

KUDEAKETAREN ETA INFORMAZIO SISTEMEN
INFORMATIKAREN INGENIARITZAKO GRADUA
GRADU AMAIERAKO LANA

HIEROGLIFOAK

Ikaslea: Uribarri Diez Ander

Zuzendaria: Mikel Villamañe Gironés

Ikasturtea: 2019-2020

Data: Bilbo, 2020ko urtarrilaren 20a

9780130330391



Universidad
del País Vasco

Euskal Herriko
Unibertsitatea

BILBOKO
INGENIARITZA
ESKOLA
ESCUELA
DE INGENIERÍA
DE BILBAO

Aurkibidea

1	Sarrera	11
1.1	Hieroglifoak, programazioaren funtsa ikasteko jokoak	11
1.2	Zergatik aukeratu dut proiektu hau?	12
2	Hasierako planteamendua	13
2.1	Helburuak	13
2.2	Aurrekariak	14
2.2.1	CodeMonkey	14
2.2.2	Code.org	14
2.2.3	Scratch	15
2.2.4	Aurrekarien laburpena	15
2.3	Bizi zikloa	15
2.4	Lanaren deskonposaketa	17
2.5	Denbora-plangintza	18
2.6	Gantt diagrama	19
2.7	Erraminten aukeraketa	20
2.8	Software baliabideak	21
2.9	Hardware baliabideak	22
2.10	Arriskuak	22
2.10.1	Planifikazio ez aproposa	22
2.10.2	Ezagutza faltarekin lotutako arazoak	22
2.10.3	Informazioa galera	23
2.10.4	Arazo pertsonalak	23
2.11	Bideragarritasun ekonomikoa	23
2.11.1	Eskulan gastuak	23
2.11.2	Software gastuak	24
2.11.3	Hardware gastuak	25
2.11.4	Beste gastu batzuk	25
2.11.5	Kostu totala	26
2.11.6	Return Of Investment	26
3	Betekizunen bilketa	27
3.1	Erabilpen kasuen eredia	27
3.2	Domeinuen eredia	28
4	Analisia eta diseinua	31
5	Unity	35
5.1	Objektuak	35

5.2	Eszenatokia.....	35
5.3	Collider eta Rigidbody	36
5.4	Script	37
6	Garapena eta ebaluazioa.....	39
6.1	Jokalari mugimendua	39
6.1.1	Talkak hautematea.....	39
6.1.2	Erabiltzailearen input-a	41
6.1.3	Jokalaria mugitu	41
6.1.4	Animazioak	42
6.1.5	Kamera	43
6.1.6	Probak	45
6.1.6.1	Talkak hautematea.....	45
6.1.6.2	Erabiltzailearen input-a eta jokalariaren mugimendua	47
6.1.6.3	Animazioak.....	52
6.1.6.4	Kamera	54
6.2	Mapa eta bizitzak	55
6.2.1	Lur-eremua.....	55
6.2.2	Bizitza sistema.....	56
6.2.3	<i>CheckPoint</i> -ak.....	58
6.2.4	Probak	59
6.2.4.1	Lur eremua	59
6.2.4.2	Bizitza sistema	59
6.2.4.3	Checkpoint-ak.....	60
6.3	Eraso sistema	61
6.3.1	Jokalaria.....	61
6.3.2	Etsaiak	62
6.3.3	Probak	64
6.3.3.1	Jokalaria.....	64
6.3.3.2	Etsaiak	65
6.4	<i>Drag</i> eta <i>Drop</i> funtzionalitateak.....	67
6.4.1	Probak	71
6.5	Ariketak	73
6.5.1	Ariketa kudeatzailea.....	73
6.5.2	Gorde/Kargatu funtzioak.....	74
6.5.3	Menuak	74
6.5.4	Probak	75

6.5.4.1	Ariketa kudeatzailea.....	75
6.5.4.2	Gorde/Kargatu funtzioak.....	76
6.5.4.3	Menuak	77
6.6	Oztopoak	78
6.6.1	Ariketa <i>trigger</i> -ak	78
6.6.2	Soinuak	79
6.6.3	Probak	80
7	Ondorioak.....	83
7.1	Helburuen betekizuna	83
7.2	Planteamendu aldaketak.....	84
7.3	Kodearen kudeaketa	85
7.4	Zer ikasi dut	86
7.5	Zer hobetu dezaket	86
8	Bibliografia	89
9	Eranskinak	91
9.1	Erabilpen kasuen azterketa sakona.....	91
9.1.1	Hasierako menua.....	91
9.1.2	Ariketak ebazteko interfazea	92
9.1.3	<i>Pause</i> menua	93
9.1.4	Jokoaren portaera orokorra	95
9.2	Sekuentzia diagramak	96
9.2.1	Partida hasi.....	96
9.2.2	Ariketa jarri.....	97
9.2.3	Ariketa emaitza konprobatu.....	98
9.2.4	Mugitu	99

Irudien aurkibidea

1. IRUDIA: BIZI ZIKLOA	16
2. IRUDIA: LDE DIAGRAMA.....	17
3. IRUDIA: GANTT DIAGRAMA	20
4. IRUDIA: UNITY ERABILPEN BALDINTZAK	24
5. IRUDIA: ERABILPEN KASUEN DIAGRAMA	27
6. IRUDIA: DOMEINUEN EREDUA	29
7. IRUDIA: KLASE DIAGRAMA	31
8. IRUDIA: TALKA MATRIZEAREN ADIBIDEA	36
9. IRUDIA: IZPIAK IGORTZEN DITUEN LAUKIA	40
10. IRUDIA: IZPIEN JATORRIA KALKULATZEKO FUNTZIOA	40
11. IRUDIA: KOLPE INFORMAZIOA JAKITEKO DATU EGITURA	40
12. IRUDIA: ERABILTZAILEAREN INPUT-A JASOTZEKO METODOA	41
13. IRUDIA: JOKALARIAK ALDAPAK JAISTEKO FUNTZIOA.....	42
14. IRUDIA: ANIMATUTAKO JOKALARIA ALDAPA IGOTZEN	43
15. IRUDIA: KAMERAREN ERABILERA ADIBIDEA	43
16. IRUDIA: PERSPEKTIBA KAMERA ETA KAMERA ORTOGRAFIKOA.....	44
17. IRUDIA: KAMERAREN MUGIMENDUA KUDEATZEKO ESPARRUAK	44
18. IRUDIA: KAMERA MUGITU DAITEKEEN MUGA	45
19. IRUDIA: TILESET-A EDO IRUDI SORTA.....	55
20. IRUDIA: TILEMAP MAPA ADIBIDEA.....	55
21. IRUDIA: TILE BATEN COLLIDER OSAGAIA	55
22. IRUDIA: MAPA ADIBIDEA	56
23. IRUDIA: KOLPEGAINAZALAK GERUZA MULTZOA SOILIK ANTZEMATEKO FUNTZIOA.....	56
24. IRUDIA: JOKALARIAREN BIZITZA INTERFAZEA.....	57
25. IRUDIA: INBENTARIOAREN SINGLETON PATROIA.....	57
26. IRUDIA: JOKALARIA KONTAKTUAN MINTZEKO FUNTZIOA.....	57
27. IRUDIA: BIHOTZAK BERRESKURATZEKO ITEMAK.....	58
28. IRUDIA: CHECKPOINT-A ALDATZEKO KODEA	58
29. IRUDIA: ITEM INTERFAZEA.....	62
30. IRUDIA: SAGUZAR ETSIAK	62
31. IRUDIA: MOMIA ETSAIA	62
32. IRUDIA: KAKALARDOA.....	63
33. IRUDIA: ESKORPIOI ETSAIA	63
34. IRUDIA: SOLDADU ETSAIA	63
35. IRUDIA: KARTA JOKOA.....	67
36. IRUDIA: ARIKETAKO DRAG ETA DROP OSAGAIK	68
37. IRUDIA: AGINDUA MOTA OKERREKO HUTSUNEAN	68
38. IRUDIA: AGINDUA MOTA ZUZENEKO HUTSUNEAN	68
39. IRUDIA: 9-SLICE TEKNIKA PIEZA BATEKIN.....	69
40. IRUDIA: ARIKETAREN EMAITZA KALKULATZEKO FUNTZIOA	70
41. IRUDIA: ARIKETA ADIBIDEA	70
42. IRUDIA: PARTIDAREN EGOERA GORDETZEKO KLASEA.....	74
43. IRUDIA: ARIKETAK JARTZEKO FUNTZIOA.....	79
44. IRUDIA: ARIKETA BATEN TRIGGER-A.....	79
45. IRUDIA: HASIERAKO MENUA FLUXU DIAGRAMA	91
46. IRUDIA: JOLASTU/IRTEN PANTAILA	92

