee vaikuttamaan ohjelmistotuotantoon tulevaisuudessa?

2 MITÄ TEKOÄLY ON?

Tekoäly (AI) on teknologia, joka on mahdollistanut koneiden simuloida ihmisen oppimista, ymmärrystä, ongelmanratkaisua ja päätöksentekoa. Nykyään erityisesti niin sanottu generatiivinen tekoäly, kuten ChatGPT pystyy tuottamaan tekstiä, kuvia sekä suorittamaan tehtäviä. Koneoppiminen on osa tekoälyä. Käytännössä tämä tarkoittaa, että tekoäly oppii sille annetusta datasta ja osaa rakentaa vastauksia perustuen saamiinsa tietoihin. Toinen koneoppimisen osa-alue on syväoppiminen, joka perustuu puolestaan neuroverkkoihin, joka mahdollistaa monimutkaisempien kuvioiden ja tietorakenteiden tunnistamisen. (IBM, 2024)

Tekoälyllä on todettu olevan paljon positiivisia vaikutuksia esimerkiksi työpaikoilla ja arjessa, mutta myös negatiivisia vaikutuksia on pystytty havaitsemaan. Tekoäly parantaa työn tehokkuutta sekä tarkkuutta, automatisoi rutiinitehtäviä kuten asiakaspalvelua chatbottien muodossa. Tällä hetkellä tekoäly on oiva apuri ja helpottava tekijä, mutta sen kehittyessä entisestään voi olla olemassa vaara sen viedä ihmisten työt kokonaan. (Techtarget, 2024)

2.1 Tekoälyn historia ja kehitys

Tekoälyn historian esiasteleerit ulottuvat 1940-luvulle, jolloin tieteen pioneereina tunnetut Warren McCulloch ja Walter Pitts pistivät jalkeille ensimmäiset versiot neuroverkoista. Heidän antamaansa työpanosta pidetään pohjana ihmisaivojen toiminnan mallintamista koneelliseksi eli tekoälyksi, joka on ollut keskeinen tekijä tekoälytutkimuksissa. Tekoälyn alku askeleilla on ollut myös suomalaista osaamista, sillä Teuvo Kohonen toi oman panoksensa neuroverkkojen kehittämiseen ja näin ollen edisti tekoälyn kehitystä kansainvälisellä tasolla. (Tampereen yliopisto, 2022)

Vaikka tekoäly otti ensiaskeleensa 1940-luvulla, pidetään kuitenkin vuotta 1956 enemmänkin tekoälytutkimuksen alkuna. Tästä vuodesta mullistavan teki Dartmouthissa pidetty konferenssi, jonne kokoontuivat kaikki sen hetkiset alan tutkijat, tällöin käytettiin myös ensimmäisen kerran termiä ”tekoäly”. Tästä vain neljä vuotta eteenpäin eli vuonna 1960 saatiin jalkeille ensimmäisiä tekoälyä hyödyntäviä sovelluksia, jotka olivat aikansa huipputeknologiaa. Tunnetuin näistä oli Joseph Weizenbaumin kehittämä ELIZA-ohjelma. ELIZA pystyi simuloimaan yksinkertaista vuorovaikutusta ihmisen kanssa, ja sitä pidetään yhtenä ensimmäisistä kokeiluista luonnollisen kielen käsittelyssä. (Tampereen yliopisto, 2022)

Tekoälytutkimus alkoi kuitenkin hieman myöhemmin kohdata haasteita ja tämä johti siihen, että 1970-luvun lopulla tutkimusrahoitusta vähennettiin. Projektien vaikeus oli myös arvioitu pahasti alakanttiin. Tästä seurasi tekoälyn kehityksessä vaikeat ajat, joista käytettiin nimitystä ”tekoälyn talvet”. (Tekoäly.info, 2023)

1990-luvulla tietokoneiden laskentateho kasvoi huomattavasti, mikä mahdollisti monimutkaisempien tekoälymallien kehittämisen. Tällöin tehtiin myös historiaa sillä IBM Deep Blue nimisen tietokoneen voittoa shakkimestari Garri Kasparovista pidettiin uskomattomana. Tämä hetki todisti tekoälyn voiman ja lisäsi uskoa tutkijoiden silmissä tekoälyn mahdollisuuksista. (Coursera, 2024)

1990-luvun puolella välissä lisääntynyt GPU-laskenta (Graphics Processing Unit) on ollut suuri tekijä neuroverkkojen kouluttamisessa sekä suurien datamäärien käsittelyyn, jonka myötä ensimmäiset 3D-lisäkortit saatiin markkinoille. Alun perin GPU:t olivat suunniteltu grafiikoiden renderöintiin, josta ne muovautuivat tehokkaiksi rinnakkaislaskennan välineiksi. (Techspot, 2023)

2000-luvulla internetin ja suurten datamäärien saatavuus edisti koneoppimisen ja syväoppimisen tutkimusta merkittävästi. Tekoälyn sovellukset laajenivat eri aloille, ja koneoppimisen ansiosta kehitettiin puheentunnistuksen ja kuvantunnistuksen kaltaisia teknologioita. (Coursera, 2024)

2010-luvulla eli nykyajassa tekoälyn kehitys otti uuden suunnan generatiivisten mallien, kuten ChatGPT ja Googlen BERT myötä. Näiden kehittyneiden neuroverkkojen ansiosta tekoäly pystyy tuottamaan uudenlaista sisältöä, kuten tekstiä ja kuvia, sekä osallistumaan monimutkaisiin vuorovaikutustilanteisiin ihmisten kanssa. Tänä päivänä tekoälyä hyödynnetään laajasti eri sektoreilla, kuten terveydenhuollossa, teollisuudessa ja kuluttajateknologiassa, ja sen rooli yhteiskunnassa kasvaa jatkuvasti. (Coursera, 2024)

2.2 Tekoälyn tulevaisuus

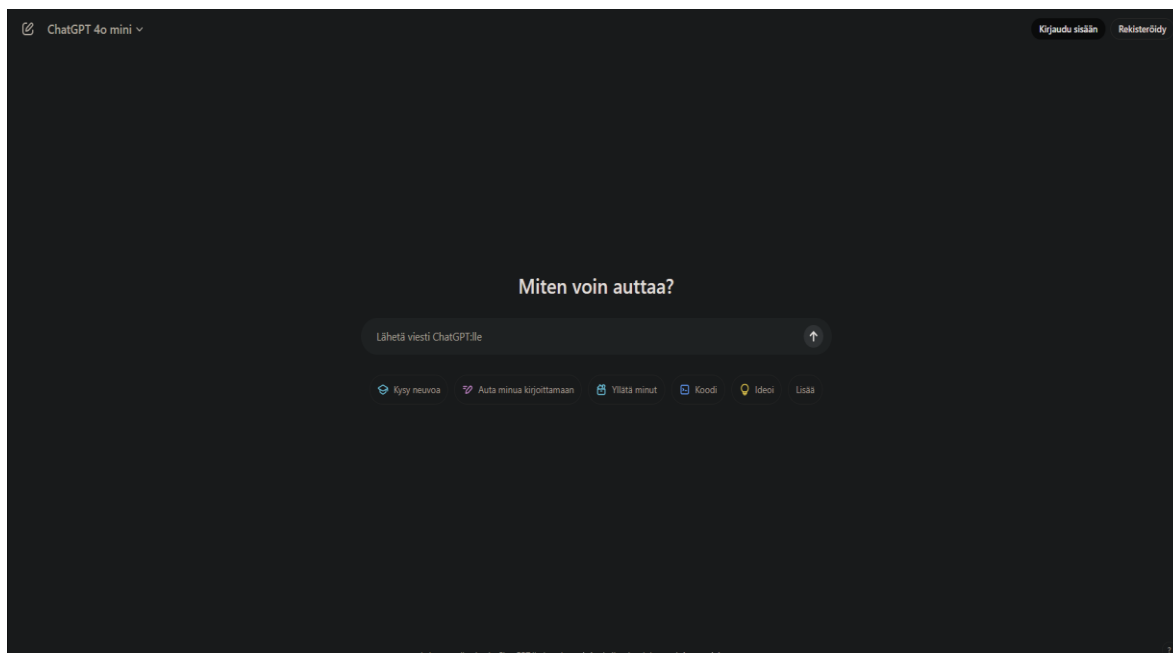
Tekoälyn tulevaisuus on meille kaikille vielä täysi mysteeri, mutta tutkijat ovat antaneet omia näkemyksiään asiasta. Tekoälyn povataan muuttavan yhteiskuntaa sekä liiketoimintaa merkittäväällä tavalla seuraavaan viiden vuoden aikana, ennusteiden mukaan tekoäly tulee pian olemaan luonnollinen osa jokapäiväistä elämäämme. Se mitä monet pitävät tekoälyn tulevaisuutena on jo todellisuudessa monien yritysten arkea, mutta pian tämän kuitenkin odotetaan olevan myös osana ihmisten jokapäiväistä toimintaa. Voidaan kerätä kuivakaapin ja jääkaapin sisältö pöydälle ja ottaa siitä kuva, jonka jälkeen tekoäly kertoo meille mitä ruokia näistä raaka-aineista voitaisiin valmistaa. (Sulava. 2024, 25. kesäkuuta)

