

Informatika Fakultatea

Informatikan Ingeniaritza Gradua

Zenbakiak batu

Unai Fraile, Amets Carrera eta Intza Larburu
2024ko maiatzaren 16a

Laburpena

Gure jokoa zenbakiz betetako taula hexagonal batean patroiak eta sekuentziak identifikatzeko duzun gaitasuna probatzeko dago diseinatua.

Taula gelaxka-matrize batez osatuta dago, eta gelaxka bakoitzak zenbaki bat dauka. Zeregina edozein norabidetan (horizontala, bertikala edo diagonal) ondoan dauden zenbaki berdinen sekuentzia luzeenak aurkitzea da eta hauek hautatzea. Jokoaren gakoak sekuentzia horiek azkar eta etenik gabe identifikatzeko gaitasunean datza, honela puntuazioa maximizatuz.

Helburu nagusia taulan zenbaki berdinen sekuentziarik luzeenak identifikatzea da. Sekuentziak horizontalki, bertikalki edo diagonaliki antolatuta egon daitezke, baina jarraituak eta etenik gabeak izan behar dira.

Aurkitzen diren eta aukeratzen diren sekuentzia bakoitzak puntuak emango ditu. Sekuentzia zenbat eta luzeagoa izan, orduan eta puntuazio handiagoa lortuko da. Puntuak lortu behar direla kontuan izanda, helburua, LLMak baina lehenago objective kasilan zehaztutako puntuak lortzea izango da. Partidan zehar, jokaldiak txandaka jokatu behar dira, hau da, lehenengo pertsona hasiko da sekuentzia bat aukeratzen eta honek jokaldia egitean LLMaren txanda iritsiko da. Honela jokoa amaierara iritsi eta bietako batek jokaldia irabazi eta besteak galtzen duen arte, edo biek jokoa galtzen duten arte.

Jolas hau ez da soilik entretenimendu-iturri bat, burua lantzeko tresna bikaina ere bada. Trebetasun kognitiboak garatuko dira, hala nola arreta, ikusi-memoria eta problemak ebazteko gaitasuna, taulako sekuentzia optimoak bilatu ahala. Partida bakoitzak hobetzeko aukera ematen du. Gehiago jolastu ahala, begi zorrotzagoa garatuko da patroiak eta sekuentziak azkar detektatzeko.

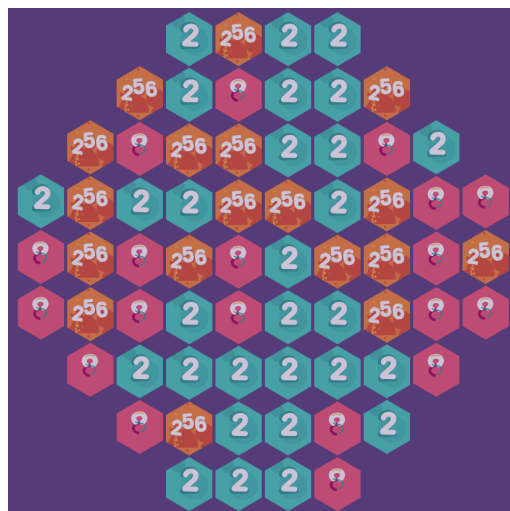
Aurkibidea

Sarrera eta Helburuak	3
Proiektuaren Kudeaketa	5
2.1: Proiektuaren irismena	5
2.2: Lanaren atazak	7
Teknologiak	9
3.1: Programazio lengoaiak	9
3.1.1: Programazio lengoaiak:	9
3.1.2: Liburutegiak:	10
3.2: Software tresnak	10
3.2.1: Garapen ingurune integratuak:	10
Soluzioaren diseinua	12
Soluzioaren inplementazioa	14
5.1: Interfazeen inplementazioa	14
5.1.1. Joko-taularen garapena:	14
5.2: Jokoaren funtzionamenduaren inplementazioa	14
5.2.2. Sekuentzien aukeraketa	14
5.2.2. Taularen eguneratzea	15
5.2.3. Puntuen eguneratzea	15
5.3: LLMaren inplementazioa	15
5.3.2. Komunikazioa	16
5.4: Traben inplementazioa.	16
Ordu-Dedikazioa	17
Ondorioak	18
7.1: Lanari buruzko hausnarketa eta ondorioak	18
7.2: Lanean ikasitakoak	19
7.3: Etorkizunerako hobekuntzak	19
Bibliografia	20

