

BILBOKO INGENIARITZA ESKOLA

KUDEAKETAREN ETA INFORMAZIO SISTEMEN
INFORMATIKAREN INGENIARITZAKO GRADUA

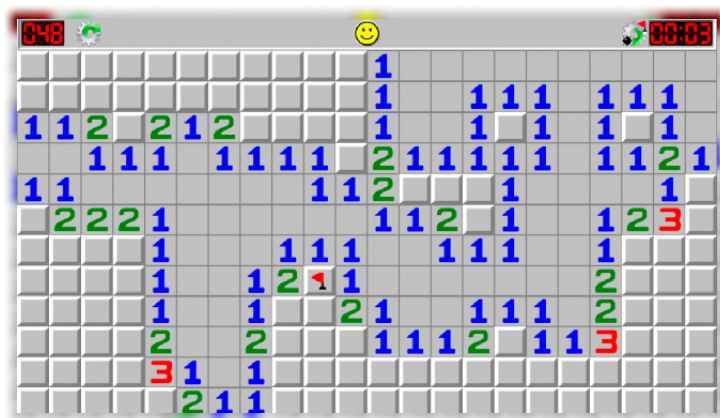
DRAGAMINA

Egileak:

Adei Arias
Jon Barbero
Álvaro Hernández
Ander Prieto
Ena Verhorst

Arloa:

Software Ingeniaritza
2. maila
46. taldea
2. lauhilabetea



Aurkibidea

1. Sarrera	1
2. Antolaketa	2
3. Diseinua	3
3.1. Patroiak	3
3.1.1. Strategy	3
3.1.2. Factory	3
3.1.3. MVC	3
3.1.4. Observer	4
3.2. Klase diagrama	4
3.3. Sekuentzia diagrama	4
3.4. 1. sprint	9
3.5. 2. sprint	9
3.6. 3. sprint	10
4. Garapena	11
4.1. 1. sprint	11
4.1.1. Enuntziatuan agertzen ez diren funtzionalitate edota lan extrak	11
4.1.2. Aurkitutako arazoak + konponbideak	11
4.1.3. Atazen dokumentazioa	11
4.1.4. Proben dokumentazioa	13
4.1.4.1. ID Proba: 2.1	13
4.1.4.2. ID Proba: 3.1	14
4.1.4.3. ID Proba: 3.2	14

4.2. 2. sprint	15
4.2.1. Enuntziatuan agertzen ez diren funtzionalitate edota lan extrak	15
4.2.2. Aurkitutako arazoak + konponbideak	15
4.2.3. Atazen dokumentazioa	15
4.2.4. Proben dokumentazioa	16
4.2.4.1. ID Proba: 1.1	16
4.2.4.2. ID Proba: 4.1	17
4.2.4.3. ID Proba: 4.2	17
4.2.4.4. ID Proba: 4.3	17
4.3. 3. sprint	18
4.3.1. Enuntziatuan agertzen ez diren funtzionalitate edota lan extrak	18
4.3.2. Aurkitutako arazoak + konponbideak	18
4.3.3. Atazen dokumentazioa	18
4.3.4. Proben dokumentazioa	20
4.3.4.1. ID Proba: 5.1	20
4.3.4.2. ID Proba: 5.2	20
4.3.4.3. ID Proba: 5.3	21
4.3.4.4. ID Proba: 6.1	21
4.3.4.5. ID Proba: 6.2	22
4.3.4.6. ID Proba: 7.1	22
4.3.4.7. ID Proba: 7.2	22
5. Ondorioak	24
Erreferentziak	25

1. Sarrera

Software Ingeniaritza ikasgaieko proiektu finala dragamina bat sortzean datza. Dragamina bideojoko bat da, Robert Donnerrek asmatuta 1989. urtean^[1]. Jokoaren helburua minaz betetako esparru bat guztiz garbitzea da, minarik ikutu gabe.

Jokoaren arauak hurrengoak dira:

- Esparrua gelaxkaz osatuta egongo da. Gelaxka batzuek zenbaki bat dute; zenbaki honek bere inguruko gelaxketatik zenbatek duten mina adierazten du. Hau da, gelaxka batek 3 zenbakia badu, bere inguruko 8 gelaxketatik (gutxiago izango dira tableroaren mugak badira) hiruk mina daukate, eta beste bostek ez.
- Zenbaki gabeko gelaxka bat altxatzeak inguruko gelaxketan minarik ez dagoela esan nahi du; beraz, zortziak altxatuko dira ere.
- **Minadun gelaxka bat altxatzen bada, partida bukatuko da, porrota lortuz.**
- Laguntza gisa, bandera bat jarri daiteke gelaxka baten gainean, mina gelaxka horretan egon daitekeela adierazteko.
- **Mina gabeko gelaxka guztiak altxatzean, partida bukatuko da, garaipena lortuz.**

Jokoak sarrerako leiho bat izango du, non jokalaria izena eta jokoaren zailtasuna (1-3 artean) eskatuko zaion. Partidan sartzean, dinamika erraza da. Zailtasunaren arabera mina kopuru bat egongo da. Gelaxkak altxatzeko, ezkerreko klika eman behar da. Mina duen gelaxka aurkitzen (uste) denean, eskumako klikarekin bandera batekin markatuko da. Banderadun gelaxka bat ezingo da altxatu, baina marka bai kendu ahalko zaio. Gainera, posizio batetan dudarik balego, berriz ere eskumako klika ematean galdera ikur batekin markatuko da. Banderaduna ez bezala, gelaxka hau bai altxa daiteke.

Jokoa irabaztean, leihoaren behealdean dagoen betaurrekodun aurpegiari klik egitean puntuazio historiko altuenak erakusten dituen beste leiho bat irekiko da.

Honetaz gain, partida berri bat hastea posible izango da une oro. Aurpegitxoa sakatzean, zailtasun bereko beste panel bat sortuko da. Partida galdu berria bada, leiho bat agertuko da galdetuz partida berria hasi nahi bada. Beste zailtasuneko partida bat hasi nahi bada, goiko menuaren bitartez egin daiteke. Menu horretan, honetaz aparte, laguntza ere egongo da. Arestian aipatutako arauak agertuko dira orduan, behar izanez gero.

Java erabiliz, Windowsen joko historiko hau birsortzea lortuko dugu. Dokumentu honetan prozesuaren zehaztasun teknikoak azalduta egongo dira, diseinutik garapenera.

2. Antolaketa

Lana hiru atal nagusitan banatu da:

- **Back-end**¹: Kode nagusia, jolasa kudeatuko duena eta erabakiak hartuko dituen.
- **Front-end**¹: Interfaze grafikoa da, erabakien arabera funtzionatzeko duena eta pantailan erabakien emaitzak bistaratuko dituen.
- **Diagramak**:
 - Alde batetik, **klase diagraman** jokuak era egoki batean funtzionatzeko sortu behar izan diren klaseak agertzen dira, haien atributu eta metodo propioekin, eta baita klaseen arteko erlazioak.
 - Bestetik, **sekuentzia diagraman** programa abiarazten denetik hartutako erabakien arabeko prozesu sekuentzia erakusten da.

Eclipse erabili dugu kodea idazteko. IDE oso potentea da, eta **WindowBuilder** pluginarekin front-enda edo bista sortzea lortu dugu. Plugin honek **Swing** liburutegi grafikoarekin lan egiten du, basikoa izan arren gure lana asko errazten digulako.

Atal nagusi hauek taldekideen artean banatu dira eta emandako aurrerapausuak lankide guztiarekin partekatzeko aplikazio desberdinak erabili dira. Batetik, eguneroko aldaketak edo sortu diren zalantzak **WhatsApp** talde batetik komentatu dira. Gainera, astean birritan (gutxienez) bilerak egin dira **Discord** aplikazioaren bitartez. Aplikazio honek audioa eta ordenagailuko pantaila partekatzeko aukera ematen du, horrelako lanak egiteko oso aproposa.