Tekoälyn tulevaisuudesta puhuttaessa osa ihmisistä on paljon skeptisempiä sen suhteen toisin kuin valtaosa, mutta kaikesta huolimatta kaikkien on hyväksyttävä fakta, että tekoäly on tullut jäädäkseen ihmisten elämään. Työpaikkoja tullaan menettämään, mutta kukaan ei vielä tiedä paljonko ja millä aikataululla. Tästä on kuitenkin oltava tietoinen ja osattava valmistautua siihen. Joidenkin osalla saattaa tulla tarpeen etsiä jopa täysin uusi urapolku. (Sulava. 2024, 25. kesäkuuta)

Vaikka tekoälyn tulemisen myötä onkin paljon ennusteita työpaikkojen vähenemisestä, niin on huomioitava myös se, että se tulee samalla avaamaan uusiakin työllisyysmahdollisuuksia. Tekoälyn ymmärtäminen vaatii ihmiseltä myös taitoa. Eniten mahdollisuuksia tekoälyn myötä on povattu terveysalalle, jossa voitaisiin saada tulevaisuudessa tarkempia diagnooseja ja tehokkaampaa hoitoa. Lisäksi koulutuksen laadun on ennustettu paranevan tulevaisuudessa tekoälyn myötä. Koulutuksen on ennustettu olevan tehokkaampaa, monipuolisempaa ja se voidaan räätälöidä tarkemmin tukemaan yksilön persoonan sekä kykyjen mukaan. (Simplilearn, 2024)

2.3 Perustietoa ChatGPT:stä

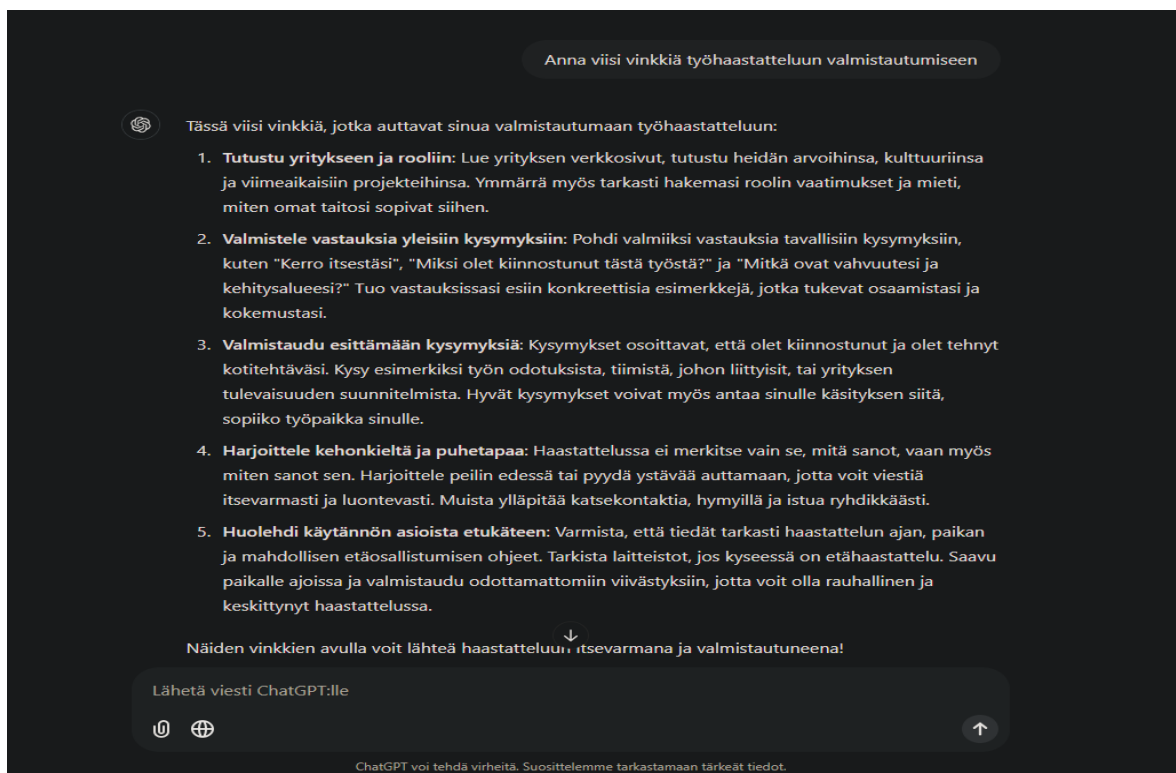
ChatGPT on vuonna 2022 päivänvalon nähnyt tekoälyyn perustuva chatbot, joka on OpenAI:n kehittämä. ChatGPT:tä voidaan nimittää vuorovaikutusohjelmistoksi, sillä sen toiminta perustuu niin sanottuun NLP:hen (Natural Language Processing), eli luonnollisen kielen prosessi. Käytännössä tämä tarkoittaa, että käyttäjä voi kirjoittaa ChatGPT:lle aivan kuin muihinkin chatteihin, sillä se osaa tulkita ihmisen kirjoittamaa tekstiä sellaisenaan. (Koulutus.fi, 2023)



Kuva 1. ChatGPT:n aloitus näyttö (ChatGPT, OpenAI, Kärkkäinen 2024)

Kuten muutkin tekoälyllä varustetut chatbotit niin ChatGPT:kin oppii sen käyttäjän kanssa käydyistä keskusteluista ja täten osaavat hyödyntää aikaisempien keskustelujen tietoja seuraaviin vastauksiin. Käytännössä ohjelma on siis suunniteltu antamaan vastauksia sille annettujen tietojen perusteella. (Koulutus.fi, 2023)

Käyttäjän tilanteessa tämä tarkoittaa siis, että ChatGPT:n tekstikenttään kirjoitetaan kysymys, johon halutaan saada vastaus, jonka jälkeen tekoäly analysoi kysymyksen sana sanalta, jonka jälkeen se rakentaa vastauksen sille opetetun datan perusteella. Tekoälyn idea piilee siinä, että se kykenee oppimaan käyttäjästä ja täten se kykenee antamaan vastauksia koko ajan tarkemmin ja tarkemmin. Se kykenee myös vastaamaan sille esitettyihin jatkokysymyksiin sekä osaa myöntää virheellisesti annetun tiedon, jos käyttäjä havaitsee virheellistä tietoa. (Koulutus.fi, 2023)



Kuva 2. ChatGPT yksinkertainen kysymys (ChatGPT, OpenAI, Kärkkäinen 2024)

2.3.1 ChatGPT:n versiot

ChatGPT:n ensimmäinen versio oli marraskuussa 2022 julkaistu GPT-3.5, joka oli ensimmäinen koskaan julkaistu chat-ympäristö, jossa oli mahdollista keskustella tekoälyn kanssa luonnollisella kielellä. ChatGPT 3.5 oli kaikille käyttäjille avoin ja ilmainen, myöhemmässä vaiheessa julkaistaan maksullisia versioita, jotka vaativat plus tilauksen. GPT-3.5:n käyttäminen vaatii vain kirjautumista, joka onnistuu google-tunnuksilla, Microsoft-tilillä tai Apple-tunnuksilla (OpenAI, 2022)