47. IRUDIA: JARRAITU/HASI PANTAILA	92
48. IRUDIA: ARIKETA PANTAILAREN ADIBIDEA	92
49. IRUDIA: PAUSE MENUA	93
50. IRUDIA: HIZTEGIA PANTAILA.....	94
51. IRUDIA: MENUA NAGUSIRA ITZULTZEKO PANTAILA.....	94
52. IRUDIA: JOKOAREN PORTAERA OROKORRA FLUXU DIAGRAMA	95
53. IRUDIA: PARTIDA HASTEKO SEKUENTZIA DIAGRAMA.....	96
54. IRUDIA: ARIKETA JARTZEKO SEKUENTZIA DIAGRAMA	97
55. IRUDIA: ARIKETAREN EMAITZA KONPROBATZEKO SEKUENTZIA DIAGRAMA	98
56. IRUDIA: JOKALARIA ESKUMARA MUGITZEKO SEKUENTZIA DIAGRAMA.....	99

Taulen aurkibidea

1. TAULA: KUDEAKETA DENBORA-PLANGINTZA	18
2. TAULA: IKERKETA DENBORA-PLANGINTZA	18
3. TAULA: BETEKIZUNEN BILKETA DENBORA-PLANGINTZA	18
4. TAULA: ANALISIA ETA DISEINUA DENBORA-PLANGINTZA	19
5. TAULA: INPLEMENTAZIOA DENBORA-PLANGINTZA.....	19
6. TAULA: DENBORA-PLANGINTZA GUZTIRA	19
7. TAULA: HARDWARE GASTUAK	25
8. TAULA: GASTU TOTALA	26
9. TAULA: TALKAK HAUTEMATEKO PROBEN EMAITZAK	46
10. TAULA: TALKAK HAUTEMATEKO ERROREA EMAN DUTEN PROBEN ERREPIKAPENA.....	47
11. TAULA: JOKALARIAREN MUGIMENDU EGOKIRAKO PROBEN EMAITZAK.....	47
12. TAULA: JOKALARIAREN MUGIMENDUAN ERROREA EMAN DUTEN PROBEN ERREPIKAPENA	51
13. TAULA: JOKALARIAREN MUGIMENDUAN ERROREA EMAN DUEN PROBAREN ERREPIKAPENA	51
14. TAULA: JOKALARIAREN ANIMAZIO PROBEN EMAITZAK.....	52
15. TAULA: JOKALARIA ANIMATZEKO ERROREA EMAN DUTEN PROBEN ERREPIKAPENA	54
16. TAULA: KAMERA PROBEN EMAITZAK.....	54
17. TAULA: BIZITZA SISTEMAREN PROBEN EMAITZAK	60
18. TAULA: CHECKPOINT-EN PROBEN EMAITZAK	61
19. TAULA: JOKALARIAREN ERASOEN PROBEN EMAITZAK	64
20. TAULA: JOKALARIAREN ERASORAKO ERROREA EMAN DUEN PROBAREN ERREPIKAPENA	65
21. TAULA: ETSAIEN MUGIMENDU ETA ERASOEN PROBEN EMAITZAK	65
22. TAULA: ETSAIEN MUGIMENDU ETA ERASOAN ERROREA EMAN DUTEN PROBEN ERREPIKAPENA.....	66
23. TAULA: DRAG AND DROP PROBEN EMAITZAK	71
24. TAULA: DRAG AND DROP SISTEMAN ERROREA EMAN DUTEN PROBEN ERREPIKAPENA.....	73
25. TAULA: ARIKETEN PROBEN EMAITZAK	76
26. TAULA: PARTIDA GORDE/KARGATU PROBEN EMAITZAK.....	76
27. TAULA: MENUEN PROBEN EMAITZAK.....	77
28. TAULA: ARIKETA, TRIGGER-EN ETA SOINUEN PROBEN EMAITZAK.....	80
29. TAULA: ARIKETA, TRIGGER ETA SOINUAN ERROREA EMAN DUEN PROBAREN ERREPIKAPENA	81
30. TAULA: PROIEKTUAREN BENETAKO IRAUPENA	84

1 Sarrera

Teknologia etengabe garatzen da. Gaur egun ohikoak dira telefono, telebista eta hozkailu adimenduak, amazon eta google-en laguntzaile birtualak edo bakarrik dabiltzan xurgagailuak. Gailu adimenduak etengabe hazten dira eta egungo gizartean etengabe erabiltzen ditugu.

Hezkuntzan ordea, berandu heldu da bizi dugun iraultza teknologikoa. 80-ko hamarkadan, Atenea proiektua [1] martxan jarri zen informatika eskola eta ikastoletan sartzeko. Zoritzarrez, informatikaren inplementazioa ofimatikaren inguruan eraiki da programazioa baztertuz. Ikasgelak pixkanaka digitalizatu dira eta koadernoak ordenagailu eramangarriengatik ordezkutzen ari dira. Teknologiaren erabilpen egokia sustatu da, ikasleak badakite informazioa Interneten bilatzen, testu editoreak edo presentazioak egiteko aplikazioak erabiltzen, baina, atzetik dagoen logika eta funtzionamenduaren berri ez daukate.

Programazioa irakasgaiak, programazio lengoaiak eta gailuen funtzionamendua irakasteaz gain, arazoaren ebazpena eta hauek ebazteko sistemak diseinatzea, logika, elkarlana eta ikaslearen sormena bezalako gaitasunak garatzea du helburu [2].

Hezkuntza gizartearekin bilakatzen da, eduki berriak irakasten diren bezala teknologiaren garapenak erraminta berriak ekarri ditu baita ere. Azken urteetan ikusten ari garen proposamen berria hezkuntzarako jokoak (edo *Game-Based Learning-a*) dira. Joko hauek ikaslearen arreta bereganatzen dute, baina benetako xedea irakastea eta trebetasun berriak bereganatzea da. Metodologia berri hau inplementatu diren ikastetxeetan ikasgaia bereganatzeaz gain ikasleen parte-hartzea eta elkarlana sustatzen dela konturatu dira [3] [4].

1.1 Hieroglifoak, programazioaren funtsa ikasteko jokoak

Proiektu hau, Hieroglifoak, programazioaren munduan lehen pausoak emateko jokoak dira. Edozein kode zati idazteko ezinbestekoak diren elementuak eta portaera egokiak zein diren erakutsiko dira eta zergatia azalduko da. Erabiltzaileak bukatu gabeko jokoan aurkituko dira eta azaltzen diren arazoei aurre egiteko falta diren kode zatiak bete beharko dituzte. Horretarako, baldintzak, aldagaiak, begiztak, zerrendak edo funtzioak dituzten ezaugarriak eta funtzionamendua azalduz.

Jokalaria aurrera jarraitu ezin duen egoera batean aurkituko da. Arazoa ebazteko, programaturik gabe dagoen funtzio bat bukatu beharko da. Horretarako, puzzle pieza itxura duten kode lerroak eskainiko dira. Piezak bultzatuz kodea osotuko daiteke eta emaitza zuzena bada, jokoan aurrera egin ahal da arazo berri bat topatu arte.

Jokoan aurrera egin ahala arazoak zailagoak izango dira eta hauek ebazteko elementu berriak eskainiko dira. Horrela, erabiltzaileak programatzeko beharrezkoak diren gaitasunak bereganatuko ditu jolasten ari den bitartean.