1

Sarrera eta Helburuak

Proiektu honetarako joko bat sortu behar izan dugu. Gure kasuan, jokoaren zenbakiak elkartzearena izan da. Jokoaren taula erle-panel itsurakoa da, eta hexagonoz osaturik dago, bakoitzean 2, 8 eta 256 balioen artean ausaz aukeraturiko balio bat dugularik. Zehazki esanda, bederatzi errenkadaz osatua dagoen panel simetriko bat da:



Honako hau da jokoaren interfaze totala, jokalaria eta LLMaren joko-aulak agertzen direlarik:



Helburua, ahalik eta puntu gehien egin eta objective kasilan zehaztutako puntuetara LLMa baina lehenago iristea da. Puntuak lortu ahal izateko, zenbaki bera duten eta elkarren ondoan dauden hexagonoen sekuentzia ahalik eta luzeenak egin behar dira. Honela, geroz eta sekuentzia luzeagoa aukeratu, puntuazioa ere altuagoa izango da.

Goiko irudian ikusi dezakegun bezala, interfazean bi tablero ikusi ditzakegu; ezkerrekoa jokalaria joko mahaia izango da eta eskuinekoa LLMarena. Bi zatietan, tableroaz gain, jokalaria zein LLMak lortu dituen puntuak ikusi ahal izateko gelaxka eta joko irabaztako lortu beharreko puntuen kopurua agertzen den gelaxka ikusi ditzakegu. Ezkerreko tableroan gainera, joko berriro hasteko botoia dugu, eta eskuinekoan aldiz, LLMari joko zailtzeko trabak aukeratzeko menua.

Jokoaren funtzionamenduari dagokionez, lehenengo jokalaria hasiko da jokatzeko, eta bere ustetan zenbaki bereko hexagonoz osaturiko sekuentzia luzeena aukeratzeko du botoiarekin lehen hexagonoan click eginda eta azken hexagonora arte arrastaka eramanda. Behin jokaldia amaituta, LLMaren txanda izango da, eta horrela txandaka ibilko dira joko amaitu arte.

Jokoan zehar, une batetik aurrera jokalariei ez zaizkie kasila berriak gehituko aukeratutako sekuentziaren tokian, eta beraz, kasila hauek hutsik geratzeko dira. Honela, joko amaierara iristen ari dela ikusiko dute eta geratzen diren kasilen artean egin beharko dituzte loturak. Bi jokalarietako bat sekuentziarik gabe geratzen bada eskatutako puntuetara iritsi baino lehen galdu egingo du jokaldia, baina beste jokalaria jolasten jarraitzeko aukera izango du. Aldiz, bi jokalarietako bat eskatutako puntuetara iristen bada kasilarik gabe geratu baina lehen, irabazlea izango da eta besteak zuzenean galdu egingo du.

2

Proiektuaren Kudeaketa

2.1: Proiektuaren irismena

Zenbakizko taula batean oinarritutako puzzle-jokoaren proiektu honen irismenak barne hartzen ditu joko interaktibo baten diseinua, garapena eta inplementazioa. Jarraian, proiektuaren funtsezko osagaiak eta funtzionaltasunak deskribatzen dira:

- Erabiltzaile-interfazea:

Hasteko, interfazea inplementatu behar izan dugu, hau da, joko-aula sortu, gelaxka hexagonalez osaturiko matrize bat eta gelaxka edo hexagono bakoitzari zenbaki bat esleitu diogu. Behin interfazea sortuta, joko-aula bakarrik betetzea inplementatu behar izan dugu. Hau da, jokoak funtziona zezan hainbat kontrol gehitu ditugu.

Kontrolei dagokienez, partida berri bat hasteko botoia, puntuak gordetzeko gelaxka eta helburura iristeko lortu beharreko puntuak gordetzen dituen gelaxkak ere gehitu ditugu. Behin kontrolak inplementatuta, LLMak jokatu ahal izateko pantaila bikoiztu behar izan dugu, hau da, bi joko-aula ezarri, baina kontuan izanik, bi tauletan ezarritako hexagonoen balioak berdinak izan behar zirela eta baita gelaxkak kentzen zirenean ondoren txertatzen zirenak ere.