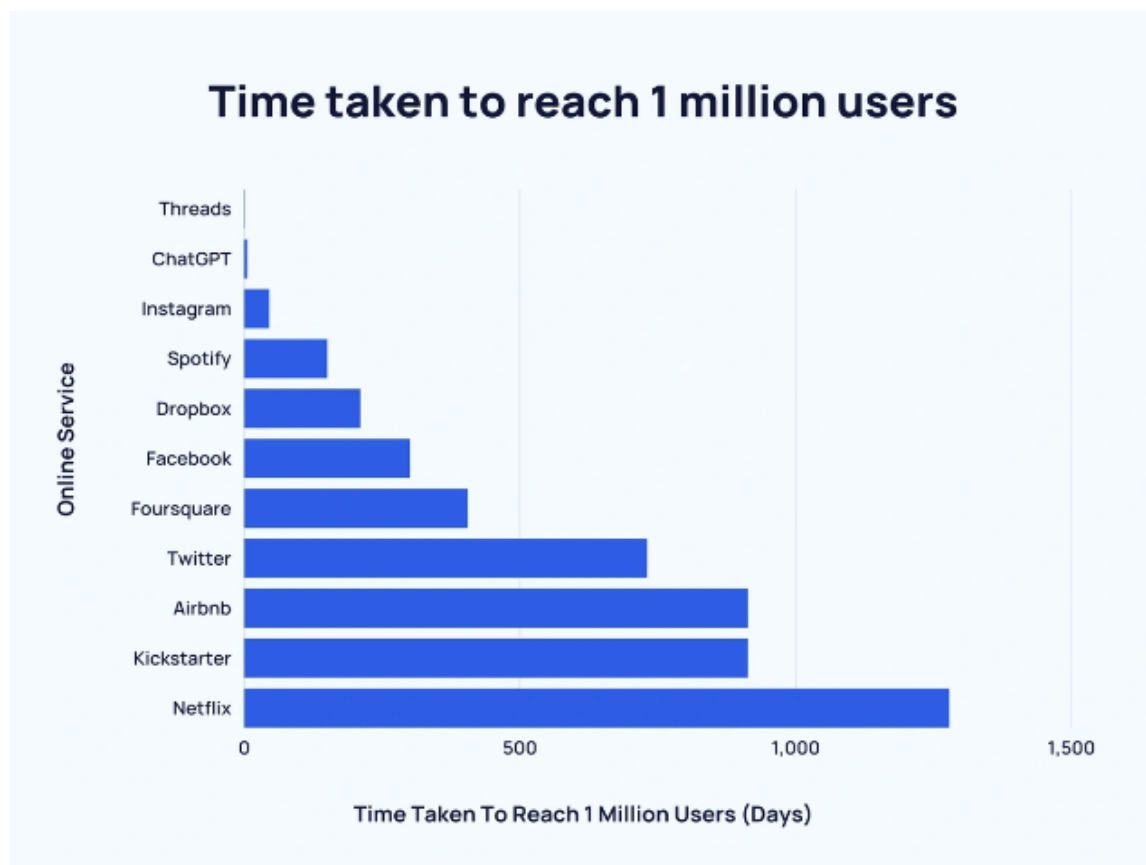
Vuonna 2023 julkaistiin kaksi seuraavaa versiota ChatGPT:stä, joita olivat GPT-3.5 turbo sekä GPT-4. GPT-3.5 turbo tarjosi käyttäjälleen täysin samat palvelut kuin edeltäjänsäkin, mutta vastauksia olivat hieman tarkempia ja nopeampia, mikä paransi jo huomattavasti käyttömukavuutta. (OpenAI, 2023) GPT-4 malli puolestaan oli ensimmäinen versio, joka ei ollut kaikille käyttäjille avoin, vaan se edellytti maksullista plus tilausta. GPT-4 oli kuitenkin selkeä harppaus parempaan, sillä se ymmärsi selkeästi monimutkaisempia kysymyksiä ja pystyi analysoimaan sille annettuja tiedostoja sekä kuvia. (OpenAI, 2023)

2024 lanseerattiin ChatGPT:n GPT-4-turbo versio, joka oli taas hieman nopeampi ja tehokkaampi kuin edeltäjänsä, se tarjoaa vastauksensa suoraviivaisemmin, sekä tarjoaa ratkaisuja haastavimpiin tehtäviin. Lisäyksenä turbo versio pystyy käsittelemään kuvien lisäksi myös videoita ja ääniraitoja. (OpenAI Platform, 2024) Pari kuukautta tämän jälkeen oli taas uutta versiota saatavilla nimittäin GPT-4o mini näki päivänvalon, joka oli tarkoitettu nopeisiin yksinkertaisiin tehtävääntoihin. Tämän version käyttötarkoitus oli siis kohdennettu arkisiin tehtävääntoihin eikä niinkään vaativiin tehtäviin, mutta pystyy samalla kuitenkin käsittelemään suuria määriä dataa. (OpenAI, 2023)

Uusimmat versiot, jotka on julkaistu ovat tällä hetkellä käytössä olevat o1-preview sekä o1-mini, jotka julkaistuun syyskuussa 2024. Molemmissa versioissa on käyttörajoitukset, jotka ovat minissä 50 kysymystä päivää kohden ja preview versiossa puolestaan 50 kysymystä viikkoa kohden. Nämä versiot ovat suunnattu monimutkaisten ongelmien ratkaisuun, kuten ohjelmointiin, fysiikkaan, kemi-
aan, biologiaan ja matematiikkaan liittyvät tehtävänannot. Erityisesti o1-mini on suunnattu enemmän ohjelmointia silmällä pitäen, sillä se on testausten perusteella todettu olevan tehokas tuottamaan koodia sekä korjaamaan virheitä. Näiden versioiden toimintaperiaatetta on pyritty tekemään vielä enemmän simuloimaan ihmisen toimintaa, sillä tekoäly on koulutettu ”ajattelemaan” kysymys tar-
kemmin läpi ennen kuin antaa vastauksen. (OpenAI, 2023)

2.3.2 ChatGPT:n käyttäjämäärät

ChatGPT julkaistiin vuonna 2022 ja julkaisun myötä sen käyttäjämäärä lähti saman tien räjähdys-
mäiseen nousuun rikkoen kaikki aikaisemmat ennätykset. Miljoonan käyttäjän rajan se pystyi rikko-
maan uskomattoman nopeasti vain viisi päivää julkaisun jälkeen. Viisi päivää on todella vähän, esi-
merkiksi Instagramilla meni vastaavaan määrään 2,5 kuukautta ja Netflixillä 3,5 vuotta. Nykyisillään
ChatGPT:n kuukausi käyttäjämäärän on raportoitu olevan noin 207 miljoonaa käyttäjää kuukautta
kohden. (Exploding Topics, 2024)



Kuva 3. Miljoonan käyttäjän rajapyykki (ExplodingTopics, 2024)

3 CHATGPT OHJELMOIJAN SILMIN

ChatGPT:n luomat mahdollisuudet sekä haasteet ovat olleet kuuma puheenaihe työpaikoilla ja siitä mitä se tuo tullessaan, mutta erityisen paljon se puhuttaa tietotekniikan sarakkeella ja tarkemmin sanottuna koodareiden keskuudessa. Tekoälyn tehokkuus on havaittu koodin tuottamisessa, mutta virheetöntä se ei kuitenkaan ole. Tekoäly tarvitsee koodarilta edelleenkin tuekseen vahvaa ymmärrystä siitä mitä tehdään ja mitä halutaan tehdä, täten koodarin työtä tarvitaan edelleenkin. (Yle, 2023)

3.1 Hyödyt ohjelmoinnissa

ChatGPT:n kaltaisen tekoälysovelluksen hyötyjä on todettu olevan kuitenkin runsaasti. Tekoälyn avulla voidaan nykyään hoitaa rutiininomaisia tehtäviä alta pois, joka antaa ihmisille mahdollisuuksia panostaa enemmän luoviin ja päättelyä vaativiin tehtäviin. Tällä tavalla voidaan vaikuttaa tuottavuuteen positiivisessa mielessä. (Yle, 2023)

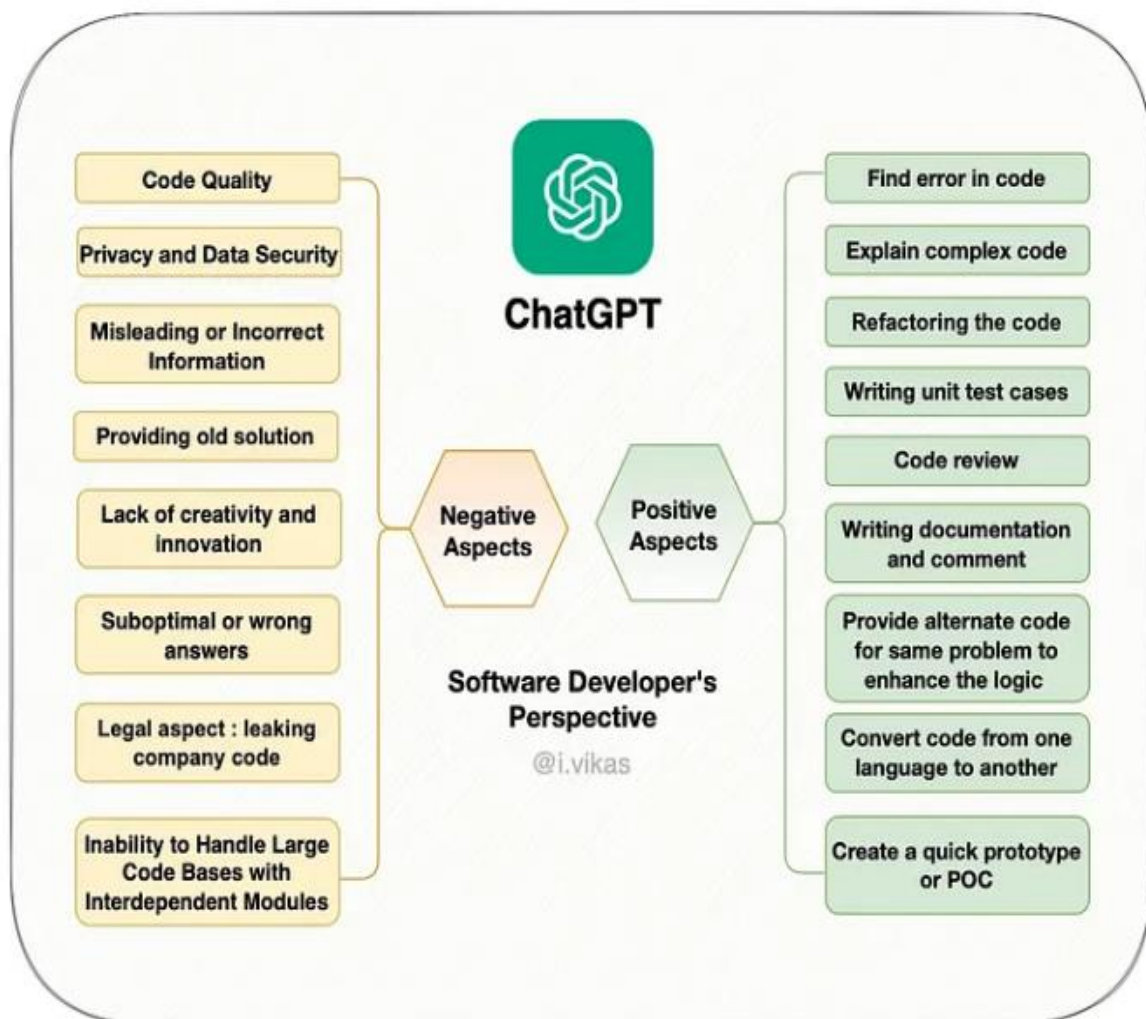
Tekoälyn osaaminen perinteisessä koodaamisessa ei rajoitu suinkaan niin sanottuihin perustoimintoihin vaan tekoäly kykenee monimutkaisempiinkin kokonaisuuksiin, kuten luokkien ja moduulien rakentamiseen. Tämä helpottaa vähemmän koodanneita pääsemään alkuun ilman, että joudutaan käyttämään paljon tunteja alkeellisiin asioihin. Koodin "siistimiseen" ChatGPT on myös ollut oiva apuri, sillä se kykenee muokkaamaan koodia yksinkertaisempaan muotoon melko tehokkaasti. Tällöin saadaan koodi helppolukuisemmaksi, mutta toiminnallisuudet säilyvät kuitenkin ennallaan (TechRepublic, 2023)