1.2 Zergatik aukeratu dut proiektu hau?

Proiektu hau aukeratzeko arrazoi bat teknologiak gizartean duten garrantzia da. Teknologia etengabe garatzen eta hazten ari da, ondorioz, Informazio eta Komunikazio Teknologikoekin (IKT) loturak dituzten lanpostuak gero eta gehiago dira. Europar batasunak 2014. urtean Hezkuntza ministroei bidalitako gutunean, eskoletan programazio trebetasunak sustatzea eskatzen zen, 2020. urte bukaerarako IKT-ekin lotutako 900.000 enplegu inguru espero direlako eta gazteen langabezia murrizteko balio dezakelako [5] [6]. Hezkuntza zentroetan fisika, kimika, biologia, matematika eta ingelesa bezalako arloak irakasten dira hauek gure bizitzako parte direlako eta horregatik, teknologiaz inguraturik gaudenez, teknologiaren funtzionamendua ere irakatsi behar litzatekela uste dut. Izan ere, programazioa berandu ematea edo ez ematea zenbait kalte ekartzen ditu. Adibidez, IKT-ekin lotutako lanpostuetarako formakuntza informatikoa ez izatea, errekurtsibitatea bezalako kontzeptuak ulertzeko zailtasunak edo pentsamendu konputazionala ez garatzea [7].

Beste alde batetik, *Game-Learning*-a oso erakargarria iruditzen zait. *Game-Learning*-a, bideo joko itxura dauka ikaslea motibatzeke, ikasgaia ulertzeko beharrezkoa den eduki teorikoa dauka eta teoria praktikan jartzeko erronkak. Joko bidezko irakaskuntzak abantaila asko ditu: ikaslearen arreta bereganatu, motibazioa areagotu, ikaslearen erabakiak hartzea eta sormena sustatu, berehalako feedbacka eta elkarlana erraztu besteak beste [8]. Naiz eta *Game-Learning*-a oso erakargarria den arren ez dut uste irakasgai guztietarako erabilgarria denik. Programazioak ordea, oso harreman estua dauka bideo-jokoekin, eta ikasleak, funtsezko kontzeptuak ikasten dituzten bitartean informatikaren mundura gerturatu eta teknologien zientzietan interesa eskuratu dezakete.

Azkenik, bideo-jokoetan dudan interes pertsonala dago. Jokozalea izanik GAL hau entretenigarria izan daiteke, horrela, jolastutako jokoen atzean dauden mekanikak eta garapen prozesuak ikasi eta ulertu ditzaket. Joko bat modu profesional baten zelan egin ikasi nahi dut eta horrela, agian, joko-programatzailea izan etorkizunean.

2 Hasierako planteamendua

Atal honetan proiektuaren helburuak, bizitza zikloa, garapen prozesuko faseak eta fase bakoitzeko denbora-plangintza azalduko dira. Horretaz gain, proiektuaren garapenerako aukeratutako software eta hardware baliabideak eta hauen justifikazioa, bideragarritasun ekonomikoa eta proiektuaren garapenean suertatu daitezkeen arriskuak, arriskuen prebentzio plana eta hauei aurre egiteko kontingentzia planak garatuko dira.

2.1 Helburuak

Proiektua amaituta egongo da ondorengo helburuak betetzen direnean:

- Arazorik gabe funtzionatzen duen 2 dimentsioko plataforma joko¹ funtzionala lortzea. Jokoa oztopoz betetako gelaz osatuta egongo da eta oztopoak gainditzeko trebetasun ezberdinak (salto, makurtu, korrika egin, irristatu...) sortuko dira.
- Programazio ariketak egiteko interfaze sinplea eta ulerterraza sortzea. Horretarako, kode lerroak *Drag and Drop* sistema erabiltzea mugituko dira eta kolore eta forma ezberdinak izango dituzte. Gainera, interfazeak emaitza ezberdinak irakurtzeko funtzionalitatea izango du.
- Bideo-jokoak eta programazioak duten harreman estua aprobetxatuz programazio irakasgaiarekin lagungarriak izan daitezkeen ariketak garatzea, ariketen emaitzak jokoan eragina dutelarik.
- Jokoan aurkitzen diren ariketak programazioko oinarritzko osagaien inguruan (if/else egiturak eta baldintzak, aldagaiak, eragigaiak, funtzioak, begiztak eta for egiturak) informazioa eta azalpenak eskaintzea.
- Joko bat garatzeko beharrezkoak diren ezagutzak bereganatzea. Jokoa garatzeko erraminta zeintzuk baliabide eskaintzen dituen eta hauek erabiltzen ikasi: fisika legeak jokoan aplikatu, animazio grafoak garatu, soinuak zelan jarri...
- Bideo-jokoa ikasgela baten aurkitu daitekeen ordenagailu batek erabili ahal izatea. Funtzionatzeko hardware baliabide gutxi erabili behar du eta erresoluzio grafiko gehienetan funtzionatu.

¹ **Plataforma jokoak:** Jokoaren ekintza nagusia alde batetik bestera salto egitea denean, joko mota plataforma jokoak dela esaten da. Mario edo Sonic plataforma jokoak dira.

2.2 Aurrekariak

Joko bidezko irakaskuntzak duen potentziala jakinda, irakasgai askotarako lagungarriak diren jokoak agertu dira hemen eta hor eta programazioa horien artean dago. Atal honetan programazioa eta informatikaren alorrean dauden jokoak aztertuko dira eta bakoitzak dituen abantailak eta ahuleziak deskribatu ere.

2.2.1 CodeMonkey

Code monkey terminoa programazio ezagutza minimoa duen pertsona deskribatzeko erabiltzen da, baina 2014 urtetik aurrera, programatzen irakasteko jokoetan oinarritutako erraminta ere bada. Jokoa web bitartez erabiltzen da eta <https://www.codemonkey.com/> webgunean eskuragarri dago. 8 urtetik aurrera duten umeentzat pentsatuta dago, baina oraindik irakurtzen ez dakiten umeentzat ere mugikor aplikazioa dago.

Jokoan tximino batek galdu dituen bananak berreskuratu behar ditu. Horretarako, ariketan erakusten diren kode lerroak idatzi behar dira (kodea irudiak sakatuz ere idatzi daiteke). Ariketan kode lerroak idatzi ondoren, tximinoak idatzitako aginduak jarraitzen ditu eta bananak berreskuratzen baditu ondorengo ariketa egin dezake.

Horretaz gain, irakasgai aurreratuek ere badaude, matematika ikasteko ariketak eta jokoak sortzeko aukerak ere eskaintzen ditu. Zoritxarrez, jokoa ez dago euskaraz itzulita, baina bai erdaraz. Azkenik aipatu behar da 14 egunez doan jolastu daitekeela, eta hortik aurrera guraso edo irakasleentzako ordainketa ezberdinak daudela.

2.2.2 Code.org

2013. urtean, ikasle guztiek programazio irakasgaia ikasteko aukera izan dezaten <https://studio.code.org/> webgunea argitaratu zen. Webgunean, programatzen ikasteko jokoak daude eskuragarri. Ariketak 3 multzutuan banatzen dira adinaren arabera: 4-11 urte bitartekoak, 10-16 urte bitartekoak eta 14-18+ urte bitartekoak.

Irakatsi nahi den gaiaren arabera, joko edo ekintza bat erabiltzen da. Kode zatiak zer egiten duten aldeztu aurretik erakusten da, gero, puzzle pieza itxura duten kode lerroak erabilia ariketak ebatzi behar dira. Jokoak ez dauka funtzionamendu lineala: ez dago zertan ariketa bat ebatzi behar hurrengo egiteko, baina oso komenigarria da piezen funtzionamendua eta azalpenak mailaz maila jasotzeko.

Aurreko adibideak bezala, jokoak sortzeko aukera eskaintzen du, baina, aurrekoak ez bezala, webgune hau euskaraz erdi-itzulita dago eta guztiz doakoa da.