Gainera, kodea guztiok sinkronizatzeko **GitHub** plataforma erabili dugu. “DRAGAMINA” izeneko repositorioa^[2] sortu genuen, eta bakoitzak aldaketa bat egitean “commit” eginez bertara igoko da, guztiok “pull” egitean eskuragarri izateko.

Proiektua hiru sprintetan banatuta egon da; sprint bakoitzeko, eginkizun batzuk bete behar izan ditugu, eta horrela “checkpoint” moduko ebaluazio periodikoa jaso dugu, proiektuak jaso beharreko aldaketen inguruan aholkuak jasotzeko: lehenengo sprinteko eginkizunak **Zailtasun mailari dagokion Dragamina osatu** eta **Gelaxka desestali** izan dira, bigarrenekoak **Zailtasun maila bateko sesioa hasi** eta **Gelaxka markatu edo marka kendu** eta hirugarrenekoak **Jokoa geratu, Jokoa jarraitu** eta **Ranking-a erakutsi**.

¹Front-end eta back-end terminologiaren inguruan: kontextuaren arabera, MVC patroia guztia front-endean egon daiteke; gure kasuan, izen hauek ematea erabaki dugu aurretik eta atzetik gertatzen dena bereizteko, hori baita hitzei ematen zaien beste esanahia

3. Diseinua

3.1 Patroiak

Proiektu honetako diseinua egiteko, Software Ingeniaritza ikasgaiaren ikasitako hainbat patroi erabiltzea erabaki dugu.

3.1.1 STRATEGY

Strategy patroia matrizearen laukiak zein motatakoak diren eta zein egoeratan dauden ikusteko erabili dugu.

Hiru lauki mota desberdindu ditugu: mina motako laukiak, hutsune motako laukiak eta zenbaki motako laukiak. Gainera, lauki bakoitzak lau egoera ezberdin izan ditzake: irekita, itxita, bandera jarrita edo galdera ikurra.

Patroi honi esker, matrizeko lauki batean klik egiterakoan, automatikoki lauki honen egoera aldatuko da. Adibidez, klik ezterra erabiliz gero, laukiaren egoera irekita izatera pasako litzateke eta klik eskuina erabilita, egoera berria bandera izango litzateke.

3.1.2 FACTORY

Bestalde, Factory patroia erabili dugu ere. Gure matrizea laukiz osatuta egongo da, eta honen ondorioz, matrizearen hasieraketa gauzatzeko (hau da, mina bakoitzaren sorketa) patroia hau erabiliko dugu. Minak jartzerakoan ere oso lagungarria izan daiteke.

3.1.3 MVC

Azkenik, gure ustez proiektu honetan erabili dugun patroia garrantzitsuenak MVC patroia. Patroi honi esker, ereduak eta bista banatzeko aukera izan dugu.

MVC patroia hiru zatitan banatuta dago:

- Alde batetik, **eredua** dugu. Zati honek, interfaze grafikoan edozein gauza gertatzean, erabakiak hartuko ditu eta Observer-aren bidez, aldaketak hauek interfazean islatuko dira.

- Ondoren, **bista** dugu; hau da, interfaze grafikoa.
- Bukatzeko, **kontroladorea**. Klase honen helburua erreza da. Azken batean, bista eta eredu packageak ezin dira zuzenean komunikatu; honen ondorioz, bitarteko klase bat jarriko da: kontroladorea, hain zuzen ere.

Orain esan dena kontuan hartuz, argi dago kontroladorea Observer bezala lan egingo duela. Honi esker, interfaze grafikoan edozein aldaketa gertatuz gero, ereduak honen berri izango du eta beharrezkoak diren aldaketa egingo ditu. Aldaketa guzti hauek gauzatu ondoren, observer-aren bidez, interfaze grafikoan islatuko dira.

3.1.4 OBSERVER

Azkenik, Observer patroia erabili dugu. Lehen esan den bezala, patroia honi esker, ereduak hartutako erabakiak interfaze grafikoan islatzea lortu dugu.

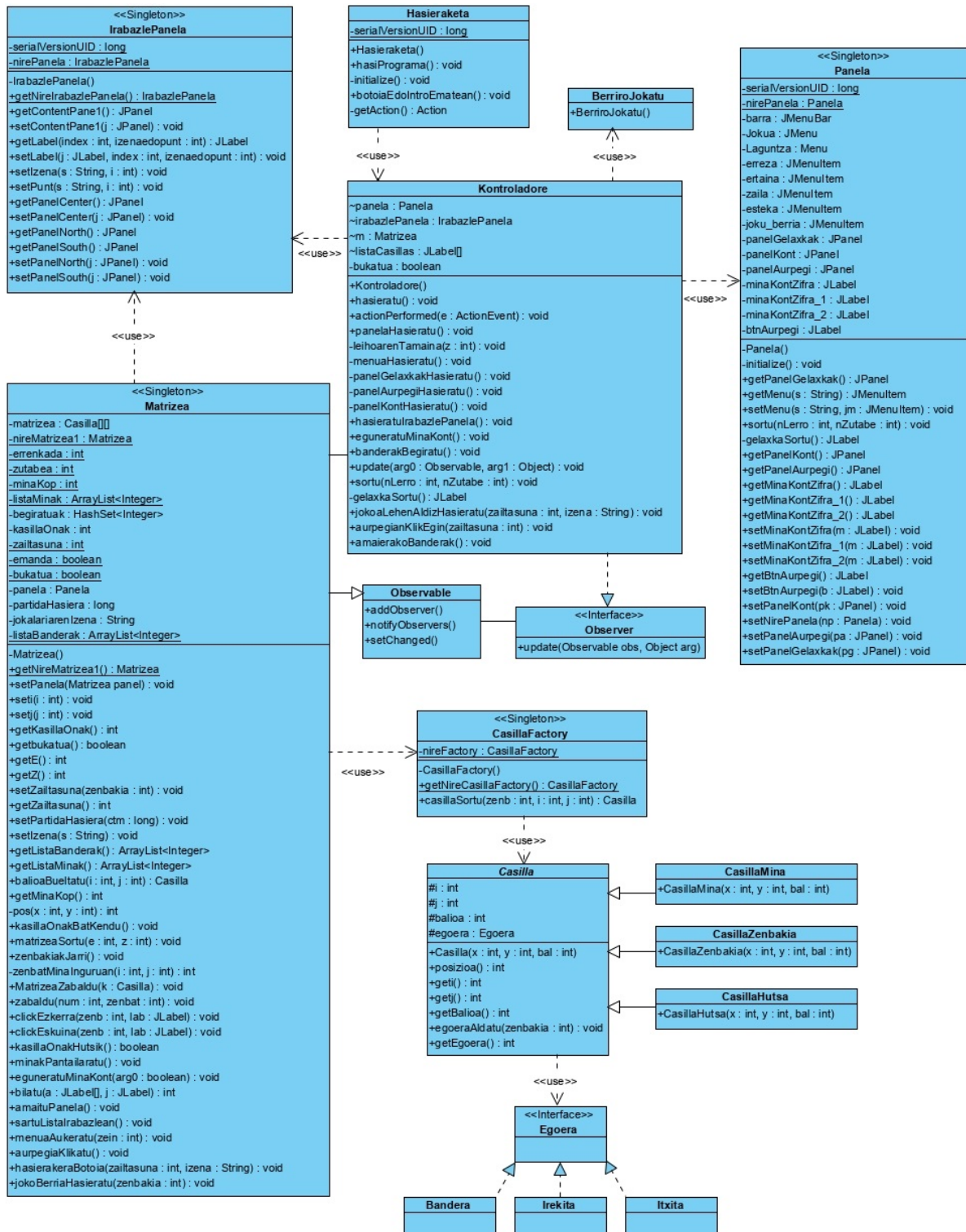
3.2 Klase diagrama

Klase diagraman proiektua aurrera ateratzeko behar izan diren klase guztiak eta haien atributu eta metodoak sartu dira. Klase diagraman erabilitako diseinu patroiak ere argi ikusi daitezke (1. irudia).