Yleinen näkemys virheiden korjaamisesta koodista kehittäjien kesken on varsin negatiivinen. Virheiden korjaaminen osaa olla varsin hermoja raastavaa ja aikaa vievää puuhaa. Monesti näissä tapauksissa virhe on vain pieni piste tai pilkku virhe, joka ei satu osumaan silmään. ChatGPT tarjoaa tähän kuitenkin suurta helpotusta ja täten säästytään paljolta tuskailulta. Tekoäly kykenee käymään läpi nopeassa ajassa jopa tuhansia rivejä koodia etsiessään virhettä, tällaiseen ihmisen ei vain yksinkertaisesti ole mahdollista kyetä. (TechRepublic, 2023)

3.2 Haitat ohjelmoinnissa

Vaikka ChatGPT tarjoaa meille varsin valideja vastauksia ja päällisin puolin oikealta näyttävää koodia, pitää silti muistaa, ettei tekoäly ole erehtymätön. Tekoälyn tarjoamat vastaukset ovat välillä uskottavan ja hyvän kuuloisia, mutta tarkemman tarkastelun jälkeen voi paljastua, että pyydetty vastaus on täysin vääränlainen tai jopa aivan järjetön. Vastaan voi tulla koodia eri ohjelmointikielellä mitä pyysit tai kaikkia pyydettyjä ominaisuuksia ei ole huomioitu, tämän takia vastaukset on syytä katsoa tarkasti läpi. (TechRepublic, 2023)

Tekoälyn tuottama koodi sisältää myös negatiivisia puolia, jotka eivät välttämättä pomppaa ensimmäisenä mieleen näitä ovat mm. yksityisyys ja tietoturva, virheelliset tiedot, laatu sekä vanhojen ratkaisujen tarjoaminen. Näistä suurimmaksi puheen aiheeksi kehittäjien keskuudessa on noussut juuri koodin laatu. Ongelmana tässä piilee, että ChatGPT pystyy kyllä tuottamaan koodia, joka toimii niin kuin pitääkin, mutta koodi on monesti sellaista, että se sisältää turhaa toistoa. Vaikka asia olisi mahdollista tehdä yksinkertaisemmassa muodossa. Näiden myötä virheiden korjaaminen ja koodin tarkastaminen ottaa enemmän aikaa. ChatGPT:n tarjoamat vastaukset voivat olla myös paikkaansa pitämättömiä, koska tietokanta, josta se etsii tietonsa ei voida pitää mitenkään kokoaikaa ajan tasalla (Vikash, 2023)



Kuva 4. ChatGPT:n hyödyt ja haitat kehittäjän näkökulmasta (Medium, 2023)

3.3 Eettinen näkökulma

Tekoälyn eettinen näkökulma on monimutkainen asia pohtia. Eettisyys liittyy oikeaan ja väärään, mutta miten se huomioidaan tekoälyn suhteen? Tekoälyn maailmassa perinteiset kysymykset liittyen oikeellisuuteen saa uudenlaista lähestymiskulmaa. Tämä johtuu täysin tekoälysovellusten tuomasta autonomisuudesta sekä adaptiivisuudesta. Tämä tuo meille kysymysmerkkejä ja keskustelua sovellusten toiminnan ennustettavuudesta sekä ihmiskontrollista. (Tampereen yliopisto, 2022)

Tekoälyn etiikka keskittyy pääosin siihen, miten kehittäjät, valmistajat, viranomaiset ja käyttäjät toimivat sen kehittämisen ja käytön aikana. Näiden toimijoiden vastuullisella käyttäytymisellä pyritään minimoimaan eettiset riskit ja luomaan kestävä pohja tekoälyteknologioiden käytölle. ChatGPT:n eettisyys on ollut laajan keskustelun aihe, herättäen niin puolustajien kuin kriitikoidenkin huomion. Vaikka ChatGPT on neutraali työkalu, sen käyttöön liittyy sekä hyötyjä että haasteita. (Moxilearn, 2024)

ChatGPT:n etuja ovat esimerkiksi tuottavuuden kasvu, jatkuva oppiminen ja kyky mukautua monenlaisiin käyttötarkoituksiin työssä, opiskelussa ja arjessa. Toisaalta huolenaiheita ovat tietosuojongelmat, mahdollinen väärinkäyttö ja liiallinen riippuvuus teknologiasta. Vaikka ChatGPT on suojattu tietomurtoja vastaan, riskiä ei voida täysin poistaa. Lisäksi on tärkeää säilyttää kriittisyys ChatGPT:n tuottamia vastauksia kohtaan, sillä kielimallit voivat toisinaan tuottaa virheellistä tai harhaanjohtavaa tietoa. Tämä korostaa käyttäjien vastuuta arvioida tekoälyn antamaa tietoa ja hyödyntää sitä harkiten. ChatGPT:n kaltaisten työkalujen tehokas ja turvallinen käyttö edellyttää, että käyttäjät ymmärtävät sen toimintaperiaatteet ja eettiset ulottuvuudet. (Moxilearn, 2024)

4 CHATGPT:N TESTAAMINEN

Tässä osiossa suoritetaan ChatGPT:lle muutama erilaatuinen testaus. Testauksilla saamme vastauksia kuinka täsmällinen, nopea ja tehokas tekoäly on suoriutumaan tehtävistä. Tehtävissä käytämme JavaScriptiä, HTML:ää sekä CSS:ää. Tehtäviä on kolmeen eri lähtöön, eli muistipelin luominen, virheen löytäminen koodista ja sen korjaaminen ja muistipelin jatkokehittäminen.

Ensimmäisessä testissä testataan kuinka tehokkaasti ja onnistuneesti tekoäly onnistuu luomaan pyydetynlaisen muistipelin. Tekoälylle annetaan tarkat ja yksityiskohtaiset ohjeet muistipelin luomiseen. Tässä testissä on tarkoitus selvittää kuinka hyvin tekoäly pystyy noudattamaan sille annettua tehtävänantoa ja saako se luotua pyydetynlaisen toimivan kokonaisuuden.

Testissä numero kaksi testaan tekoälyn taitoa löytää virheitä valmiista koodista, jonne on upotettu tarkoituksenmukaisesti muutama virhe sekaan. Tällä testillä halutaan selvittää tekoälyn nopeutta ja kyvykkyyttä tunnistaa virheitä koodista, sekä katsoa millaisia korjausehdotuksia se tarjoaa. Lisäksi testaamme samassa koodin korjaamisen tehokkuutta, jossa tekoälyä pyydetään kirjoittamaan koodista korjattu versio. Testissä halutaan selvittää onnistuuko tekoälyltä koodin korjaaminen ja tuleeko korjauksista ylipäätään kelpaavia.

Viimeisenä kokeilemme miten tekoäly pystyy jatkokehittämään itse luomaansa muistipeliä. Tekoälylle annetaan muutama idea mitä pelin ominaisuuksiin voisi lisätä ja katsotaan, onnistuuko pyydettyjen ominaisuuksien lisääminen.