Behin joko-aula ongi sortuta, sekuentzia bat aukeratzeko gelaxkak nabarmentzeko, aukeratutakoak desagertzeko... animazioak gehitu ditugu.

- Joko Logika:

Behin interfazea sortuta, jokoaren logika inplementatu dugu. Hasteko, joko-aula ausaz sortutako 2, 8 eta 256 balioen arteko zenbakiekin bete ahal izateko algoritmoa inplementatu dugu. Hori egin ostean, sekuentziak identifikatzea inplementatu dugu. Honela, jokalaria sekuentzia bat identifikatzean aukeratutako balioak desagerrarazi eta balio berriak txertatu ditugu. Horretarako, erabiltzaileak gelaxka

baten gainean click egin eta arrastaka eramanda identifikatu dituen zenbakiak hartu ditugu (gutxienez 2-koa baldin bada sekuentzia), click-a askatu arte.

Behin sekuentziak identifikatuta, puntuazioak eguneratu behar izan ditugu. Honetarako, aukeratutako sekuentzia zuzena zela begiratu, eta honen luzeraren arabera jokalariaen puntuak gehitu ditugu. Gainera, interfazean bertan gehitutako puntuak gordetzeko kontrola ere eguneratu dugu.

- Interakzioa Hizkuntza Ereduarekin (LLM)

Azkenik, Groq hizkuntza-ereduarekin komunikatzea ahalbidetu dugu, honek, sekuentzia optimoak itzuli ditzan inplementatuz. Horretarako, erantzunak proposatu dizkiogu eta berak horien artean optimoena aukera zezan eskatu diogu. Jasotako erantzuna kontuan edukita LLMaren interfazearen zatia eguneratu dugu, honek aukeratutako sekuentzia desagerraraziz eta gelaxka berriak txertatuz.

2.2: Lanaren atazak

Jokoa garatzerako orduan, hainbat fase ezberdin izan ditugu.

- Plangintza-fasea:

Hasteko, irakasleak zenbakiak lotu jokoa inplementatu behar genuela esan zigunean, proiektuaren betekizunak zehaztu genituen. Horretarako, hainbat erabaki hartu behar izan genituen, hala nola, interfazea nolakoa izango zen, partida irabazteko zer egin beharko zen, puntu konkretu batzuetara iritsi, denbora bat igaro ostean puntu gehien izan..., gehituko genituen LLMari jarriko genizkion trabak zein izango ziren... Erabaki hauek taldekide guztien artean hartu genituen, honela, lantalde guztiaren onespena bermatuz.

Behin egin behar genuena ongi ulertu ostean, lan banaketa egin genuen. Kideetako bat LLMaren inguruan informazioa bilatzen eta ulertzen hasi zen, eta beste bi kideak frontend atalean murgildu ziren. Honela, garapen edo inplementazio fasera igaro ginen.

- Garapen fasea:

Proiektua garatzen hasterakoan, lehenengo interfazeak garatu ditugu, hau da, joko-taula inplementatu dugu. Horretarako React tresna erabili dugu. Behin joko-taula eta bertan integratutako gelaxkak sortuta, jokorearen inplementazioa egin dugu. Hasteko, gelaxkak hautatzeko eta nabarmentzeko funtzionaltasunak gehitu ditugu. Horren ostean, zenbakien sekuentzia jarraituak identifikatzeko algoritmoak eraiki ditugu, ondoren hautatutako hauek ezabatu eta taula eguneratzeko logika inplementatu ahal izateko. Azkenik, puntuak eguneratu ditugu hautatutako sekuentzia kontuan izanda.

Behin jokorearen logika inplementatuta, hizkuntza-eredua integratu dugu, honekin komunikazioa ahalbidetuz. Horretarako Groq SDK erabili dugu eta honen erantzunak jaso ahal izateko logika inplementatu dugu. Behin bere erantzunak jasota, jokorean integratu ditugu, hau da, berak egindako mugimendua kontuan izanda taula eguneratu dugu. Behin jokorearen inplementazio orokorra izanda probak egin ditugu.

- Proben fasea:

Garatutako inplementazioa probatu ahal izateko probak egin ditugu. Honetarako, jokoa exekutatu dugu orain arte egindakoan izandako akatsak identifikatu eta zuzendu ahal izateko. Honetarako garrantzitsua izan da taldeko kide guztiok probatzea jokoa, bakoitzak gabezia desberdinak identifikatu baitugu.