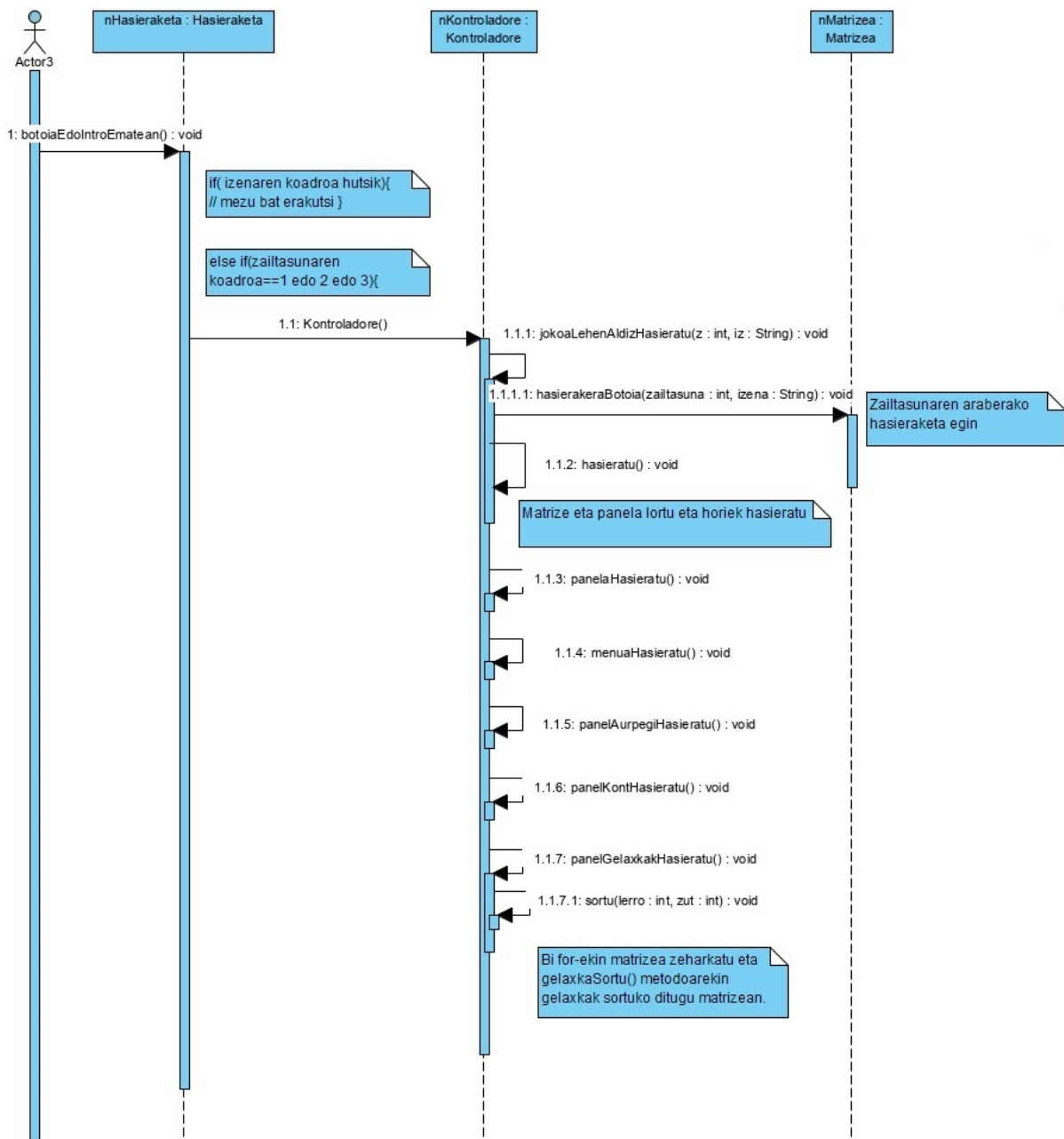
3.3 Sekuentzia diagrama

Sekuentzia diagrametan ekintza konkretu baten garapenaren eskemak ageri dira. Eskema hauek kodearen konplexutasuna errazteko erabiliak dira. Gure proiekturako hiru sekuentzia diagrama sortu ditugu:

- hasieraketa(): Partida hasieratzean burutzen diren ekintzak gordetzeko (2. irudia).
- eskumaClick(): Bitan banatu dugu. Bata, matrizea lehen aldiz klikatzean gertatzen dena azaltzeko (3. irudia); bestea, matrizea jada behin klikatu denerako (4. irudia).
- ezkerClick(): Hiru zatitan banatu dugu sekuentzia diagrama hau. Lehenengoa, matrizea sortzen den kasurako, hau da, lehen aldiz ezker click egitean (5. irudia); bigarrena, partidak irauten duen bitartean: mina, zenbakia edo hutsunea agertuko da klikatzeko (6. irudia); azkena, KasillaOnak 0 denean, minak ez diren kasillak desestali direnean, alegia (7. irudia).