2.2.3 Scratch

Scratch da ziur asko hiru adibideetatik ezagunena. Webgune hau (<https://scratch.mit.edu/>) 2007. Urtean argitaratu zen eta urte horretatik gaur arte (2019-ko otsailean 3.0 bertsiora eguneratu da) ezaugarri berriak jaso ditu. Aplikazioa 8 urtetik gorako umeak erabiltzeko pentsatuta dago.

Scratch adibide guztietatik aukera gehien eskaintzen dituen da, baina baita azalpen gutxien ematen dituen. Izan ere, Scratch, animazioak, istorioak edo jokoak sortzeko, aldatzeko eta partekatzeko helburuarekin garatuta dago. Erabiltzaileak aukera denak ditu eskuragarri hasieratik, ia argibiderik jaso gabe. Kode lerroak puzzle pieza itxura dute ere, eta hauen funtzionamendua probak eginez ikasten da.

Web bitartez edo mahai gaineko aplikazioa bezala aurkitu daiteke, eta hauek euskaraz itzulita daude ia osotasunean. Biak doakoak dira.

2.2.4 Aurrekarien laburpena

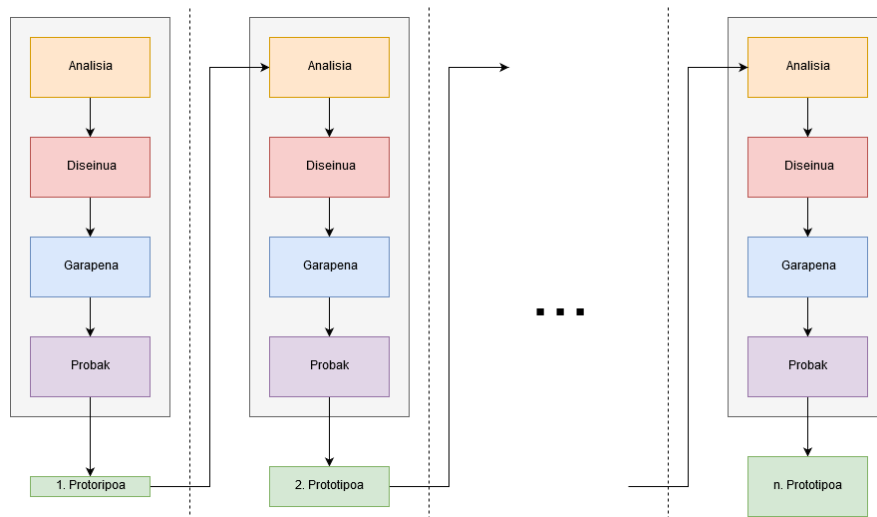
Oso antzeko ikuspegia daukate aztertutako adibideak: ariketa bat aurkezten da eta ondoan dagoen leihoan idatzitako kodea interpretatzen da. Leiho horrek joko baten antza duen arren, ariketa aginduak jarraitzen dituzten animazioak besterik ez dira. Erabiltzaileak ez dauka jokoan inolako kontrolik ariketatik kanpo.

Hieroglifoak jokoaren ikuspegitik bideratuko da eta alde ludikoari garrantzi handiagoa emango zaio. Funtsean plataforma joko arrunt bat izango da zeinetan erabiltzaileak ariketak egin beharko ditu jokoaren portaera aldatzeko eta jolasten jarraitu ahal izateko. Ondorioz, ariketak ez dira bakarrik programazio probak izango beste kasuetan gertatzen den bezala, baizik eta jokoaren esperientzia hobetuko dituzten puzzleak.

Orotara, Hieroglifoak plataforma joko bat izango da zeinek programazioa ikasteko lagungarriak diren ariketak izango dituen. Ariketak programatzen ikasteko lagungarriak izateaz gain jokoaren funtzionamendua aldatuko dute. Jokoa euskaraz egongo da eta doakoa izango da.

2.3 Bizi zikloa

Proiektuaren garapenerako eredu inkrementala erabiliko da. Eredu mota honetan lanaren osotasuna zati txikiagotan banatuko da: prototipoak. Prototipoetako bakoitzak bere analisi, diseinu, inplementazio eta probak izango ditu eta prototipo berriak aurrekoen emaitzatik hedatuko dira. Bizi zikloaren egitura [1](#). irudian ikus daiteke.



1. IRUDIA: BIZI ZIKLOA

Eredu honek abantaila ugari eskainiko ditu. Hasteko, proiektuaren fase goiztiarretan zati funtzionalak lortzea ahalbidetuko du. Horrela, lanaren helburuak betetzen ari diren jakingo da garapen prozesuaren fase goiztiarrean. Gainera, prototipo bakoitzak bere probak izango dituen akatsak bizkorrago konponduko dira. Prototipo berriak aurrekoen emaitzetatik hedatuko direnez, bigarrenak lehenengoan eragiten badu eta arazoak azaltzen badira, konponketak egiteko aukera emango du ere. Azkenik, analisia eta diseinua fase bakoitzaren barruan egongoenez aldaketak egiteko malgutasuna eskainiko du.

Proiekturako 6 prototipo pentsatu dira:

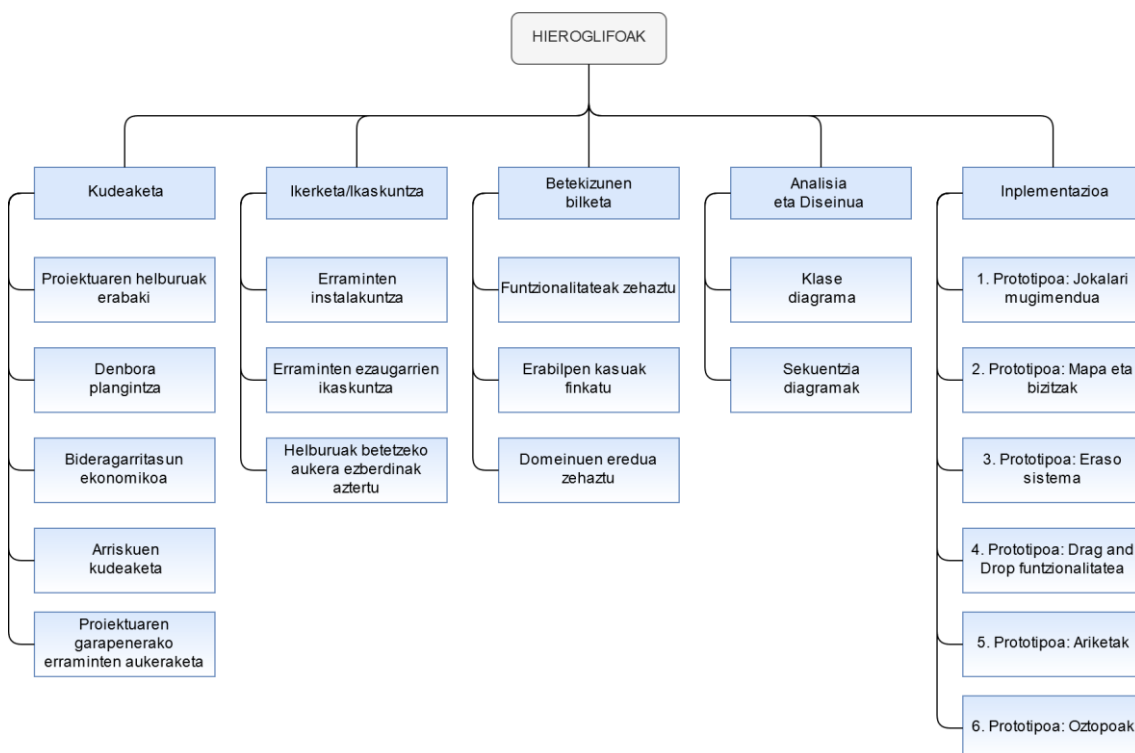
1. **Jokalariaren mugimendua.** Prototipo honetan jokalaria egin ahal dituen ekintzak garatuko dira talkak hautemateko berezko sistemarekin eta mugimendu bakoitzerako animazioak egingo dira.
2. **Mapa, galdu eta berriz hasi egoerak.** Prototipo honetan mapa irudikatze tresnak prestatuko dira, jokalaria bizitzak galtzeko egoerak sortuko dira eta partidano egindako aurreraketak ez galtzeko *checkpoint*² sistema garatuko da.
3. **Eraso sistema.** Prototipo honetan jokalaria aurka doazen etsaiak sortuko dira, etsai mota bakoitzak mugitzeko eta erasotzeko modu bakarra izango du eta jokalaria etsai aurre egiteko itemak eta eraso ekintzak sortuko dira.
4. **Drag and Drop funtzionalitatea.** Prototipo honetan ariketak egiteko interfazea garatuko da: kode lerroak mugitzeko *Drag and Drop* funtzionalitatea garatuko da, kode lerroak mota eta kolore ezberdinetan sailkatuko dira eta ariketetan emaitza ezberdinak irakurtzeko funtzionalitatea sortuko da.
5. **Ariketak.** Prototipo honetan jokalaria ekintza edo mugimendu bakoitzeko ariketak sortuko dira, trebetasunak gaitu eta ezgaitzeko funtzionalitatea implementatuko da, partida gordetzeko eta kargatzeko funtzionalitatea garatuko da eta hasiera eta *pause* menuak egingo dira bakoitza bere menu barruko funtzioekin.