Probak gehien bat amaieran egin ditugun arren, garapen fasearekin nahastuta egin dugula fase hau esan dezakegu. Hau da, kontrol edo funtzio berriren bat inplementatzen genuen bakoitzean, honen funtzionamendu egokia frogatu ahal izateko probatu egin dugu, honela, amaierara ahalik eta produktu optimoenarekin iristen saiatuz.

- “Despliege” fasea:

Fase hau egiteko docker plataforma erabili dugu. Docker teknologiaren bitartez posible da aplikazio baten dependentzia guztiak “container” batean sartzea eta horrela lortzen da modu independentean tratatzea aplikazio bakoitza, horrela proiektua isolatzea lortuz. Hori horrela, herraminta hau erabilita posible izan dugu proiektu guztia komando bakar batekin desplegatzea, front-end eta back-end “container” ezberdinetan utziz eta zati hauetako bakoitza modu independentean exekutatzeke aukera izanez. Beraz gure

proiektua exekutatzeko nahikoa da hurrengo pausuak jarraitzea:

- Lehenik eta behin, gure Githubeko errepositorioa klonatu behar da terminaletik **`git clone <repoaren_https_kodea>`** komandoa erabilia.
- Ondoren errepositorioko “root”-era joan behar da **`cd lanPraktikoa_Carrera_Fraile_Larburu`** komandoa exekutatuz.
- Hor gaudela, komando bakar batekin posible da dependentzia guztiak instalatzea eta martxan jartzea gure aplikazioa, komandoa hau litzateke: **`docker-compose up --build`**

Modu horretan nabigatzailetik <http://localhost:3000> url-a atzituz lortuko dugu gure frontend atala ikustea eta <http://localhost:4000> url-a atzituz backend atala atzitzea (kasu honetan ez dugu erabili behar izan).

3

Teknologiak

3.1: Programazio lengoaiak

3.1.1: Programazio lengoaiak:

Proiektua garatzeko hainbat programazio-lengoaia eta teknologia erabili ditugu; JavaScript eta CSS. JavaScript programazio lengoaia izan da gehien erabili duguna. JS erabili dugu jokoan interaktiboak diren ataza guztiak garatzeko eta dinamismoa emateko, %70 inguruan izan da honen erabilera. Beste aldetik CSS eta HTML erabili ditugu, lehenengoa estiloak eta jokoaren diseinua egiteko erabili dugu eta honen erabilera proiektuan %24 ingurukoa izan da. HTML proiektuaren estruktura edo egitura egiteko erabili dugu, eta hau erabilera gutxien eman diogun programazio lengoaia izan da, %6 inguruan.

Jarraian, labur azalduko ditugu erabilitako lengoaiak eta hauek erabiltzeko arrazoiak:

JavaScript: Programazio-lengoaia interpretatua da eta oso ohikoa da web garapenetan erabiltzea, orri dinamiko eta interaktiboak sortzeko. JS funtsezko elementua da DOMa manipulatzeko, eta horri esker, web orriaren egitura eta edukiak eguneratu eta aldatu ahal izan ditugu, berriro kargatu behar izan gabe.

CSS: HTML edo XMLn idatzitako dokumentuen aurkezpena deskribatzeko erabiltzen den estilo-orrien lengoaia da. CSSk webguneen diseinu bisuala kontrolatzea ahalbidetzen du; diseinua, koloreak... Gure kasuan, CSS jokoaren interfazea diseinatzeko eta itxura emateko erabili dugun lengoaia izan da.

3.1.2: Liburutegiak:

Lengoaia ezberdinez gain, gure proiektua garatu ahal izateko hainbat liburutegi erabili ditugu; React, SCSS eta Groq-SDK.

React-DOM: Liburutegi honek React osatzen du, nabigatzailearen DOMa manipulatzeko metodo espezifikoak ematen dituelako.

Groq-sdk: Groq-ek duen lengoaiaren eredua gure aplikazioan erabili ahal izateko erabili dugu liburutegi hau. Honek, Groq-eko APIari eskaerak egiteko aukera ematen digu, jokoaren inguruko testuingurua emanda berak erantzunak eman ahal izateko. Gure adibidean, partidako hainbat sekuentzia eman eta onena aukeratzeko eskatu diogu, honela, pertsonaren aurka partidak jolastuz.