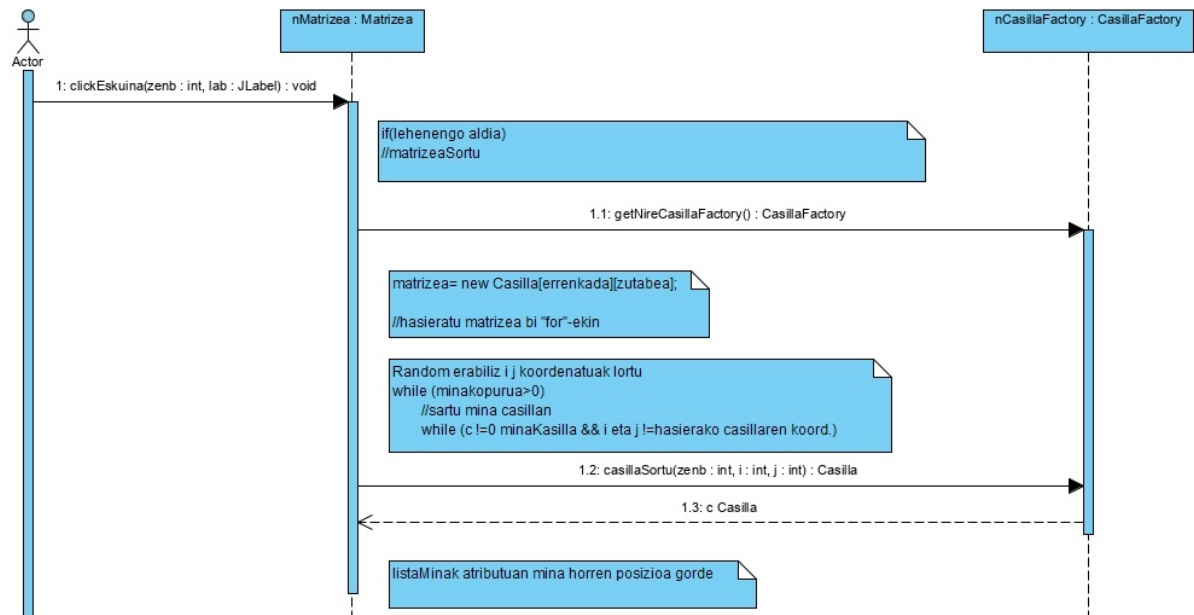
1. irudia: Klase diagrama



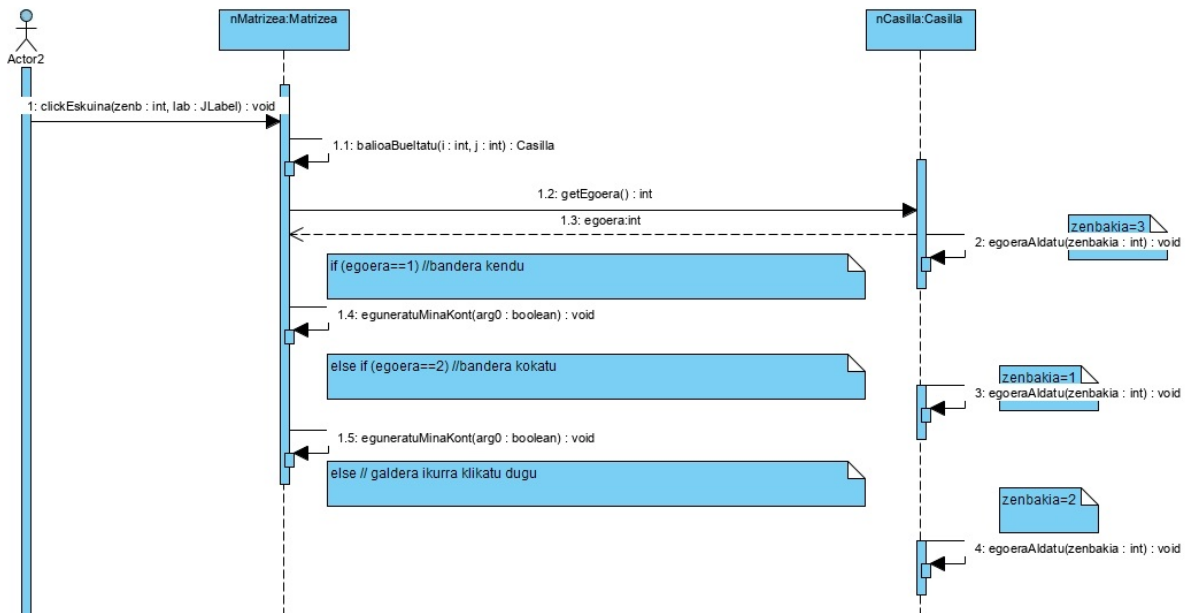
2. irudia: Diagrama honetan jokoaren hasieraketa azalduko dugu. botoiaEdoIntroEmatean() metodotik hasiko gara, izenak dioen moduan lehenengo panelean (izena eta zailtasuna eskatzen den panelean) intro edo botoia sakatzean gertatuko dena.

Lehenik izena hutsik sartu dugun ikusiko dugu if batekin. Izena ondo sartu bada kontroladore klasea lortuko dugu eta bertan hasieraketa egingo dugu erabiltzaileak erabakitako zailtasunaren arabera.

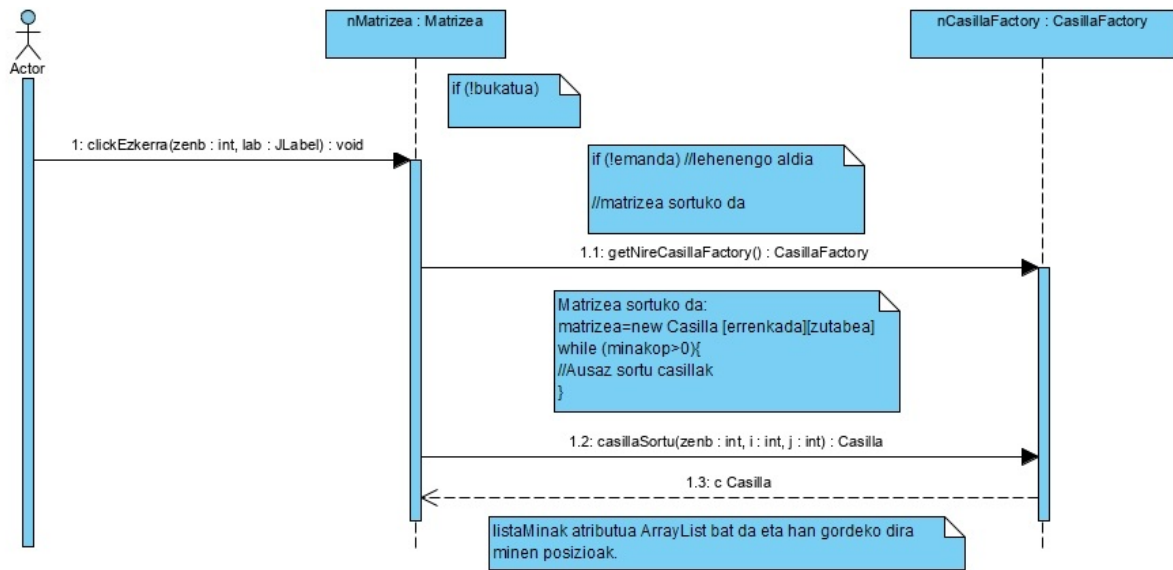
Horren ondoren, kontroladore klasean dauden hasieraketaren hainbat metodo egikarrituko dira; panelaHasieratu(), menuaHasieratu(), panelAurpegiHasieratu(), panelKontHasieratu() eta panelGelaxkakHasieratu(). Azken honen barruan sortu() metodoa daukagu, non matrizea bi “for”-ekin zeharkatu eta gelaxkak sortzen dituen.



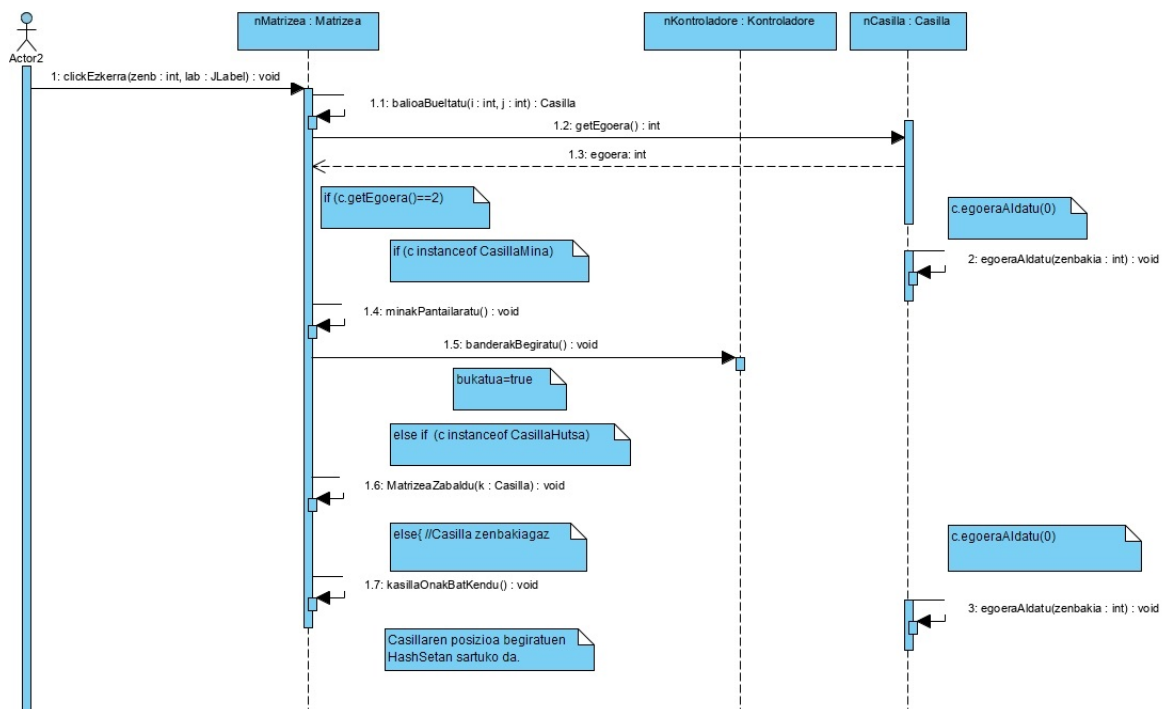
3. irudia: Lehenengo zati honetan matriceaSortu() metodoaren azalpena ageri da. Matricea lehen aldiz klikatzean deitzen zaio metodo honeri, eta horrela minak sakabanatu eta haien inguruko gelaxken zenbakiak lortuko ditugu.