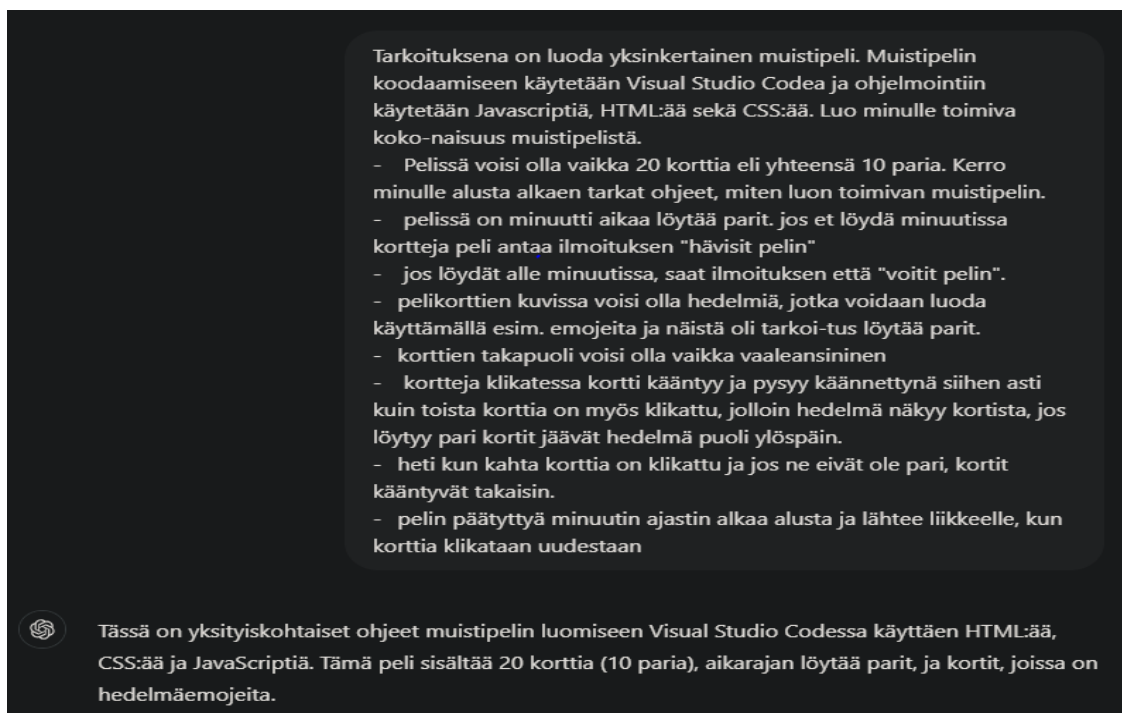
4.1 Muistipeli

Muistipelin luomiseen annettiin tekoälylle erittäin tarkat ohjeet pelin luomiseen. Tarkoituksena oli selvittää pystyykö tekoäly seuraamaan ohjeita tarkasti ja täten tuottamaan toimivan kokonaisuuden. Ohjeissa määriteltiin mm. koodaamiseen käytettävä ohjelma, ohjelmointikieli, pelikorttien määrä, korttien kuvat, peliaika sekä korttien takapuolen kuviointi.

Muistipelin on tarkoitus olla pelattavissa tietokoneella paikallisesti. Online-pelimahdollisuutta peliin ei ole aikomuksena toteuttaa. Myöskään pelisuorituksia ei kirjata minkäänlaiseen tulostaulukkoon, sillä pääpainona on pelin toiminnallisuuksien onnistuminen.

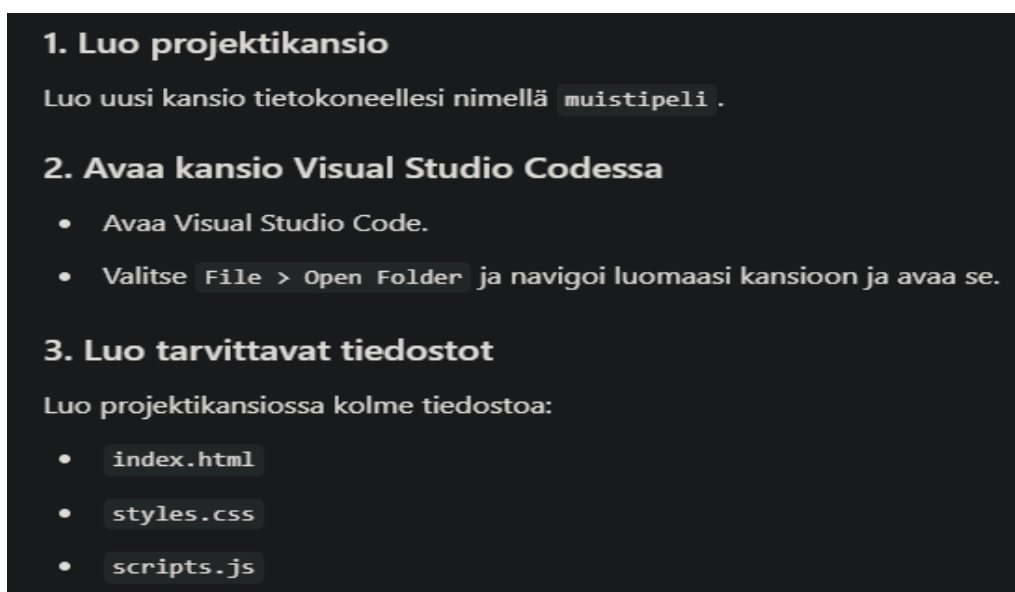
ChatGPT alkoi saman tien kirjoittamaan koodia ja antoi tarkat ohjeet ihan projektin luomisesta lähtien. Tekoälylle meni noin 30–45 sekuntia luoda meille muistipeli. Ihmisellä menisi vastaavanlaisen tuotoksen luomiseen arviolta 1–2 työpäivää. Koodi oli suoraan kopioitavissa ChatGPT:n tekstikentästä, joten sen liittäminen Visual Studioon ei vienyt aikaa kuin pari minuuttia.

Myöhemmässä vaiheessa kokeilemme vielä hieman jatkokehittää muistipeliä eli testaamme, pystyykö ChatGPT tekemään jälkikäteen pyydettyjä ominaisuuksien lisäyksiä.



Kuva 5. Ohjeet muistipeliin (ChatGPT, OpenAI, Kärkkäinen 2024)

ChatGPT antoi ohjeet muistipelin luomiseen aivan juurta jaksuen. Ohjeet alkoivat projektin luomisesta. Näin tarkkojen ohjeiden saatua, ei tarvitse käyttäjällä periaatteessa olla juuri ollenkaan kokemusta edes Visual Studio Coden käyttämisestä, sillä ohjeet ovat niin selvät ja yksityiskohtaiset.



Kuva 6. Ohjeet projektin luomiseen (ChatGPT, OpenAI, Kärkkäinen 2024)

4.1.1 HTML-rakenne (index.html)

ChatGPT:n luoma koodi muistipelille oli varsin selkeää luettavaa. Koodiin oli sisällytetty myös kommentointia, vaikka tätä ei pyydetty. Kommentointi helpottaa koodirakenteen kokonaisuuden ymmärtämistä ja on suuri apu jatkokehityksessä.

Ensimmäisenä ChatGPT generoi koodin HTML-tiedostolle. "Index.html" koodia voidaan nimittää muistipelin rungoksi, joka luo pelille niin sanotusti rakenteellisen perustan ja yhdistää visuaaliset tyylit sekä interaktiivisen logiikan. Ilman HTML-tiedoston luomista pelillä ei olisi visuaalista näkymää eikä käyttöliittymää.

Tässä selitettynä muutamia tärkeimpiä ominaisuuksia liittyen index.html tiedoston koodiin:

1. "<div class="timer">" näyttää pelin ajastimen, joka informoi käyttäjää jäljellä olevasta ajasta. span-elementtiä käytetään ajastimen sekuntien dynaamiseen päivittämiseen JavaScriptin avulla.
2. "<script src="scripts.js"></script>" funktio, joka linkittää sen pelin JavaScript-tiedostoon, joka sisältää puolestaan pelin logiikan.
3. "<section class="game-board"></section>" määrittelee alueen, jolle pelikortit renderöidään dynaamisesti JavaScriptin avulla. Tämä elementti toimii "pelilautana".

```

<!DOCTYPE html>
<html lang="fi">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Muistipeli</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <h1>Muistipeli</h1>
  <div class="timer">Aikaa jäljellä: <span id="timer">60</span> sekuntia</div>
  <section class="game-board"></section>
  <script src="scripts.js"></script>
</body>
</html>

```

Kuva 7. HTML-koodi (Visual Studio Code, Kärkkäinen 2024)

4.1.2 CSS-tyylit (styles.css)

Seuraavaksi on käsittelyssä styles.css-tiedosto, joka määrittää muistipelin visuaalisen ilmeen. Toisin sanoen se määrittää käyttäjäkokemuksen. CSS-tiedoston avulla määritellään mm. muistipelin värit, fontit sekä elementtien sijoittelut "pelilaudalle". Lisäksi CSS-tiedostossa voidaan muokata korttien kääntymiseen vaikuttavia animaatioita, korttien kokoa sekä pelialueen kokoa, jotka vaikuttavat suoraan käyttäjän samaan kokemukseen pelistä.

```

body {
  font-family: Arial, sans-serif;
  display: flex;
  flex-direction: column;
  align-items: center;
  height: 100vh;
  margin: 0;
  background-color: #f4f4f9;
}

.game-board {
  display: grid;
  grid-template-columns: repeat(5, 120px);
  grid-gap: 10px;
  margin-top: 20px;
}

.card {
  width: 120px;
  height: 120px;
  background-color: #add8e6; /* Vaaleansininen */
  display: flex;
  justify-content: center;
  align-items: center;
  font-size: 48px; /* Suuri emoji */
  cursor: pointer;
  border-radius: 10px;
  position: relative;
}

.card.flipped {
  background-color: white; /* Kääntyessä tausta vaihtuu valkoiseksi */
}

.timer {
  font-size: 24px;
  margin: 20px;
}

```

Kuva 8. CSS-koodi (Visual Studio Code, Kärkkäinen 2024)

4.1.3 JavaScript-logiikka (scripts.js)

Muistipelin kolmas ja viimeinen vaadittava kooditiedosto on scripts.js. Se on JavaScript-koodi, joka hallitsee pelin logiikkaa ja interaktiivisuutta. Tämä mahdollistaa käyttäjän syötteiden käsittelyn, pelitilan ylläpidon ja visuaalisten muutosten toteuttamisen reaaliajassa. Scripts.js tiedostoon sisältyy mm. funktiot sekä koodi, joka hallitsee pelin sisällä tapahtuvaa logiikkaa eli pelikorttien kääntämisen, parien tarkistamisen, ajanoton ja lopputuloksen käsittelyn.