SCSS: CSS lengoaiaren hedapen bat da, funtzionalitate aurreratuak eskaintzen dituen; aldagaiak... Honela, orriei estilo garbi eta erabilgarriagoa ematea ahalbidetuz. Gure proiektuan liburutegi hau estilo konplexuagoak garatzeko erabili dugu.

3.2: Software tresnak

3.2.1: Garapen ingurune integratuak:

Proiektu hau garatzeko hainbat software-tresna erabili ditugu, proiektuaren kudeaketa errazteko; Visual Studio Code eta Node.js. Jarraian labur azalduko ditugu tresna hauek:

Visual Studio Code: Microsoftek garatutako iturburu-kodeko editorea da, eta oso erabilgarri eta ezaguna da web-garatzaileren artean. Tresna honek JavaScript eta React erabilita garapena errazteko hedapen sorta zabala eskaintzen ditu. Bestalde, oso erabilgarria den Copilot tresna ere du integratua, proiektuak garatzeko erraztasun handiak ematen dituen tresna. Guk, gure proiektua garatzeko tresna hau erabili dugu, eta beraz, implementazio guztia bertan egin dugu.

Node.js: JavaScripten exekuzio-gune bat da, eta zerbitzariaren aldean, JavaScript leangoaiean idatziriko kodea exekutatzeak aukera ematen du. Gure proiektuan, Node.js mendeotasunak instalatzeko eta eraikuntza- eta hedapen-zereginak automatizatzen dituzten script-ak exekutatzeak erabili da.

Aipatutako tresna eta liburutegiez gain, beste hainbat zerbitzu ere erabili ditugu proiektuaren gaitasunak handitzeko, hala nola, gitHub.

GitHub: Iturburu-kodearen kudeaketarako erabili dugu gure proiektuan. Honela, kodearen segurtasuna babesteaz gain, taldeko kideen arteko lankidetzan ere lagundu digun tresna erabilgarria izan da.

Laburbilduz, esan dezakegu teknologia hauek erabiltzeak web aplikazio sendo eta eraginkor bat sortzea ahalbidetu digula. JavaScript eta React konbinazioak oinarri sendoa eman digu interfazea garatzeko unean. Liburutegi eta tresna gehigarriak erabili izanak, proiektuaren kudeaketan eta funtzionalitate aurreratuen integrazioan lagundu digu. Beraz, erabilitako tresnek garapena azkartzeaz gain, kalitatezko kode bat egitea ahalbidetu digute.

4

Soluzioaren diseinua

Kasila hexagonalak batzeko jolasa garatzeko, hainbat teknika eta teknologia moderno erabili ditugu, proiektuaren eraginkortasuna, modularitatea eta eskalagarritasuna ziurtatzeko. Hauek dira erabilitako teknika nagusiak:

Interfazeak garatu ahal izateko React erabili dugu. React JavaScripten liburutegi bat da, erabiltzaile-interfazeak eraikitze asko erabiltzen dena. Osagai berrerabilgarriak sortzeko aukera ematen du, eta aplikazioaren egoera eraginkortasunez maneiatzen lagutzen du. Proiektuan, React jokoaren interfazea osagai berrerabilgarrietan zatitzeko erabili dugu, hau da, joko-aula (LLMBoard, Board), banakako gelaxkak (Tile), eta beste elementu batzuk sortzeko.

Jokoaren diseinua eta estilorako CSS erabili dugu. CSS aurreprozesadore bat da, estiloen kodearen mantentzea eta eskalagarritasuna errazten duena, eta honegatik erabili dugu guk gure kodeko estiloa garatu ahal izateko.

Jokoaren egoera eta erabiltzailearen interakzioak kudeatzeko, useState eta useEffect erabili ditugu. Hauei esker, erabiltzailearen ekintzen arabera lan egiteko egoera interaktibo bat mantendu dugu.

Azkenik, LLMaren integrazioa egin ahal izateko Groq-sdk liburutegia erabili dugu, jokoaren erabiltzailearen aurka jolasean adimen artifizial bat ibili dadin inplementatu ahal izateko. Ereduak, taularen egoera kontextu gisa jaso eta mugimendu estrategikoak itzultzen ditu.