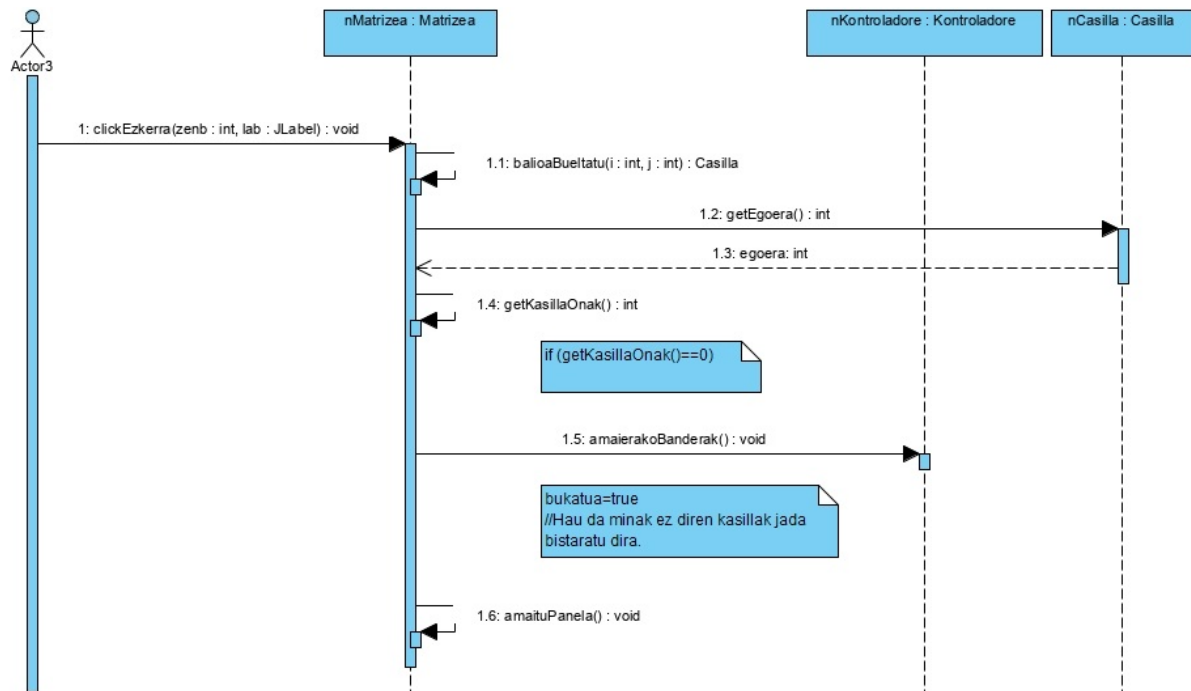
4. irudia: Bigarren zati honetan, matricea jadanik sortuta dagoenean gertatzen dena ikusiko dugu. Klikatu dugun gelaxkaren egoera lortuko dugu getEgoera() metodoarekin, eta honen egoeraren arabera bandera kokatuko/kenduko dugu.



5. irudia: Sekuentzia diagramaren zati honetan, matrizeaSortu() metodoa agertzen da. Jokalariak ezker click lehen aldiz egiterakoan, minak, hutsuneak eta zenbakiak jarri behar dira panelean, hau da, panela hasieratzen da.



6. irudia: Jokoa hasieratuta dago eta oraindik ez da bukatu. Argi eta garbi, hiru kasilla mota daude: hutsunea, zenbakia eta mina. Mina klikatzerakoan partida amaitu da eta minak pantailaratuko dira. Hutsunea denean matrizea zabaltu egingo da eta azkenik zenbakia denean KasillaOnak den atributua -1 egingo dugu (gutxiago falta da irabazteko).



7. irudia: Azken atal hau KasillaOnak atributua 0 denean gertatzen da; horrek esan nahi du partida amaitu egin dela.

3.4 1. sprint

Lehenengo sprintean Dragamina jokoak funtziona dezan beharrezkoak diren metodoak garatu ziren. Lehenik eta behin 'Casilla'-z beteriko matrizea sortu zen. Gainera, ausaz, maila bakoitzarentzat, 'Mina' kopuru bat finkatu zen, [zutabea * zailtasun maila] formula erabiliz. Behin minak eta gelaxka onak matrizean jarrita, gelaxka bakoitzaren inguruan aurkitzen diren mina kopurua pantailaratzea lortu behar zen. Horretarako metodo bat sortu zen. Metodo honek gelaxkako auzokideek ia mina duten begiratzen du. Horrela izanez gero, mina kopurua pantailaratzen da; bestela, iteratzen jarraituko da, minak aurkitu arte, era errekursiboan.

Mina bat klikatuz gero, partida bukatzen da, porrota jasoz.

3.5 2. sprint

Aurreko sprintean aurkitu ziren akatsak zuzendu ziren. Izan ere, MVC diseinu patroia ez zegoen ondo implementatuta. Dena nahasita zegoen, eredia eta bistaren kodeak nahasita zeuden, kodean aurkitzen ziren akats txikiak handiagoetan bueltatzea ahalbidetzen. Kodea

behin bereizita zegoenean eta eredu eta bista pakete ezberdinetan banaturik egonda, bigarren sprinteko atazekin hasi ginen. Hau guztia klase diagramara eraman zen gauzak argi ikusteko, interfazearekin bakarrik erlazioa zuten klaseak eta ereduko klasean alde batean eta bestean utziz.

Lehenengo ataza berrian partida hasieratzeko leihoa sortu behar zen. Bertan, jokalaria-
ren izena sartzeko textfield bat eta jokoaren zailtasuna aukeratzeko beste bat dago. Azken textfield honek hiru balio jasotzen ditu, 1etik 3ra, 1a zailtasun errazena eta 3a zailena izanik. Textfield honetan sartutako balioaren arabera hurrengo leihoan sortutako jokoaren matrizea-
ren dimentsioak ezberdinak izango dira, eta baita mina kopurua, aurreko atazan erabilitako formula jarraituz.

Honetaz aparte, gelaxka markatzeko funtzioa sortu zen. Hau ondo inplementatzeko Strategy patroia erabili da. Honen bitartez, gelaxka bakoitzaren egoera (irekita, itxita, markatuta, galdera ikurrarekin) momentu oro kontrolatu daiteke.

Sprint honetan beste funtzionalitate batzuk gehitu ziren. Joko barruan menu batzuk sortu ziren, 'Jokoa' eta 'Laguntza'. Lehenengoak aukera ezberdinak ematen ditu, haien artean jokoa berriro hastekoa edota zailtasun ezberdineko jokoa hastekoa. 'Laguntza' izeneko menuak laguntza ematen digun esteka bat zabaltzen du. Esteka honetan, jokoak nola funtzionatzen duen azaltzen duen testu bat aurkezten zaigu, baita jolasten ikasteko lagungarria izan daitekeen bideo-tutorial bat ere.

Bukatzeko, 'aurpegia' funtzioa landu zen. Botoi honek joko berria sortzen du sakatzen denean. Gainera, jokoaren egoeraren arabera bere irudia aldatzen doa.