JavaScript logiikka on scripts.js tiedostossa koostuu kuudesta eri funktiosta, joista jokainen toteuttaa oman toiminnallisuutensa peliin. Funktioita ovat:

- **shuffleArray** = Tehtävä on sekoittaa muistipelin korttipakan kortit
- **createCards** = Pelikorttien alustaminen. Sekoittaminen, korttien data, sijoittaminen pelilaudalle.
- **flipCard** = Hallitsee korttien kääntämiseen liittyvän logiikan, kuten käyttäjän klikkaukset ja ettei kortteja voi enemmän kuin kaksi käännettynä yhtä aikaa


```

function checkForMatch() {
  const [firstCard, secondCard] = flippedCards;
  if (firstCard.dataset.emoji === secondCard.dataset.emoji) {
    matches++;
    if (matches === 10) {
      clearInterval(timer);
      alert('Voitit pelin!');
      resetGame();
    }
  } else {
    setTimeout(() => {
      firstCard.classList.remove('flipped');
      secondCard.classList.remove('flipped');
      firstCard.textContent = '';
      secondCard.textContent = '';
    }, 1000);
  }
  flippedCards = [];
}

function startTimer() {
  let seconds = 60;
  timer = setInterval(() => {
    seconds--;
    timerDisplay.textContent = seconds;
    if (seconds === 0) {
      clearInterval(timer);
      alert('Hävisit pelin!');
      resetGame();
    }
  }, 1000);
}

function resetGame() {
  clearInterval(timer);
  timer = null;
  matches = 0;
  flippedCards = [];
  gameBoard.innerHTML = '';
  createCards();
}

createCards();
});

```

Kuva 10. Scripts-tiedoston koodi osa 2 (Visual Studio Code, Kärkkäinen 2024)

Tiivistetysti ilmaistuna scripts.js-tiedostolla on viisi tehtävää pelin toiminnallisuuden kannalta. Näitä ovat korttien alustus ja sekoitus, korttien luominen, interaktiivisuus ja tapahtumankäsittely, ajastimen hallinta sekä pelin uudelleenaloitus.

4.1.4 Muistipelin pelinäkömä

Muistipeli on tallennettu tietokoneella "muistipeli" nimiseen kansioon, jossa pelin tiedostot sijaitsevat. Pelin käynnistäminen onnistuu klikkaamalla "index.html" tiedostoa kansiossa, jolloin peli avautuu selaimeen ja sitä pystyy pelaamaan.

Pelin jakaminen onnistuu myös toiselle tietokoneella jakamalla kansion kokonaisuudessaan ja tallentamalla sen sinne. Pelin käynnistäminen ei siis vaadi Visual Studio Codea toimiakseen. Peli toimii myös jokaisella selaimella.



Kuva 11. Muistipelin "pelilauta" (Kärkkäinen, 2024)



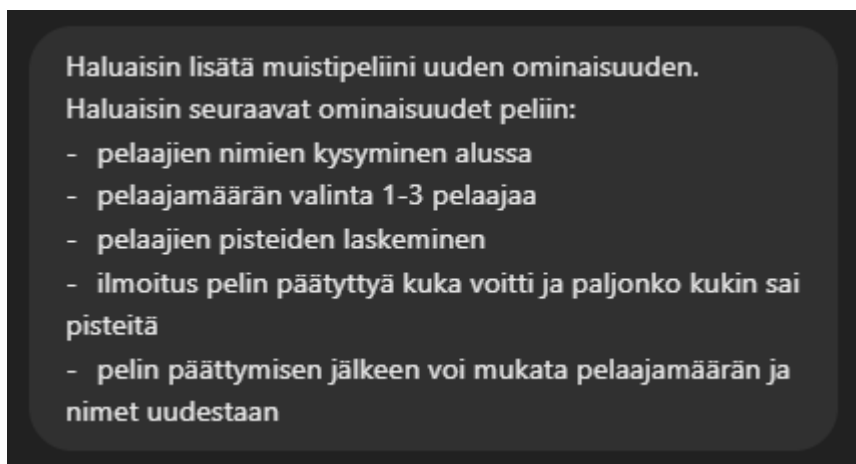
Kuva 12. Voitto ilmoitus (Kärkkäinen, 2024)



Kuva 13. Häviö ilmoitus (Kärkkäinen, 2024)

4.2 Uusien ominaisuuksien lisääminen muistipeliin

Tässä osiossa on tarkoituksena antaa ChatGPT:lle ohjeet lisäkehitykselle muistipelin suhteen ja selvittää kykeneekö tekoäly tekemään pyydetty ominaisuudet muistipeliin.



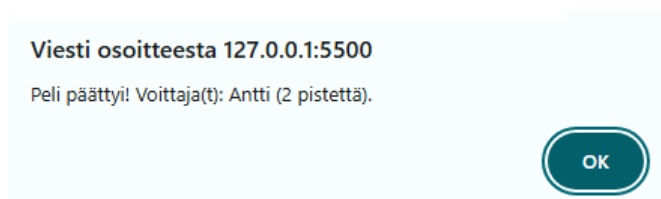
Kuva 14. Jatkokehitysidea (ChatGPT, OpenAI, Kärkkäinen 2024)

Kokonaisuudessaan tekoäly onnistui varsin mallikkaasti tekemään pyydetty ominaisuuksien lisäykset muistipeliin. Kaikki pyydetty ominaisuudet eli pelaajamäärän valinta, pelaajien nimien syöttäminen, pistelaskuri, voittaja ilmoitus sekä pelin päättymisen jälkeen muutosten tekeminen toimi moitteettomasti.

ChatGPT kertoi selkeästi mitä koodista pitää muokata, jotta halutut ominaisuudet saadaan lisättyä. ChatGPT jopa kirjoitti koodit kaikille tiedostoille alusta loppuun valmiiksi, joten tehtäväksi ei jäänyt muuta kuin kopioida koodi ja liittää se vanhan tilalle.



Kuva 15. Pelaaja määrän valinta ja nimien syöttäminen (Kärkkäinen 2024)



Kuva 16. Moninpelin voittaja ilmoitus (Kärkkäinen 2024)



Kuva 17. "Pelilauta" monipelissä (Kärkkäinen 2024)

4.2.1 Päivitetyt koodit

Päivitetty HTML-tiedosto:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Muistipeli</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="setup">
    <h2>Muistipeli</h2>
    <label for="players">Valitse pelaajien määrä:</label>
    <select id="players">
      <option value="1">1 pelaaja</option>
      <option value="2">2 pelaajaa</option>
      <option value="3">3 pelaajaa</option>
    </select>
    <div id="player-names">
      <input type="text" id="player1" placeholder="Pelaaja 1 nimi">
      <input type="text" id="player2" placeholder="Pelaaja 2 nimi" style="display: none;">
      <input type="text" id="player3" placeholder="Pelaaja 3 nimi" style="display: none;">
    </div>
    <button id="start-game">Aloita peli</button>
  </div>

  <div class="game" style="display: none;">
    <div class="timer">Aika: <span id="timer">60</span>s</div>
    <div id="scores"></div>
    <div id="current-player"></div>
    <section class="game-board"></section>
    <button id="restart-game" style="display: none;">Palaa valikkoon</button>
  </div>

  <script src="scripts.js"></script>
</body>
</html>

```

Kuva 18. Päivitetty HTML-koodi (Visual Studio Code, Kärkkäinen 2024)


```

        secondCard.classList.remove('flipped');
        firstCard.textContent = '';
        secondCard.textContent = '';
    }, 1000);
}
flippedCards = [];
currentPlayerIndex = (currentPlayerIndex + 1) % players.length;
currentPlayerDisplay.textContent = `Vuorossa: ${players[currentPlayerIndex]}`;
}

// Ajastimen aloitus
function startTimer() {
    let seconds = 60;
    timer = setInterval(() => {
        seconds--;
        timerDisplay.textContent = seconds;
        if (seconds === 0) {
            clearInterval(timer);
            announceWinner();
            resetGame();
        }
    }, 1000);
}