Proiektua hainbat osagaiz osatua dago, hauetako bakoitzak jokoan erantzukizun bat duelarik. Jarraian, osagai nagusiak eta bakoitzaren funtzionalitateak azaltzen dira:

BoardView osagaia joko taula kudeatzeaz arduratzen da. Taularen eta erabiltzailearen arteko interakzioaz arduratzen da, hautatutako tilak ezabatuz, eta jokoa eguneratuz. Tile-a joko-aulan txertatutako hexagono bakoitza irudikatzen duen osagaia da. Honek ere, erabiltzailearen interakzioak kontuan izanda, bere burua eguneratzen du aukeratua izan den ala ez kontuan izanik.

GameOverlay osagaia jokoaren amaierako mezuak erakusteko erabili dugu, hala nola, Winner edo Game Over, Try Again mezuak erakutsiz, jokoaren egungo egoeraren arabera. Kasu honetan jokalaria batek edo LLM-ak ez badu sekuentzia gehiago aukeratzeko modurik edo ez bada helburu puntuaziora iritsi, orduan Game Over pantaila agertuko da. Ordez, helburu puntuaziora iritsi edo gainditzen bada orduan Winner pantaila erakutsiko da eta aurkariari GameOver, pertsonak hauetako batean sakatzean jokua berrabiaraziko da.

LLMBoardView Board-en antzeko osagaia da, baina hizkuntza-ereduaren edo LLMaren ereduak kontrolatzen ditu. Zehazki esanda, osagai hau AAaren ereduarekin komunikatzen da mugimenduak lortu ahal izateko eta jokoaren egoera eguneratzen du.

Board osagaia, joko-taularen logika kudeatzeaz arduratzen da, tilak sortuz, taula betez, markatutako tilak kenduz... Jokalariak irabazi edo galdu duen zehazteko logika ere barne hartzen du.

5

Soluzioaren implementazioa

5.1: Interfazeen implementazioa

5.1.1. Joko-taularen garapena:

Hasteko, proiektuaren interfazea garatu dugu, jokoaren implementazioa egin baina lehen, oinarri bat edukitzea beharrezkoa delako. Gure jokoaren interfazea erle panel baten itsura duen tablero bat da, hexagono forma duten laukiekin osatua. Hori da gure joko-taularen oinarria eta horren gainean txertatzen dugu kasilla edo hexagono zenbakidun bakoitza. Horretarako, lehenengo Board klase bat implementatu dugu eta gainean Tile klaseko elementuak txertatu dizkiogu, hausaz aukeratutako 2,8,256 balioekin.

Behin erle panela ongi inplementatuta, beste hainbat funtzio gehitu dizkiogu gure interfazeari; newGame botoia, lortutako puntuak gordetzeko kutxatxo eta jokia irabazteko lortu beharreko puntuak gordetzen dituen kutxatxo.

5.2: Jokoaren funtzionamenduaren implementazioa

5.2.2. Sekuentzien aukeraketa

Aurreko atalean azaldutako joko-taula izanda, jokoak funtziona dezan funtzionamendu ezberdinak gehitu dizkiogu. Hasteko, taulan bertan sekuentzia bat aukeratzeko aukera gehitu dugu. Horretarako, kasilla batean click egindako unetik, click-a askatzen den arte gainetik igarotako kasilla guztiak gorde eta markatu ditugu. Behin hori eginda, eta zenbaki berekoak direla aztertuta, hau da, sekuentzia bat osatzen dutela aztertuta, hau markatu dugu, eta taulatik borratu dugu.

5.2.2. Taularen eguneratzea

Aukeratutako sekuentziaren kasilak une horretan hutsik geratzen dira, beraz, hauen gainean elementu berriak txertatu ditugu, erabiltzaileak jolasten jarraitu ahal izateko. Jokoari amaiera bat eman behar diogunez, plangintzan aukeratu genuen 150 Tile berri txertatu ostean, sekuentzia bat aukeratu eta gero ez genituela kasila gehiago txertatuko. Horretarako, 65 elementuz osaturiko hausazko array bat sortu dugu, eta elementuak amaitu arte gehitu ditugu jokoan. Behin horiek amaituta, erabiltzaileak taulan geratzen diren hexagonoen artean egin beharko ditu sekuentziak, agortu arte.