3.6 3. *sprint*

Hirugarren eta azken sprintean kodea ia bukatuta zegoen. Joko bukaerako leihoa sortu eta bere funtzionalitatea programatu behar zen. Leiho honetan, 10 jokalaria onenen taula bat erakusten da, jokalaria bakoitza bere puntuazioarekin. Gainera beste funtzionalitate bat gehitzea eskatzen zen. Funtzio honek, partida bukatzerakoan, zailtasun bereko partida berria hasteko aukera ematen dio jokalaria-ri. Horretarako, bukaerako panelean, irabazleen panelean, aurpegiari emanda azken partidako zailtasun bera zuen partida berria hasieratzen da.

Hau bukatuta, sekuentzia diagramak egitea faltatzen zen, baita azkenengo funtzio hauek egiteko beharrezko metodoak eta klaseak klase diagraman sartzea ere.

Hiru sekuentzia diagrama egin dira. Ezkerreko klika, eskumako klika eta hasieraketa. Bukaerako klase diagraman argi ikusi daitezke erabili diren diseinu patroiak (State, Factory, MVC eta Observer).

4. Garapena

4.1 1. sprint

4.1.1 ENUNTZIATUAN AGERTZEN EZ DIREN FUNTZIONALITATE EDOTA LAN EXTRAK

Sprint honetarako ez genuen funtzionalitate extrarik gehitu; hala ere, 2. sprinterako ziren zeregin batzuk sartu genituen, lana aurreratzeko eginda utzi genituelako.

4.1.2 AURKITUTAKO ARAZOAK + KONPONBIDEAK

Hasieran, 'Matrize' klaseko metodo batzuetan, zenbakiakJarri eta zenbatMinaInguruan, IF multzo handia erabiltzen genituen matrizea errekorritzeko eta aztertzeko. Azkenean, konponbide bezela, horrelako atazak bi for erabiliz askoz hobe egitea zela ikusi genuen, kode lerro gutxiago erabiliz, eta kodearen fidagarritasuna eta eraginkortasuna handituz.

4.1.3 ATAZEN DOKUMENTAZIOA

Sp.	Ataza	Ardu-raduna	Planifik. denbora	Denbora erreal	Aipuak
1	matrizeaSortu() metodoa	Adei	2 ordu	1 ordu eta 30 minutu	Hemen bai matrizea eta bai minen posizioak sortuko dira.
1	GUI (jokoaren panela)	Ander	2-3 ordu oinarrikoa bukatzeko + egin beharreko hobekuntzak	Planifikatu-takoaren antzera	Panela konfiguratu: Gelaxkak kokatu, matrizearekin sinkronizatu, minen kontadorea jarri...
1	Klase Diagrama	Alvaro	2 ordu	Pare bat egun	Azkeneko momentuko aldatetak direla eta klase diagrama hainbat alditan aldatu behar izan du.

1	klikatuak atributua eta bere metodoak	Adei	1 ordu	30 min	Hasieran, ez genuen metodo hau jarriko, baina irabazteko metodoa inplementatzeko beharrezkoa zela iruditu zitzaigun.
1	matrizeaZabaldut() metodoa	Adei, Alvaro, Jon	2-3 ordu + aldaketak	Planifikatutako denbora + hobekuntzak	Hasieran metodo hau if zerrenda luze baten bidez inplementatu genuen.
1	Irabazteko Panela	Ena	30 min	2 egun	Arazoak sortu ziren gelaxkak hutsik zeuden ikusteko.
1	Dokumentazioa	Ena, Jon	30 min	50 min	Dena koordinatu eta dokumentu garrantzitsu hau egin.
1	zenbatMinaInguruan() metodoa	Jon	30 min	1 ordu	2 for erabiltzean metodoa laburragoa egiten da.
1	Menuak eta hauen aukeren inplementazioa	Adei	2 ordu	Planifikatu antzera	Hasieran, bakarrik Joko berria aukera sartu nuen, baina ondoren erreza, ertaina eta zaila aukerak gehitu nituen. Aukera horiekin, Jokoa hasieratu daiteke edozein momentutan.
1	zenbakiakjarri()	Adei, Alvaro	2 ordu	2 ordu eta 30 minutu	Metodo honen barruan, aldaketa batzuk egin genituen, adibidez, zenbat mina inguruan metodoan, zuk proposatutakoa bi for ipintzea.
1	JokuBerriaHasieratu() Hasieraketa klasean	Adei	2 ordu	2 ordu + hobekuntzak	Menuetan dauden aukerak, metodo hauen bitartez inplementatuta daude.

1	minakPantaila-ratu()	Ena	15 min	15 min	ArrayList erabiltzea pentsatu dut izan ere bakarrik minak pantailaratzean datza metodo hau.
1	irabazia()	Ander	15 min	15 min	Banderak agertzen diren metodo simple bat da honako hau.
1	Casilla klasea	Jon	10 min	15 min	Hasieran bandera (boolean) atributua gehitzea ahaztu egin zitzaidan.
1	Laguntza menua	Ander	15 min	30 min	Web orrialdean dragamina joko nola jokatzen den azaltzen du.
1	Metodo txikiak	Guztiak	30 min	45 min	Bigarren mailako metodoak taldekideen artean egin.

4.1.4 PROBEN DOKUMENTAZIOA

4.1.4.1 ID Proba: 2.1

- **Helburua:** Gelaxka desestali.
- **Deskribapena:** Gelaxka baten gainean ezkerreko klik egitean bere azpian duen balioa agertzea.
- **Sarrera:** Ezkerreko klika.
- **Exekuzio baldintza:** Gelaxka estalita egotea.
- **Esperotako emaitza:** Mina izatekotan mina agertzea; Mina ez izatekotan bere auzokideak diren mina kopurua agertzea. Auzokideak diren minarik ez egotekotan errekurtsiboki desestali auzokideak aurkitu arte.
- **Lortutako emaitza:** Froga desberdinak egitean emaitza ezberdinak: Mina klikatzean mina bat agertzen da. Mina auzokideak dituen gelaxka desestaltzean mina kopurua agertzen da, eta azkenik, mina auzokiderik ez dagoenean beharrezko gelaxkak altxatu eta hutsik usten ditu gelaxka horiek.
- **Bukatuta:** Bai.

4.1.4.2 ID Proba: 3.1

- **Helburua:** Zailtasun maila bateko sesioa hasi.
- **Deskribapena:** Erabiltzailea jokoan logeazten denean, bere izena eta zailtasun maila adieraziz, adierazitako zailtasunaren araberako panela sortzea (tamaina eta mina kopuru ezberdinekin).
- **Sarrera:** String eta Integer bat.
- **Exekuzio baldintza:** Zenbaki osoa eta 1-3 tartekoa.
- **Esperotako emaitza:** Hautatutako zailtasun mailaren araberako panela, tamaina eta mina kopuru ezberdinekin.
- **Lortutako emaitza:** Baldintzak betetzen dituen hiru dimentsio ezberdinetako 3 matrize, baina mina kopurua berdina da kasu guztietarako.
- **Bukatuta:** Ez.

4.1.4.3 ID Proba: 3.2

- **Helburua:** Zailtasun maila bateko sesioa hasi.
- **Deskribapena:** Erabiltzailea jokoan logeazten denean, bere izena eta zailtasun maila adieraziz, adierazitako zailtasunaren araberako panela sortzea (tamaina eta mina kopuru ezberdinekin).
- **Sarrera:** String eta Integer bat.
- **Exekuzio baldintza:** Zenbaki osoa eta 1-3 tartekoa.
- **Esperotako emaitza:** Hautatutako zailtasun mailaren araberako panela, tamaina eta mina kopuru ezberdinekin.
- **Lortutako emaitza:** Baldintzak betetzen dituen hiru dimentsio ezberdinetako 3 matrize, bakoitza mina kopuru ezberdinekin (zutabea * zailtasun maila formula erabiliz).
- **Bukatuta:** Bai.