// Ilmoita voittaja
function announceWinner() {
    const maxScore = Math.max(...scores);
    const winners = players.filter((_, index) => scores[index] === maxScore);
    alert("Peli päättyi! Voittaja(t): ${winners.join(', ')} (${maxScore} pistettä).");
}

// Pelin uudelleenkäynnistys
function resetGame() {
    clearInterval(timer);

```

Kuva 23. Päivitetty Scripts-tiedosto osa 4 (Visual Studio Code, Kärkkäinen 2024)

```

    timer = null;
    matches = 0;
    flippedCards = [];
    scores = scores.map(() => 0);
    currentPlayerIndex = 0;

    setupDiv.style.display = "block";
    gameDiv.style.display = "none";
}

restartButton.addEventListener('click', resetGame);
});

```

Kuva 24. Päivitetty Scripts-tiedosto osa 5 (Visual Studio Code, Kärkkäinen 2024)

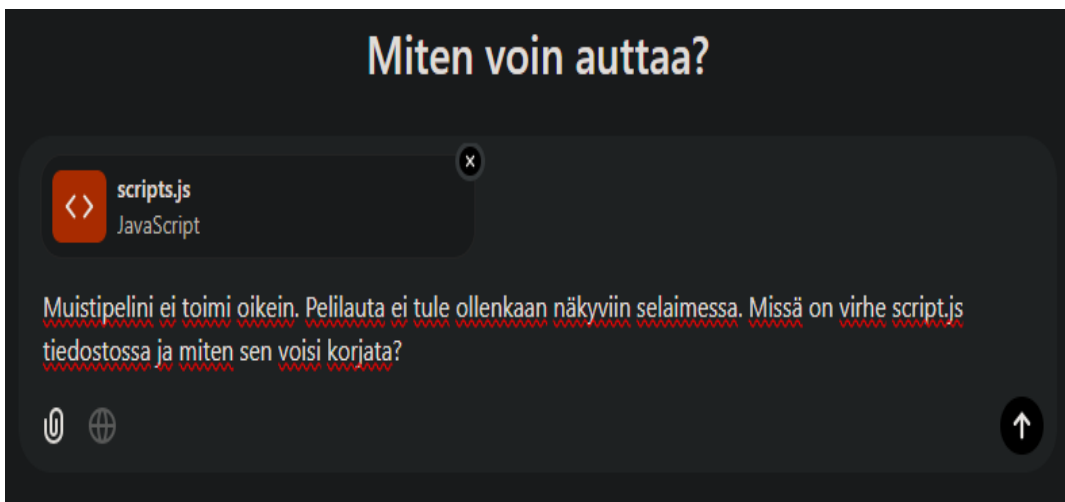
4.3 Virheen löytäminen ja korjaaminen koodista

Toinen testi oli etsiä virhe koodista. Koodiin on tarkoituksen mukaisesti upotettu virhe ja ChatGPT:tä pyydetään etsimään se. Koodiksi valittiin muistipelin scripts.js tiedostosta pieni pätkä ja koodi oli seuraavanlainen. Koodiin on merkitty punaisella kohdat, joihin on tehty virheitä. Virheet on tehty niin ettei Visual Studio Code anna suoraan virheilmoitusta. Vaikka virheet on tehty vain yhteen valikoi- tuun funktioon niin ChatGPT:lle annetaan tarkasteltavaksi koko script.js tiedoston koodi, sillä oletus tilanne on, ettei tiedetä missä funktiossa virhe sijaitsee.