² **Checkpoint:** Joko baten hil ostean jokalaria berriro agertzen den tokiari esaten zaio. Partida hasieratik berriz ez jolasteko erabiltzen da, eta hasiera eta bukaera artean hainbat jartzen dira.

6. **Oztopoak.** Prototipo honetan mapa marrartzuko da, ariketak egin behar diren tokietan oztopoak jarriko dira, ariketak osatzean oztopoa ebazteko trebetasunak gaituko dira, soinua implementatuko dira eta irudi apaingarriak jarriko dira.

2.4 Lanaren deskonposaketa

Proiektuaren planteamendutik bukaerako emaitza izan arte jarraituko diren pausua deskribatuko dira. Lanaren garapeneraren antolakuntza egokirako, atazak moduluetan sailkatu dira hauen eginkizunaren arabera. Lanaren deskonposaketa egitura 2. irudian ikus daiteke.



2. IRUDIA: LDE DIAGRAMA

- **Kudeaketa:** Modulu honetan proiektuaren helburuak definitzeko eta zereginen plangintza eta antolakuntzaz arduratzen diren atazak egongo dira.
- **Ikerketa / Ikaskuntza:** Proiektua burutzeko lan tresnak eta joko bat garatzeko urratsak ezezagunak direnez, hauen instalakuntzaren eta erabileraren inguruan informazioa bildu eta barneratu behar izango da.
- **Betekizunen bilketa:** Modulu honetan proiektuaren funtzionalitateak eta erabilera zehazteko atazak egongo dira.

- Analisia eta Diseinua:** Modulu honetan proiektuaren helburuak betetzeko atazak zelan ebatziko diren adieraziko da. Lana zati txikietan banatuko da proiektuko erronkei errazago aurre egiteko. Gainera, problemak ebazteko prozesua zein izango den ere erabakiko da.
- Implementazioa:** Modulu honetan proiektuaren helburuak pixkanaka betetzen dituzten prototipoen garapena aurkituko dira. Gainera, 1. irudian adierazi den bezala prototipo bakoitzak behar dituen analisi eta diseinu prozesuak izango ditu. Azkenik, lan egiten ari den prototipoaren zein aurreko prototipoen funtzionalitateak egokiak direla ziurtatzeko probak ere izango ditu.

2.5 Denbora-plangintza

Hemen aurreko ataleko prozesu eta azpi-prozesu bakoitzaren iraupenaren estimazioa egingo da:

1. TAULA: KUDEAKETA DENBORA-PLANGINTZA

PROZESUA	IRAUPENA
Proiektuaren helburuak erabaki	3 ordu
Denbora-plangintza	2 ordu
Bideragarritasun ekonomikoa	3 ordu
Arriskuen kudeaketa	4 ordu
Proiektuaren garapenerako erramintak aukeratu	3 ordu
KUDEAKETA	15 ordu

2. TAULA: IKERKETA DENBORA-PLANGINTZA

PROZESUA	IRAUPENA
Erraminten instalakuntza	3 ordu
Erraminten ezaugarrien ikaskuntza	55 ordu
Helburuak betetzeko metodo ezberdinak aztertu	34 ordu
IKERKETA / IKASKUNTZA	92 ordu

3. TAULA: BETEKIZUNEN BILKETA DENBORA-PLANGINTZA

PROZESUA	IRAUPENA
Funtzionalitateak zehaztu	4 ordu
Erabilpen kasuak finkatu	12 ordu
Domeinuen eredia zehaztu	2 ordu
BETEKIZUNEN BILKETA	18 ordu

4. TAULA: ANALISIA ETA DISEINUA DENBORA-PLANGINTZA

PROZESUA	IRAUPENA
Klase diagrama	16 ordu
Sekuentzia diagramak	22 ordu
ANALISIA ETA DISEINUA	38 ordu

5. TAULA: INPLEMENTAZIOA DENBORA-PLANGINTZA

PROZESUA	IRAUPENA
Jokalari mugimendua	72 ordu
Mapa eta bizitzak	68 ordu
Eraso sistema	34 ordu
<i>Drag and Drop</i> funtzionalitateak	44 ordu
Ariketak	50 ordu
Oztopoak	52 ordu
INPLEMENTAZIOA	320 ordu

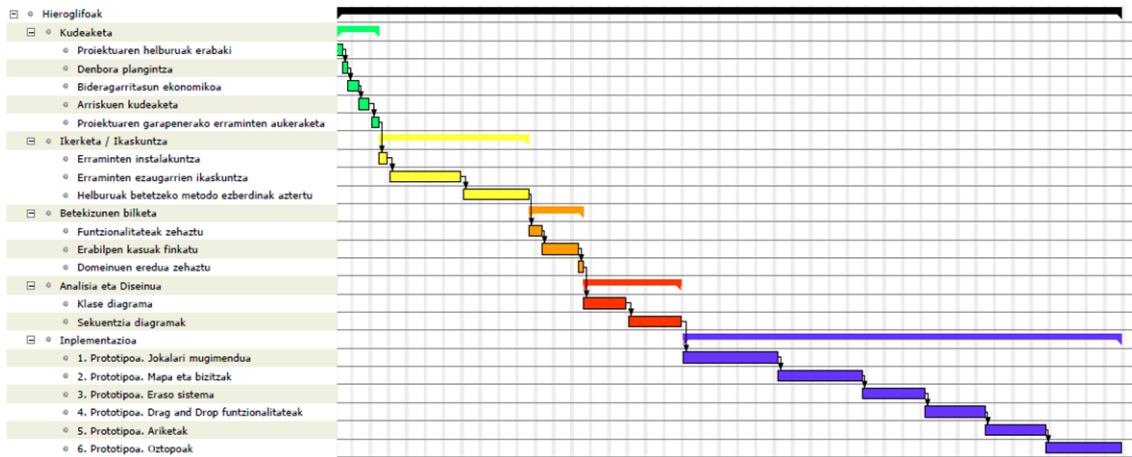
6. TAULA: DENBORA-PLANGINTZA GUZTIRA

PROZESUA	IRAUPENA
KUDEAKETA	15 ordu
IKERKETA / IKASKUNTZA	92 ordu
BETEKIZUNEN BILKETA	18 ordu
ANALISIA ETA DISEINUA	38 ordu
INPLEMENTAZIOA	320 ordu
GUZTIRA	483 ordu

Proiektua 483 ordu inguru irautea espero da. Astean 25 ordu inguru lan egitea espero da eta ondorioz proiektua 20 asteren buruan bukatzea.