4.2 2. sprint

4.2.1 ENUNTZIATUAN AGERTZEN EZ DIREN FUNTZIONALITATE EDOTA LAN EXTRAK

Joku barruan bi menu jarri ditugu jokalariaen joku esperientzia hobetzeko. Alde batetik 'Joku' menua inplementatu dugu, zailtasun maila batetik bestera azkarrago aldatzea ahalbidetzeko edota partida berria hasteko. Bestetik 'Laguntza' menua landu dugu, duda kasuetan jokalariai lagundu ahal izateko.

4.2.2 AURKITUTAKO ARAZOAK + KONPONBIDEAK

Hasieran, MVC patroiarekin arazoak izan genituen ez genekielako zein klaseetan sartu hainbat metodo. Kotsultak egin ondoren, ordena zuzena kokatu dugula uste dugu.

Bestalde, Observer patroiak ere arazo asko sorrarazi dizkigu. Hemen ere, arazoa ondo gainditu dugula uste dugu.

4.2.3 ATAZEN DOKUMENTAZIOA

Sp.	Ataza	Ardu-raduna	Planifik. denbora	Denbora erreal	Aipuak
2	GUI	Ander	15 minutu	15 minutu	2. sprint honetarako, banderak sartu behar-ko ziren interfazeaz, gainontzekoa konfiguratze-ko.
2	Observer	Adei	3 ordu	4 ordu	Patroi hau erabili du-gu ereduaren eta bista-ren artean menpekota-sunik ez egoteko
2	MVC	Ander, Alvaro, Jon	4 ordu	4 ordu	MVC patroia-aren bidez, eredu eta bista banatu ditugu. Honi esker, ob-server patroia erabiltze-ko aukera izan dugu
2	Klase diagrama	Alvaro	ordu 1	2 ordu	Klase Diagrama proiektuan aldatutako gauze-kin eguneratu dugu.

2	STATE & FACTORY	Adei, Ena	3 ordu	5 ordu	1.sprint-ean, matrizearen posizio bakoitzean, zenbaki bat jarri genuen, baina orain kasillak ditugu jarrita. Horretarako STATE-a eta FACTORY erabili dugu
2	Banderak begiratu	Ena	15 minutu	15 minutu	Partida bat galtzen dugunean, bandera bat mina ez zegoen kasilla batean jarri ezker, mina gurutzearekin agertuko da kasilla horretan.
2	Eguneratu mina-Kont	Ander	20 minutu	20 minutu	Metodo hau jarri dugu, partida galtzean, mina kontadorean “- - -” agertzeko.

4.2.4 PROBEN DOKUMENTAZIOA

4.2.4.1 ID Proba: 1.1

- **Helburua:** Eskatutako zailtasun mailako panela sortu eta erakutsi.
- **Deskribapena:** Hautatutako maila zenbakiaren arabera dimentsio ezberdineko panelak lortu.
- **Sarrera:** Zenbaki oso bat.
- **Exekuzio baldintza:** Zenbakia osoa eta 1-3 tartean.
- **Esperotako emaitza:** Dimentsio ezberdinetako matrizeak.
- **Lortutako emaitza:** Baldintzak betetzen dituen hiru dimentsio ezberdinetako 3 matrize.
- **Bukatuta:** Bai.

4.2.4.2 ID Proba: 4.1

- **Helburua:** Gelaxka markatu edo marka kendu.
- **Deskribapena:** Gelaxka bat eskumako botoiarekin klikatzean, mina estalita egonda, marka bat jarri edo kenduko zaio.
- **Sarrera:** Eskumako klik-a.
- **Exekuzio baldintza:** Mina estalita egotea.
- **Esperotako emaitza:** Gelaxka estalita egonda, lehenengo eskubiko botoiarekin bandera jartzea eta bigarrenarekin kentzea.
- **Lortutako emaitza:** Lehenengo klik-ak bandera jarri, baina hori ezin daiteke kendu, horrela geratzen da.
- **Bukatuta:** Ez.

4.2.4.3 ID Proba: 4.2

- **Helburua:** Gelaxka markatu edo marka kendu.
- **Deskribapena:** Gelaxka bat eskumako botoiarekin klikatzean, mina estalita egonda, marka bat jarri edo kenduko zaio.
- **Sarrera:** Eskumako klik-a.
- **Exekuzio baldintza:** Mina estalita egotea.
- **Esperotako emaitza:** Gelaxka estalita egonda, lehenengo eskubiko botoiarekin bandera jartzea eta bigarrenarekin kentzea.
- **Lortutako emaitza:** Lehenengo klik-ak bandera jarri eta bigarrenarekin kendu. Ostera, bandera duenean ezkerreko klikarekin klik eginez gero gelaxka desestaltzen da.
- **Bukatuta:** Ez.

4.2.4.4 ID Proba: 4.3

- **Helburua:** Gelaxka markatu edo marka kendu.

- **Deskribapena:** Gelaxka bat eskumako botoiarekin klikatzean, mina estalita egonda, marka bat jarri edo kenduko zaio.
- **Sarrera:** Eskumako klik-a.
- **Exekuzio baldintza:** Mina estalita egotea.
- **Esperotako emaitza:** Gelaxka estalita egonda, lehenengo eskubiko botoiarekin bandera jartzea eta bigarrenarekin kentzea.
- **Lortutako emaitza:** Lehenengo klik-ak bandera jarri eta bigarrenarekin kendu. Ezin daiteke desestali banderadun gelaxka. Bandera bat duen gelaxkan eskumako klikarekin klik eginez galdera ikurra jarri. Bandera kontagailuak ondo funtzionatzen du.
- **Bukatuta:** Bai.

4.3 3. sprint

4.3.1 ENUNTZIATUAN AGERTZEN EZ DIREN FUNTZIONALITATE EDOTA LAN EXTRAK

Strategy patroian aldaketa bat egin genuen. Izan ere, irudi packean galdera ikurdun blokea ikusi genuen eta jokoan sartzea erabaki genuen; blokea bloke normal bat bezala portatuko da, ezkerreko klikarekin altxatu ahalko baita, baina galdera ikurraren marka izango du; izan ere, jokalaria dudan balego berarentzat baliagarria izan daiteke.

4.3.2 AURKITUTAKO ARAZOAK + KONPONBIDEAK

Irabazle panelarekin hainbat arazo izan ditugu: Izena “null” agertzen zen, puntuazioarekin arazoak zeudenez irabazlea ez zen agertzen...

4.3.3 ATAZEN DOKUMENTAZIOA

Sp.	Ataza	Ardu-raduna	Planifik. denbora	Denbora erreala	Aipuak
3	Jokoa geratu eta partida berria hasi	Adei eta Alvaro	4 ordu	3 ordu	Jokalariak nahi duenean partida berria hasi dezake aurpegia erabiliz edota menua erabiliz.