5.2.3. Puntuaren eguneratzea

Aukeratutako sekuentziak kontua izanda, erabiltzaileak lortutako puntuak eguneratu behar dira. Horretarako, erabiltzaileak egindako mugimendu bakoitzaren ostean puntuak eguneratu ditugu, aukeratutako sekuentziaren luzeraren arabera. Orduan eta sekuentzia luzeagoa aukeratu lortutako puntuak ere gehiago izango dira.

5.2.4. NewGame botoiaren inplementazioa

Aurrez azaldu bezala, interfazean botoi bat gehitu dugu joko berri bat hasi nahi izanez gero. Botoi hau sakatzean, joko-taula eguneratu dugu berriro hausaz sorturiko hexagonoz beteaz. Gainera, puntuaren laukia ere hasieratu dugu, eta 0an ezarri dugu berriro.

5.3: LLMaren inplementazioa

5.3.1. Interfazearen bikoiztea

Behin jokoaren funtzionamendu totala inplementatuta, daukagun interfaze osoa bikoiztu behar izan dugu, LLMarekin lana egin ahal izateko. Horretarako, kontuan eduki behar izan dugu bi aldeetan hausaz sorturiko balioak berdinak izan behar direla, eta sekuentzia bat aukeratu ostean agertuko diren kasila berrien balioak ere berdinak izango direla. Azkenean, orriaren ezkerreko aldean jokalaria joko-taula ezarri dugu eta eskuinekoan LLMarena.

5.3.2. Komunikazioa

LLMarekin jolastu ahal izateko interfazea sortuta, komunikazioa hasieratu dugu. LLMari bere joko-taulan aukera gisa dituen hiru sekuentzia bidali dizkiogu kontextu gisa, eta berak, erantzun bezala hiru horien artean egokiena aukeratu eta itzultzen du. Jasotako sekuentzia hori erabilia LLMaren joko-taula eta puntuak eguneratu ditugu.

LLM-arekiko konexioa eta galderak backend ataletik egitea izango litzateke optimoena, bai logika aldetik eta baita kodearen berrerabulpen eta mantenuaren ikuspuntutik. Hala ere, irakaslearekin kontsultatu ondoren eta kontuan hartuz gure kasuan ez zela beharrezkoa backend zatia garatzea (ez baitugu datu-ereduaren beharrik), adostu zen ez zela implementatuko backend-ean eta horren ordez frontend zatian egingo zela komunikazioa LLM_arekin.

5.4: Traben implementazioa.

Jokoaren funtzionamenduari lotuta, jokalariai lagundu ahal izateko hiru funtzionamendu berri gehitu dizkiogu proiektuari, LLMari joko zailtzeko asmoz. Funtzionalitate hauek "Model degradation", "Max path degradation" eta "Information penalty" izan dira; lehenengoa, Groq LLMaren modeloa aldatzeko da, hau da, hasieran zehaztuta dagoen "Llama3-70b-8192" modelotik "Gemma-7b-It" modelora aldatzen du, bigarrenak zehaztasun txikiagoa duela kontuan izanik, eta ondorioz, erabakiak hartzerako unean ez ditu aukera optimoenak aukeratuko. Bigarren trabak, LLMari aukeratzeko pasatako sekuentzia aukerak okertzen ditu, honela, aukerek gehienez 6 elementuko lotura izan dezakete. Azkenik, hirugarren trabak lehenengoaren logika antzekoa erabiltzen du, baina kasu honetan modeloa aldatu beharrean, LLMari idazten zaion eskaera aldatzen da, eta erabaki egokiena eskatu ordez, erabaki okerreana (beti ere hiru sekuentzia luzeen artean) hartzeko eskatzen zaio.

6

Ordu-Dedikazioa

Honako hauek izan dira taldeko kide bakoitzak proiektua garatzen igarotako orduak:

	Unai	Ametz	Intza
Orduak gelan	9	9	9
Orduak gelatik kanpo	16	20	18

7

Ondorioak

7.1: Lanari buruzko hausnarketa eta ondorioak

Proiektu honen garapena orokorrean aberasgarria eta atsegina izan da guretzat, aukerak eman baitizkigu softwarearen garapenaren eta adimen artifizialaren hainbat arlotan gehiago sakondu eta trebetasunak handitzeko. React erabilia taula hexagonal bat sortu eta aurkari gisa AA bat izateak, joko erakargarri bat garatzeaz gain, proiektu konplexuak egiterako unean dugun gaitasuna indartzen lagundu digu.