2.6 Gantt diagrama

Lanaren fase guztien gutxi gorabeherako iraupena kalkulatu ondoren proiektuaren iraupena jakin daiteke. Gantt diagrama erabilita, proiektuaren faseak denboran zehar zelan hedatuko diren ikus daiteke eta proiektuaren iraupena lortu. Proiektuko zeregin batzuk aldi berean garatu daitezke, baina proiektuaren garapenean pertsona bat bakarrik lan egingo duenez, ez dago paraleloan garatzen diren atazarik. Diagramaren itxura 3. irudian ikus daiteke.



3. IRUDIA: GANTT DIAGRAMA

2.7 Erraminten aukeraketa

Proiektua burutzeko behar den software garrantzitsuena joko eta ariketak egiteko baliabideak eskaintzen duena da. Aztertu diren aukerak GameMaker Studio 2, Unity eta Unreal Engine dira.

GameMaker Studio 2-k jokoak garatzeko erraztasun handia eskaintzen du. Erraminta hau erabiltzeko ez da ia programatzen jakin behar (ia dena *drag and drop* bidez egiten da), horren ondorioz, software honek eskaintzen dituen garapen aukerak mugatuta daude.

Unity-k funtzionalitate gehiago eskaintzen ditu, beste software askorekin bateragarria da eta ospetsu samarra denez, softwarearen dokumentazioaz gain, erabiltzaileek eskaintzen duten informazio asko aurkitu daiteke Interneten. Horretaz gain, software honek denda bat dauka zeinetan erabiltzaileek Unity-rentzat erraminta gehigarriak eskaini dezakete.

Unreal Engine-k funtziorik konplexuena garatzeko aukera eskaintzen du, baina prozesu erraz bat garatzeko ere antzeko pausuak jarraitu behar dira. Software honek besteak baino funtzionalitate eta erraminta gehiago ditu, baina erabiltzaile berri batentzat, hauek erabiltzen ikasteko prozesua luzea eta zaila izan daiteke.

Hau guztia kontuan hartuta, erdiko bidea hartzea eta proiektua Unity-rekin egitea erabaki da. Ez da joko bat garatzeko baliabide gehienak eskaintzen dituen, baina jokoaren betebeharrak asetzeko nahikoak dira eta prozesu konplexuak burutzeko informazio asko dago. Ondorioz, proiektuaren helburuak betetzeko ondo baino hobeto dago.

2.8 Software baliabideak

Atal honetan proiektuaren helburuak betetzeko eta memoria idazteko zein software eta erramintak erabiliko diren eta hauetako bakoitza zertarako erabiliko den azalduko da.

- **Unity Personal:** Jokoa garatzeko irudiak, fisika aginduak, talka hautemateak, script-ak, soinuak eta animazioak kudeatzen dituen tresna da. Doako bertsioa da (urtean 100 mila dolarreko irabazia baino gutxiago duten enpresentzat) eta erabiltzaile-kontua sortu behar da. Unity-ren denda erabilita beste 3 pakete osagarri erabiliko dira:
 - **Cinemachine:** Unity-ren software osagarria, kameraren mugimendua kontrolatzeko erabiliko da.
 - **2D Pixel Perfect:** Unity-ren software osagarria, *pixel art*³ motako irudiekin lan egitean sortu litezkeen arazoak konpontzeko erabiliko da.
 - **Auto Letterbox:** Unity-ren software osagarria, pantaila erresoluzio ezberdinekin lan egiteko lagungarria izango da.
- **Visual Studio:** C# lengoian egindako script-ak idazteko erabiliko den IDE-a (*Integrated Development Environment*).
- **Piskel:** Pixel motako irudi eta animazioak sortzeko online erraminta da. Jokoa dauden irudi, interfaze eta animazio denak *pixel art* motakoak izango dira eta aplikazio honekin sortuko dira. Web-orria <https://www.piskelapp.com/> da.
- **Microsoft Word:** Memoria idazteko erabiliko den testu prozesatzailea.
- **Git eta GitHub Desktop:** Kodea eguneratuta izateko, bertsio kudeaketa egiteko eta bi gailurekin lan egiteko erabiliko den softwarea. Kode aldaketak GitHub web-orria (<https://github.com/>) igoko dira. Erabiltzaile-kontua sortu behar da.
- **TwistedWave:** Jokoa erabiliko diren audio pistak eraldatzeko erabiliko den doako eta online erraminta. Webgunea <https://twistedwave.com/online> da.
- **Draw.io:** Diagramak marrazteko erabiliko den web-orria (<https://www.draw.io/>). Online eta doakoa.
- **GanttProject:** Gantt diagrama marrazteko erabiliko den softwarea.

³ **Pixel art:** Marraztu nahi den irudiko pixelak banan-banan marraztean datza. Eraitza erresoluzio txikioa izaten da, baina ospe handia dauka joko retroekin duen antzazatik.

2.9 Hardware baliabideak

Hauek dira joko garatzeko, testak burutzeko eta memoria osatzeko erabiliko diren gailuak eta ezaugarriak:

- Asus All Series mahai-gaineko ordenagailua:
 - Intel Core i5.
 - 8GB RAM DDR3.
 - 1TB HDD.
- Acer Swift 3 ordenagailu eramangarria:
 - Intel Core i5.
 - 8GB RAM DDR4.
 - 256GB SSD.

2.10 Arriskuak

Proiektu baten gauzak espero bezala ez gauzatzea gertatu daiteke beti. Horregatik, proiektuarekin hasi baino lehen arrisku horiek zeintzuk izan daitezkeen zehaztu, hauen prebentzio plana egin eta arazorik izanez gero kontingentzia plana prestatu behar da.

2.10.1 Planifikazio ez aproposa

Deskribapena: Proiektuarekin aurrera goazela egindako plangintza egokia ez izatea gertatu daiteke. Ataza batentzat denbora epeak zehaztasunez esleitzea ez da batere erraza eta askotan kalkulaturako denbora ez da nahikoa izaten.

Prebentzio plana: Denbora plangintzarekin arazoak ez izateko, denbora epeak uste baino pixka bat luzeagoak jartzea komeni da. Horrela, arazorik badago epeak betetzea posible da oraindik.

Kontingentzia plana: Hala eta guztiz ere, denbora planifikazioa motz geratzen bada, atzerapen denborak plangintza barruan sartuko dira eta ondoren egin beharreko atazentzako epeak berriz kalkulatu dira.

2.10.2 Ezagutza faltarekin lotutako arazoak

Deskribapena: Lan egiten ari garen gaia berria denez, posiblea da beharrezkoa den informazio gutzia bilatu ez izana edo informazio dena barnerratu ez izatea. Horrela bada, hasieran egindako ikerketa labur geratuko da eta ikerketa prozesua ondorengo faseetara hedatu daiteke.

Prebentzio plana: Arazoari aurre egiteko, ikerketa prozesuari denbora asko eskainiko zaio. Proiektua osatzeko bete behar diren jardueren inguruan ahalik eta informazio gehien izatea komeni da, horrela, garapen prozesuan oztopo batekin topatzen bagara irtenbidea aurkitzea errazago izango da.

Kontingentzia plana: Hala ere, nahiz eta landuko diren gaien inguruan ikerketa sakona egin, egoera honekin topatzeko aukera dago. Horrela bada, gertatutako ezustekoaren inguruan azterketa sakona egingo da berriz ez gertatzeko eta denbora epeak gainditzen badira denbora planifikazioa moldatuko da.

2.10.3 Informazioa galera

Deskribapena: Gertatu daitekeen arazorik larriena istripu batengatik (hardwarea hondatu, hondamendi naturalak, fitxategiak ezabatzea) aurreratutako lana guztiz galtzea da.

Prebentzio plana: Hori ez gertatzeko, aldiro aurreratutako lana gordeko da eta astero (egunero garrantzizko aldaketak egiten badira) babes kopiak egingo dira. Babes kopiak egiteko git sistema erabiliko da eta aldaketak Github web-orrian gordeko dira.