3	Jokoan jarraitu	Adei eta Ander	ordu 1	45 minutu	Partida galduta aurpegi botoian, edo partida irabazita jarraitu botoian klik eginez, JDialog bat agertuko da esanez partida jarraitu nahi duzun ala ez.
3	Irabazleen panela	Ander eta Ena	30 minutu	3 ordu	Enak panela eraiki du eta Anderrek irabazleen listaren kudeaketaz enkargatu da.
3	Proben dokumentazioa	Alvaro eta Ena	30 minutu	30 minutu	Prozesu guztian egindako probak bildu eta dokumentatu, euren emaitzekin. Hainbat proba egon dira, emaitza onak eman dituztenak eta ikasten edo hobetzen lagundu digutenak.
3	Klase diagrama	Alvaro	30 minutu	1 ordu	Behin behineko kodea bukatu ahal izateko egindako aldaketak klase diagraman jarri.
3	Hasieraketaren sekuentzia diagrama	Ena	30 minutu	1 ordu	Metodorik garrantzitsuenak soilik agertuko dira.
3	Gelaxka desestaltzearen sekuentzia diagrama	Jon	40 min	1h 30 min	Hiru zatitan banatzean hobeto ulertzen da.
3	Gelaxka markatzearen sekuentzia diagrama	Ena	20 minutu	1 ordu	Handiegi ez geratzeko bitan banatu da.
3	Dokumentazioa idatzi	Guztiok	6 ordu	8 ordu	Bakoitzak bere banakako lana azaldu du eta beste guztia denon artean idatzi dugu.
3	Dokumentazioa maketatu	Ander	2 ordu	3 ordu	Dokumentazioa polit eta dotore geratzeko LaTeX erabili dugu

4.3.4 PROBEN DOKUMENTAZIOA

4.3.4.1 ID Proba: 5.1

- **Helburua:** Erabiltzaileak nahi duenean partida bukatu eta maila bereko joko berria hasi.
- **Deskribapena:** Erabiltzaileak partidaren edozein momentutan joko hori utzi eta berri bat hasteko aukera dauka. Partida berri hori bertan behera utzitakoaren maila berekoa izango da beti.
- **Sarrera:** Integer bat.
- **Exekuzio baldintza:** Zenbaki osoa eta 1-3 tartekoa (zailtasunaren arabera).
- **Esperotako emaitza:** Hautatutako zailtasun mailaren arabera panela, tamaina eta mina kopuru ezberdinekin.
- **Lortutako emaitza:** Joko berria leihorik berri batean hasi da leihorik berdinean hasi behar zenean.
- **Bukatuta:** Ez.

4.3.4.2 ID Proba: 5.2

- **Helburua:** Erabiltzaileak nahi duenean partida bukatu eta maila bereko joko berria hasi.
- **Deskribapena:** Erabiltzaileak partidaren edozein momentutan joko hori utzi eta berri bat hasteko aukera dauka. Partida berri hori bertan behera utzitakoaren maila berekoa izango da beti.
- **Sarrera:** Integer bat.
- **Exekuzio baldintza:** Zenbaki osoa eta 1-3 tartekoa (zailtasunaren arabera).
- **Esperotako emaitza:** Hautatutako zailtasun mailaren arabera panela, tamaina eta mina kopuru ezberdinekin.
- **Lortutako emaitza:** Joko berria hastean eta gela klikatzean matrizeak ez zituen ondo desestaltzen.
- **Bukatuta:** Ez.

4.3.4.3 ID Proba: 5.3

- **Helburua:** Erabiltzaileak nahi duenean partida bukatu eta maila bereko joko berria hasi.
- **Deskribapena:** Erabiltzaileak partidaren edozein momentutan joko hori utzi eta berri bat hasteko aukera dauka. Partida berri hori bertan behera utzitakoaren maila berekoa izango da beti.
- **Sarrera:** Integer bat.
- **Exekuzio baldintza:** Zenbaki osoa eta 1-3 tartekoa (zailtasunaren araberakoa)
- **Esperotako emaitza:** Hautatutako zailtasun mailaren araberako panela, tamaina eta mina kopuru ezberdinekin.
- **Lortutako emaitza:** Partida berria hasteko beharrezkoa den matrize eta mina kopuruak zehaztuta (aukeratutako zailtasunaren araberakoa).
- **Bukatuta:** Bai.

4.3.4.4 ID Proba: 6.1

- **Helburua:** Jokoa jarraitu.
- **Deskribapena:** Dragamina bat bukatzerakoan, erabiltzaileak jolasten jarraitzeko aukera dauka, betiere maila berean.
- **Sarrera:** Klik bat.
- **Exekuzio baldintza:** Partida erdian edo bukaeran egotea eta dialogean bai botoia sakatzea.
- **Esperotako emaitza:** Joku berria hastea, maila berekoa.
- **Lortutako emaitza:** Joku berria hasten da, baina erabiltzaileak sartutako izena galtzen da.
- **Bukatuta:** Ez.

4.3.4.5 ID Proba: 6.2

- **Helburua:** Jokoan jarraitu.
- **Deskribapena:** Dragamina bat bukatzerakoan, erabiltzaileak jolasten jarraitzeko aukera dauka, betiere maila berean.
- **Sarrera:** Klik bat.
- **Exekuzio baldintza:** Partida erdian edo bukaeran egotea eta dialogean bai botoia sakatzea.
- **Esperotako emaitza:** Joku berria hastea, maila berekoa.
- **Lortutako emaitza:** Joku berria hasten da eta ez du arazorik ematen.
- **Bukatuta:** Bai.

4.3.4.6 ID Proba: 7.1

- **Helburua:** 10 erabiltzaile hoberenen puntuazioa erakutsi.
- **Deskribapena:** Fitxategi batean gordeko ditugu erabiltzaileen puntuazioak eta lehenengo 10-ak partida irabaztean agertuko dira.
- **Exekuzio baldintza:** Partida irabazi izana.
- **Esperotako emaitza:** Jokoa jolastutako 10 erabiltzaile hobearen puntuazioak erakustea, eta jolastutako azken partidaren puntuazioa 10 hoberenatarikoa bada bertan sartzea.
- **Lortutako emaitza:** Jolasten ari den erabiltzailearen izena ez da irabazleen panelean agertzen nahiz eta 10 hoberenatarikoa izan.
- **Bukatuta:** Ez.

4.3.4.7 ID Proba: 7.2

- **Helburua:** 10 erabiltzaile hoberenen puntuazioa erakutsi.
- **Deskribapena:** Fitxategi batean gordeko ditugu erabiltzaileen puntuazioak eta lehenengo 10-ak partida irabaztean agertuko dira.

- **Exekuzio baldintza:** Partida irabazi izana.
- **Esperotako emaitza:** Jokoa jolastutako 10 erabiltzaile hobearen puntuazioak erakustea, eta jolastutako azken partidaren puntuazioa 10 hoberenatarikoa bada bertan sartzea.
- **Lortutako emaitza:** Jolasten ari den erabiltzailearen izena puntuazioarekin batera irabazleen panelean agertu da.
- **Bukatuta:** Bai.

5. Ondorioak

Proiektu hau egiten denbora asko inbertitu ostean hainbat gauza ikasi ditugu. Haien artean, talde lana egitea, hau da, lana banatzea eta bata besteari laguntzea arazo baten aurrean. Gure artean oso ondo lan egin dugu, bakoitzak bere trebetasunak konpartitzen eta bere ahultasunak gogortzen saiatuz. Lan banaketa egokia egin dugula uste dugu, eta bata besteari laguntzeko prest egon gara. Mutilok beste bi proiektuetan ere batera egon gara, eta hori ere abantaila handia izan da.

Proiektuaren hasieran, 1. sprintean, ez genuen jokia irakasleak eskatutako moduan bideratu, baina denbora nahikoa izan genuen konpontzeko. Gaur egun ditugun ezaguerekin proiektua hobetuezina dela esango genuke. Hurrengo urtean edo etorkizunean proiektu honetara bueltatzen bagara, ziur gaude hobetzeko gauza asko dituela konturatuko ginatekela.

Amaitzeko, joko hau garatuz asko ikasi dugu eta gure ikasketa-prozesuan garrantzitsua izan dela deritzogu. Gainera, jokia probatzerakoan oso ondo pasatu dugu!

Erreferentziak

- [1] Marcos, Laura. Historia y origen del clásico Buscaminas. muyinteresante.es, 2018. URL <https://www.muyinteresante.es/tecnologia/video/historia-y-origen-del-clasico-buscaminas>.
- [2] GitHub. anderra57/DRAGAMINA. github.com, 2020. URL <https://github.com/anderra57/DRAGAMINA>.