```
function createCards() {
  shuffleArray(card);
  cards.forEach(card => {
    const cardElement = document.createElement('div');
    cardElement.classList.add('cards');
    cardElement.dataset.emoji = card;
    cardElement.addEventListener('click', flipCard);
    gameBoard.appendChild(cardElement);
  });
}
```

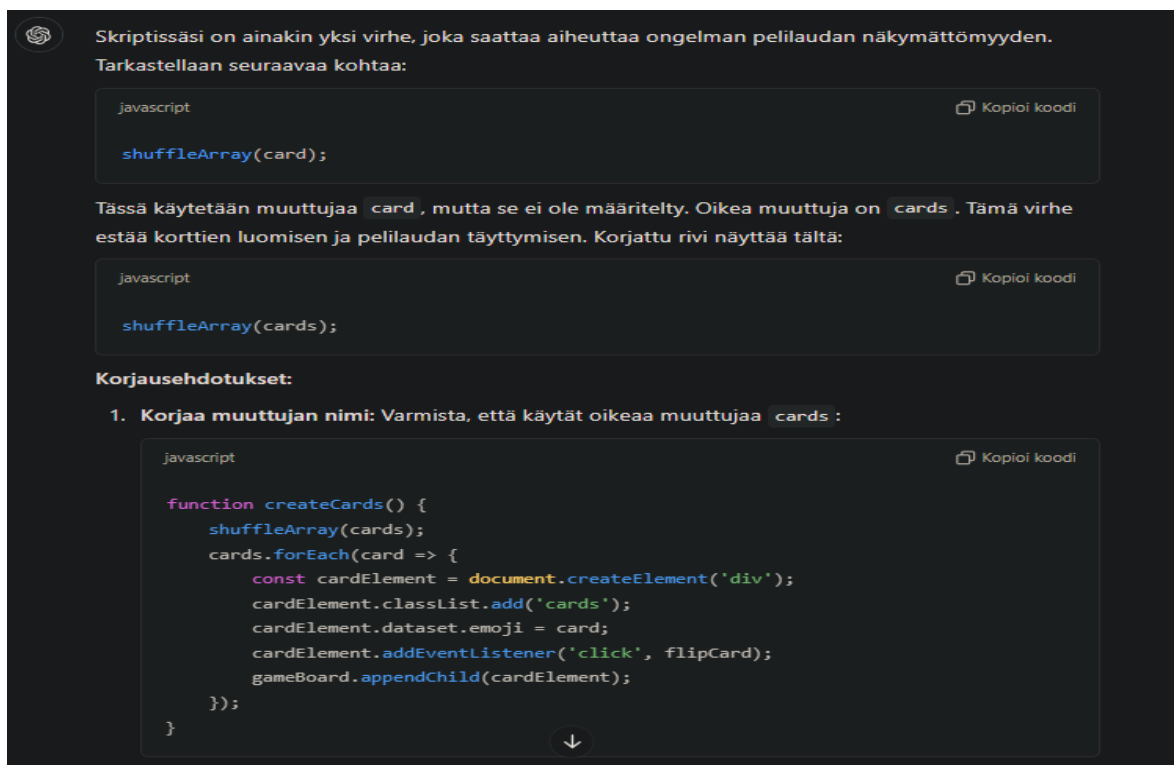
Kuva 25. Virheet koodissa (Visual Studio Code, Kärkkäinen 2024)

ChatGPT:lle syötettiin kysymys seuraavanlaisessa muodossa sekä laitettiin liitteeksi scripts.js tiedosto, jossa virheet sijaitsevat.



Kuva 26. Virheen etsiminen (ChatGPT, OpenAI, Kärkkäinen 2024)

Tässä tapauksessa ChatGPT tarjoaa suoraan ehdotuksen missä virhe voisi mahdollisesti olla sekä antaa myös korjatun koodin kokonaisuudessaan liitettäväksi vanhan tilalle. Tämän jälkeen käyttäjällä on helppo työ liittää korjattu koodi vanhan koodin tilalle.



Kuva 27. Korjaus ehdotukset (ChatGPT, OpenAI, Kärkkäinen 2024)

4.4 Tulosten analysoiminen

Kokonaisuudessaan ChatGPT onnistui tehtävänannossa ehkä jopa paremmin mitä osasin odottaa. Käytössä oli kuitenkin ilmaisversio. Koodin generoimisessa ei ollut mitään ongelmia eikä tehtävänannon noudattamisessakaan. Virheiden etsiminen ja korjaaminen onnistui myös todella vaivattomasti eikä se vienyt kauaa aikaa. ChatGPT:n tarjoama ehdotus korjaamisesta oli täysin oikeanlainen ja täten koodi saatiin korjattua.

Tuloksista ilmenee, että ChatGPT pystyy tuottamaan koodia ja korjaamaan virheitä hyvällä tasolla, jolloin koodaamisesta vähemmänkin perillä oleva voi tehdä yksinkertaisia kokonaisuuksia. Kuitenkin, kun mennään vaikeampiin asioihin niin ChatGPT kykenee tarjoamaan enää vain ohjeita ja ehdotuksia, miten asian voisi saada tehtyä. Tässä korostuu, että tekoäly vaatii tuekseen käyttäjältä vahvaa ymmärrystä koodaamisesta.

4.4.1 Muistipelin analyysi

Ensimmäinen testi oli muistipeli. ChatGPT:llä ei ollut ongelmia tehtävänannon analysoimisessa eikä niiden seuraamisessa, sillä se toteutti pyydytetyt toiminnallisuudet lähes tulkoon täydellisesti. Koodi ei vaatinut muutoksia vaan se oli valmista sellaisenaan, ei tarvinnut tehdä muuta kuin liittää ChatGPT:n tuottamat koodirivit Visual Studioon. Koodi oli myös yleisilmeeltään selkeää ja sitä oli kommentoitu myös.

Peli avautuu aivan oikein selaimen ilman, että Visual Studiota tarvitsee olla koneella asennettuna. Peli näyttää myös päällisin puolin täysin oikeanlaiselta, mutta tarkemmin peliä testailtaessa pelistä löytyy yksi pieni bugi. Pelissä on mahdollista saada peräti neljä eri symbolia edustavaa korttia käännettyä yhtäaikaan, jos käyttäjä klikkailee kortteja nopealla tahdilla. Tämä on kuitenkin pientä siihen nähden, että ChatGPT loi pelin alle minuutissa, eikä tällaisen bugin korjaaminen vaadi montaa minuuttia. Muita bugeja ei pelistä löytynyt, kaikki muut toiminnallisuudet toimivat aivan kuten tehtävänannossa pyydettiin ja peli on pelaamis- kelpoinen.

Muistipelin luomisen jälkeen testasimme kuinka pelin jatkokehittäminen onnistuisi. Jatkokehittämisen jälkeen ei pelissä ilmennyt mitään uusia bugeja. Moninpelin valinta, pelaajien nimien syöttäminen sekä pistelaskuri toimivat kaikki kuten pitikin. Toisaalta tekoäly ei huomannut aikaisemmin ilmenyttä bugia pelikorteissa tehdessään muutoksia sekä lisäyksiä pelin toiminnallisuuteen, joten bugi jäi peliin pysyvästi.

Ajatuksena olisi ollut myös liittää muistipeliin MySQL-tietokanta, mutta sitä tekoäly ei pystynyt tekemään. Tietokannan liittäminen vaatii ihmiseltä muokkaustyötä koodin ja tietokannan välillä, jotta toimiva kokonaisuus voitaisiin saavuttaa. ChatGPT kykeni antamaan ns. yleispätevät ohjeet, miten tietokannan luominen ja liittäminen tapahtuisi, mutta suoraan toimivaa kokonaisuutta se ei pystynyt luomaan.

Teoriassa ChatGPT:n ohjeet tietokannan liittämiseen olivat riittävät, mutta niiden tulkitseminen vaatii käyttäjältä osaamista tietokantojen ja koodaamisen sarakkeella. Siksi tämä testaus ei tuottanut haluttua tulosta, koska tarkoitus oli selvittää, pystyykö tekoäly suorittamaan pyydetyn tehtävänannon ilman ihmisen avustusta.



Kuva 28. Muistipelin bugi (Kärkkäinen 2024)

5 POHDINTA

Aikaisempaa kokemusta vastaavanlaisesta tutkimuksesta minulla ei ollut, toki ChatGPT:tä olen käyttänyt apurina koodaamisessa aikaisemminkin, mutta en ollut aikaisemmin kokeillut tehdä tekoälyllä toimivaa kokonaisuutta alusta alkaen, joten siihenkin peilaten tämä oli varsin opettavainen sukellus tekoälyn maailmaan.

Havaintojen perusteella en lähtisi rakentamaan suurta kokonaisuutta luottamalla pelkästään tekoälyyn. Tekoälyn tuottama koodi on jokseenkin virheeltistä ja suurempiin kokonaisuuksiin se ei vielä kykene. Vaikka tekoäly pystyy generoimaan tehokkaasti koodia, vaatii sen tuotokset kuitenkin vielä tuekseen ihmisen suorittaman tarkastamisen ja mahdollisten virheiden korjaamisen. Tekoälyn avulla voi kuitenkin helpottaa ja nopeuttaa omaa tekemistä ja antaa tekoälyn hoitaa esimerkiksi itselle valmiiksi rutiinomaisia kokonaisuuksia ja täten voi itse panostaa vaikeampiin asioihin enemmän. Lisäksi tekoäly voi toimia oivana apurina ja opettavana kanavana vähemmän koodanneille ja koodausta vasta opetteleville.

Tiedon hakeminen tällaisesta aiheesta onnistui todella vaivattomasti aiheen ollessa erittäin ajankohtainen, eläessämme tekoälyn sarakkeella murrosvaihetta. Aiheesta oli paljon tarjolla materiaalia niin suomen kuin englanninkin kielellä. Tämä kertoo siitä, että aiheesta puhutaan paljon ja se herättää kiinnostusta ihmisten keskuudessa.

Tulevaisuudessa näen tekoälyllä väistämättä suuren roolin ihmisten elämässä niin työpaikalla kuin arjessakin. Tekoäly on tullut jäädäkseen ihmisten keskuuteen, mutta se millaisia hyötyjä tai ongelmia se tuo tullessaan voidaan jättää vielä silkkään arvailuiden varaan. Yhteenvetona suosittelen kokeilemaan tekoälyn tarjoamia mahdollisuuksia. On tärkeää muistaa kuitenkin, että tekoäly on vain oiva apuri ja vaatii tuekseen ihmiseltä vahvaa osaamista ja ymmärrystä koodaamisen sarakkeelta, jotta siitä voidaan saada paras mahdollinen hyöty irti.

LÄHTEET

- Coursera. (2024). *Bert vs. Gpt: comparing language models*. Haettu 11.11.2024 osoitteesta <https://www.coursera.org/articles/bert-vs-gpt>
- Coursera. (2024). *The history of ai*. Haettu 11.11.2024 osoitteesta <https://www.coursera.org/articles/history-of-ai>
- Exploding topics. (2024). *Chatgpt users*. Haettu 18.11.2024 osoitteesta <https://exploding-topics.com/blog/chatgpt-users>
- Ibm. (2024). *Artificial intelligence (ai)*. Haettu 8.11.2024 osoitteesta <https://www.ibm.com/topics/artificial-intelligence>
- Koulutus.fi. (2023). *Mikä on chatgpt ja mitä siitä pitäisi tietää?* Haettu 13.11.2024 osoitteesta <https://www.koulutus.fi/artikkelit/mika-on-chatgpt-ja-mita-siita-pitaisi-tietaa-23286>
- Moxielearn.ai. (2024). *Chatgpt ethical issues*. Haettu 9.12.2024 osoitteesta <https://moxielearn.ai/blog/chatgpt-ethical-issues>
- Openai. (2022). *Chatgpt*. Haettu 15.11.2024 osoitteesta <https://openai.com/index/chatgpt/>
- Openai. (2023). *Gpt-3.5 turbo fine-tuning and api updates*. Haettu 15.11.2024 osoitteesta <https://openai.com/index/gpt-3-5-turbo-fine-tuning-and-api-updates/>
- Openai. (2023). *Gpt-4*. Haettu 16.11.2024 osoitteesta <https://openai.com/index/gpt-4/>
- Openai. (2023). *Gpt-4 and gpt-4 turbo*. Haettu 9.12.2024 osoitteesta <https://platform.openai.com/docs/models/gpt-4-and-gpt-4-turbo#gpt-4-turbo-and-gpt-4>
- Openai. (2023). *GPT-4O Mini: Advancing cost-efficient intelligence*. Haettu 18.11.2024 osoitteesta <https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/>
- Openai. (2023). *Openai o1 mini: advancing cost-efficient reasoning*. Haettu 27.11.2024 OSOITTEESTA <https://openai.com/index/openai-o1-mini-advancing-cost-efficient-reasoning/>
- Simplilearn. (2024). *Future of artificial intelligence*. Haettu 5.12.2024 osoitteesta <https://www.simplilearn.com/future-of-artificial-intelligence-article>
- Sulava. (2024). *Näkökulmia tekoälyn tulevaisuuteen*. Haettu 13.11.2024 osoitteesta <https://sulava.com/metaverse/nakokulmia-tekoalyn-tulevaisuuteen/>
- Tampereen yliopisto. (2022). *Eettiset kysymykset ja haasteet*. Haettu 22.11.2024 osoitteesta <https://projects.tuni.fi/kite/tekoalyn-etikka/eettiset-kysymykset-ja-haasteet/>
- Tampereen yliopisto. (2022). *Tekoälyn historia*. Haettu 11.11.2024 osoitteesta <https://projects.tuni.fi/kite/tekoalysta-yleisesti/tekoalyn-historia/>
- TechRepublic. (2023). *How chatgpt can benefit developers*. Haettu 21.11.2024 osoitteesta <https://www.techrepublic.com/article/chatgpt-benefits-developers/>

Techspot. (2014). A history of the gpu – graphics processing unit. Haettu 4.12.2024 osoitteesta <https://www.techspot.com/article/650-history-of-the-gpu/>

Techtarget. (2024). *What is artificial intelligence (ai)?* Haettu 8.11.2024 osoitteesta <https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence>

Teköly.info. (2023). Tekoälyn historia. Haettu 4.12.2024 osoitteesta https://xn--tekoly-eua.info/tekoaly_historia/

Vikash, i. (2023). *The positive and negative aspects of chatgpt for me as a software developer.* Medium. Haettu 21.11.2024 osoitteesta <https://medium.com/@i.vikash/the-positive-and-negative-aspects-of-chatgpt-for-me-as-a-software-developers-48e5c653c43a>

Yle. (2023). Haettu 20.11.2024 osoitteesta <https://yle.fi/a/74-20040724>