Kontingentzia plana: Horrelako arazorik gertatzen bada, azken babes kopia berreskuratuko da eta lana hortik jarraituko da.

2.10.4 Arazo pertsonalak

Deskribapena: Garapen prozesuan teknikoak ez diren arazoak suertatu daitezke. Istripuak, lesioak edo beste konpromezu batzuk lanaren jarraitasuna frenatu edo guztiz gelditu dezakete.

Prebentzio plana: Arazo honentzako ezin da prebentzio planik garatu, baina, zorionez egoera honekin topatzeko probabilitatea ez da oso altua.

Kontingentzia plana: Horrelako ezuste batekin topatuz gero, egoera arruntera itzuli ondoren egin beharreko lehen gauza denbora plangintza moldatzea da.

2.11 Bideragarritasun ekonomikoa

Proiektuaren helburua ez da dirua irabaztea, hala ere, honen bideragarritasun ekonomikoa aztertuko da eta proiektuari probetxu ekonomikoa ateratzeko zenbait metodo planteatuko dira.

2.11.1 Eskulan gastuak

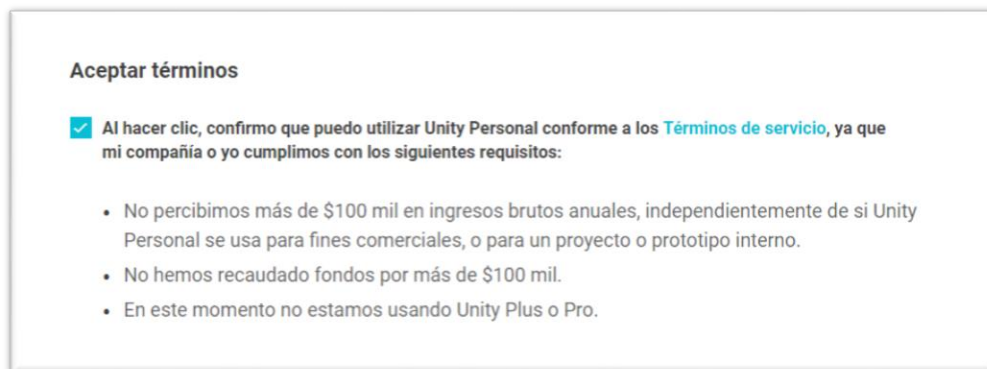
Junior programatzaile baten urteko soldata 15.860€-koa bada eta urtero 1.800 ordu lan egiten dituela kontuan hartuta [9] junior programatzaile baten orduko soldata gutxi gora behera 9€-koa dela esan daiteke. Denbora plangintza zuzena dela suposatuz (483 ordu) eskulanaren kostua estimatu daiteke:

$$\text{Eskulana: } 483 \text{ ordu} * 9\text{€/orduko} = 4.347\text{€}$$

2.11.2 Software gastuak

Proiektua garatzeko orduan softwarea doakoa izatea edo lizentziak erosi behar ez izatea kontuan hartuko da. Erabiliko diren softwaretik doakoak ez direnak ondorengoak dira:

- **Unity:** Unity Personal (doako bertsioa) erabiltzeko baldintzak Unity Plus edo Pro ez izatea, erakundeak urtero 100.000\$-ko baino gehiagoko irabaziak ez izatea (90.123€) eta 100.000\$-ko baino gehiagoko diru laguntza ez izatea dira. 4. Irudian softwarearen erabilpen baldintzak agertzen dira. Proiektu honek Unity Personal izateko baldintzak betetzen dituen ez da beste bertsiorik erosi behar izan.



4. IRUDIA: UNITY ERABILPEN BALDINTZAK

- **Microsoft office:** Ordenagailuak erostean office lizentzia aktibatuta zegoen. Ondorioz, lizentziaren kostua erositako hardwarean islatuta agertuko da.
- **Windows:** Ordenagailuak Windows sistema eragilearekin erosi ziren. Ondorioz, lizentziaren kostua erositako hardwarean islatuta agertuko da.

Gauzak horrela, software aldetik ez da gasturik egingo eta honen kostua 0€ dela esango dugu.

Softwarea: 0€

2.11.3 Hardware gastuak

Lan egiteko ezinbestekoa izango da ordenagailua erabiltzea. Gainera, etxetik kanpo lan egin ahal izateko mahai-gaineko ordenagailuaz gain ordenagailu eramangarri bat ere erabiliko da. Hardware baliabideak atalean ordenagailuen ezaugarriak ikus daitezke.

- Mahai gaineko ordenagailua: 750€ (periferikoak barne).
- Ordenagailu eramangarria: 680€

Proiektua 4 hilabete inguru iraungo duela kontuan hartuta, 7. taulan erabilitako bi ordenagailuen amortizazioa ikusten da.

7. TAULA: HARDWARE GASTUAK

Hardwarea	Prezioa	Denbora	Amortizazioa 4 hilabetetan
Mahai-gaineko ordenagailua	750€	8 urte	31,25€
Ordenagailu eramangarria	680€	5 urte	45,4€
GUZTIRA			76,65€

Hardwarearen amortizazioa: 76,65€

2.11.4 Beste gastu batzuk

Azkenik proiektua burutzeko ezinbestekoak izango diren beste gastu txiki batzuen gutxi gora beherako kostua azalduko da:

- Argi korrontea: 50€
- Sare konexioa: 100€
- Beste materiala (koadernoak, inprimaketak...): 15€
- Garraioa / Bazkariak: 30€

Beste gastuak: 195€

2.11.5 Kostu totala

Aurreko gastu guztiak batuta proiektuaren gastu totala kalkulatu daiteke. Proiektuaren gastu partzialak eta gastu totala 8. taulan ikus daitezke.

8. TAULA: GASTU TOTALA

KONTZEPTUA	KOSTUA
Eskulana	4.347€
Softwarea	0€
Hardwarea	76,65€
Bestelakoak	195€
Gastuak guztira	4.618,65€

2.11.6 Return Of Investment

Proiektua ez da dirua irabazteko asmoarekin sortu eta horregatik bukaerako emaitzan ez da probetxu ekonomikoa ateratzeko arrastorik aurkituko. Izan ere, Hieroglifoak jokoaren kodea eskuragarri egongo da Github-en edonor deskargatu, konpilatu eta erabili edo eraldatu dezan. Hala eta guztiz ere, ondorengo lerroetan garapen kostua asetzeko aplikatu daitezkeen zenbait metodo azalduko dira.

Erabiltzaileak funtzionalitate guztiak dituen jokoaren kopia erostea da lehen aukera. Horretarako, merkatua aztertu behar da zenbat kopia saldu daitezkeen jakiteko eta erabiltzaileak zenbat diru ordaintzeko prest dauden ezagutzeko.

Bigarren aukera, jokia kapituluetan edo kasu honetan ariketa multzoetan saltzea da. Demagun jokoaren hasierako zatia, oinarrizko kontzeptuak azaltzen dituen adibidez, doakoa dela. Zati horri dagozkion ariketak bukatu ondoren, jokoaren ondorengo zatiak salgai aurkitu daitezke, eta aurreko adibidearekin bezala, prezioa merkatua aztertu ondoren ezarriko litzateke. Aukera honek jokoaren bizitza luzatu dezake, eta hasieran planteatutako ariketak baino gehiago gehitu daitezke, honek duen jasoera eta arrakastaren arabera.

Beste aukera bat, joko barruan publizitatea jartzea litzateke. Esaterako, energia neurgailu bat jarri daiteke, eta ariketaren erantzuna zuzena den ikusteko energia zati bat erabiliko da. Behin energia agortuta dagoela iragarki bat ikusi behar da energia berreskuratzeko. Aukera hau ez da batere dotorea, izan ere, hezkuntzan erabiltzeko asmoarekin garatu da.

Azkenik, gobernu- eta kultura-erakundeei laguntza ekonomikoak eskatzea dago. Adibidez, erakunde hauek jokia hezkuntza-sisteman aplikatu dezakete eskainitako diru laguntzaren ordez. Aukera guztietatik hau aproposena litzateke, jokoaren xedearekin lotuta dagoelako.