Esan dezakegu proiektua garatzeko unean asakatasun handiagoa izan dugula gainerako irakasgaietako poriketuekin alderatuta, hau da, jokoaren funtzionamendua nolakoa izan behar zen zehaztua izan dugun arren, malgutasuna eduki dugulako proiektua garatzerako orduan. Adibidez, interfazea guk nahi dugun bezalakoa izan da, LLMari trabak jartzerakoan ere guk aukeratu ditugu... irakasleak dena zehaztuta eman beharren. Honek ere asko lagundu digu gure gaitasunak garatzerako orduan.

Aldi berean, onartu beharrekoa da, honek gaitasunak garatzerako unean onurak ekarri dizkigun arren, zailtasunak ere eragin dizkigula. Hasieran, interneten informazioa bilatzerako orduan ez genuen gure jokoaren antzekoa zen inolako adibiderik topatu, eta beraz, zailtasunak izan genituen hasieran proiektuaren nondik norakoak zehaztu eta hau hasterakoan. Honen ondorioz, interfazearen eta diseinuaren ataleran denbora asko igaro behar izan genuen, ez baigenuen esan bezala interfazea inplementatua zuen inolako adibiderik topatu.

Bestalde, LLMaren zatiari dagokionez, ez genuen lehenago inoiz adimen artifizial batekin lanik egin, eta hasieran nahiko galduta ibili ginen eta asko kosta zitzaigun honekin lan egiten hastea. Zehatzago esanda, LLMarekin komunikazioa ongi hasi bagenuen ere, gure jokoaren kontextua pasatzerako unean ez zigun behar genuen emaitza itzultzen, eta hori lortzera bidean zailtasun handiak izan genituen.

Laburbilduz, proiektua interesgarria eta baliagarria izan den arren, zailtasun ugari ere izan ditugu hau garatzerako bidean, baina azkenean, entretenigarria, dinamikoa eta bisuala den joko egoki bat garatu ahal izan dugu.

7.2: Lanean ikasitakoak

Orain arte aipatutakoaren haritik, hainbat baliabide eta tresna berri erabiltzen ikasi dugu, baina alor teknikoa alde batera utzita, proiektu honek beste hainbat irakaspen ere eman dizkigu.

Alde batetik, lan banaketa egoki bat egiteak garrantzi handia duela ikasi dugu. Hasieran hirurok frontend-aren zatiarekin edo interfazearen zatiarekin hasi ginen eta LLMaren zatia alde batera utzi genuen. Proiektuan aurrera joan ahala konturatu ginen denbora galtzen ari ginela, eta irakaslearen gomendioei kaso egin eta kide bat LLMan lanean jartzea beharrezkoa zela ikusi genuen denbora optimizatu ahal izateko.

Bestalde, atazak egiten hasi aurretik informazioa bilatzea eta egiten ari garena ongi ulertuz egitea beharrezkoa dela ikusi dugu, bestela, denbora asko galtzen baita akatsak zuzentzen.

7.3: Etorkizunerako hobekuntzak

Esan bezala, proiektuan askatasun handiagoa izateak ikasteko balio izan digun arren, agian LLMaren zatiaren inguruan zerbait gehiago sakontzea proiektua azaltzerako orduan baliagarria izan daiteke. Guk gure kontura ikasi behar izan dugu LLM baten funtzionamendu guztia, eta honek, denbora asko eskaini behar izateaz gain, zailtasun handiak ekarri dizkigu proiektua garatzerako unean.

Arazo hau txikitzeko, agian proiektuarekin batera LLMaren inguruko azalpen hedatuago bat ematea lagungarria izan daiteke nondik hasi behar den jakiteko.

Bibliografia

- [1] <https://es.react.dev/learn/tutorial-tic-tac-toe>
- [2] <https://chatgpt.com/?oai-dm=1>
- [3] https://js.langchain.com/v0.1/docs/get_started/introduction/
- [4] <https://console.groq.com/playground>
- [5] https://www.youtube.com/watch?v=qx3adFfbJR&t=188s&ab_channel=AMPTech
- [6] <https://github.com/puigalex/LangChain>