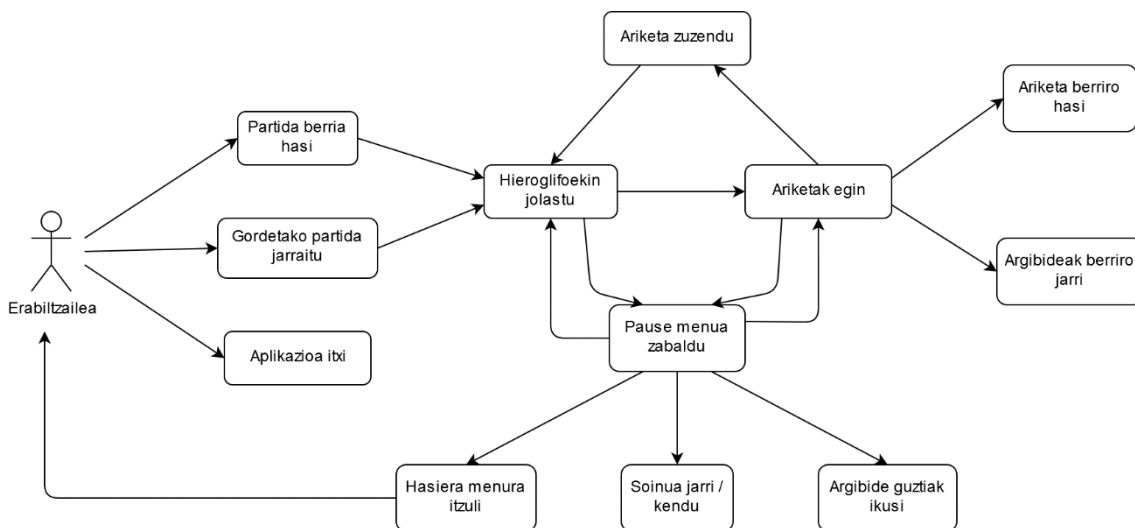
Aipatuko proposamenez gain ez da behar inbertsioaren berreskurapenerako metodo gehiago bilatzea, beraz, alor ekonomikoaren inguruan ez da analisi sakonagorik egingo.

3 Betekizunen bilketa

Atal honetan Hieroglifoak jokoak zeintzuk betekizun bete beharko dituen eztabaidatuko da. Horretarako, erabiltzaileak egin ahal izango dituen ekintzak eta egoera horietara zelan heldu erakutsiko da.

3.1 Erabilpen kasuen eredua

Erabiltzaileak jokoan egin ahal izango dituen ekintzak aipatuko dira eta bakoitza zertan datza azalduko da. Erabiltzailearen eta aplikazioaren arteko elkarrenerginak 5. irudian ikusi daitezke. Eranskinetan erabilpen kasuak eta jokoaren funtzionamendua sakonago aztertuko dira.



5. IRUDIA: ERABILPEN KASUEN DIAGRAMA

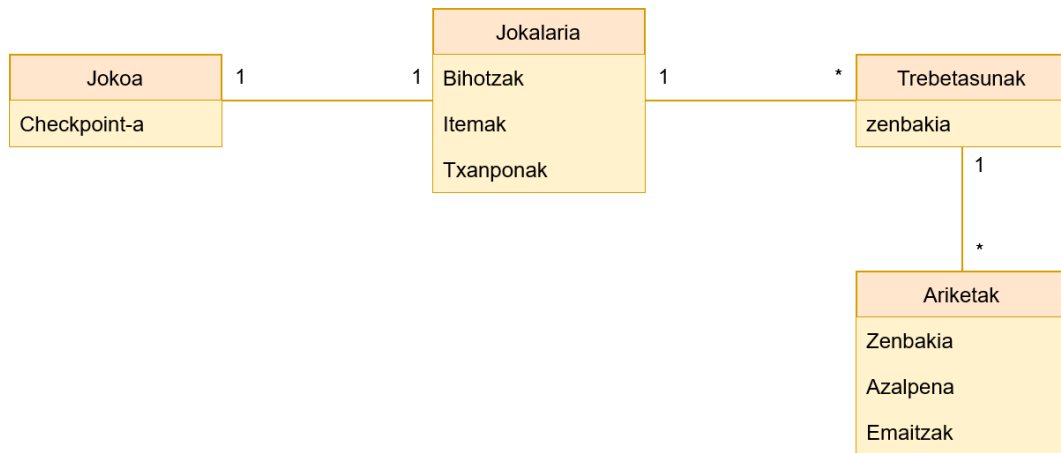
- Partida berria hasi:** Erabiltzaileari partida hasieratik jolasteko aukera eskainiko zaio. Aukera hau aplikazioaren hasiera menuan agertuko da.
- Gordetako partida jarraitu:** Erabiltzaileari aurretik hasitako partida jolasten jarraitzeko aukera eskainiko zaio. Partida ez da utzitako toki zehatzean jarraituko, jokalariak zeharkatutako azken *checkpoint*-ean baizik. Aukera hau aplikazioaren hasiera menuan agertuko da.
- Aplikazioa itxi:** Erabiltzaileari aplikazioa ixteko aukera eskainiko zaio. Aukera hau aplikazioaren hasiera menuan agertuko da.

- **Hieroglifoekin jolastu:** Erabiltzaileari jolasteko aukera emango zaio. Partida hasieratik edo, gordetako partida kargatzen bada, dagokion *checkpoint*-ean jarraituko da. Hasiera menuan jolasteko aukera sakatu ostean gertatuko da.
- **Ariketak egin:** Erabiltzaileak ariketak egiteko aukera izango du. Jokoa aurrera egin ahala ariketak agertuko dira eta agertzen diren oztopoak gainditzeko ezinbestekoa izango da hauek ebaztea.
- **Argibideak berriro jarri:** Erabiltzaileari ariketaren argibideak berriro ikusteko aukera eskainiko zaio. Aukera hau ariketak egiteko interfazeaz soilik egongo da. Argibideak ikusi ondoren ariketa agertuko da berriz.
- **Ariketa berriro hasi:** Erabiltzaileak ariketa hasieratik egiteko aukera izango du. Botoi honekin lerro kodeak hasieran zeuden posiziora itzuliko dira. Ariketa interfazeaz bakarrik agertuko da aukera hau.
- **Ariketa zuzendu:** Erabiltzaileari ariketaren emaitza zuzena den ikusteko aukera eskainiko zaio. Zuzena bada jolasten jarraituko du, bestela, ariketa egiten jarraituko du.
- **Pause menua zabaltu:** Erabiltzaileak edozein momentuan joko gelditu eta *pause* menua zabaltzeko aukera izango du. Menua zabaltzean denbora gelditu egingo da: jokalaria ezin izango da mugitu, ez du minik jasoko eta ariketak ezin izango dira kendu. Ondorioz, menua ixtean, aurretik egiten zegoen jarduera jarraituko du erabiltzaileak.
- **Argibide guztiak ikusi:** Erabiltzaileari ariketen inguruan informazioa ikusteko aukera eskainiko zaio. Izan ere, aurreko ariketetan ikusitako kontzeptuak aurrerago ere erabiliko dira. *Pause* menua erabilita, argibide guztiak edozein momentuan ikusi ahal izango dira.
- **Soinua jarri/kendu:** Erabiltzaileari jokoaren soinuak aktibatzeke edo desaktibatzeke aukera eskainiko zaio. Menuan dagoen botoia erabilita soinu efektuak eta musika ezgaituko dira. Botoia berriz sakatzen bada berriro aktibatuko dira.
- **Hasiera menura itzuli:** Erabiltzaileari hasierako menura itzultzeko aukera eskainiko zaio. *Pause* menuaren bitartez egingo da ekintza hau eta irten aurretik jokoa egindako aurrerapenak gordeko dira.

3.2 Domeinuen ereduak

Domeinuen ereduarekin aplikazioaren objektuak izango duten egitura eta atributuak ezagutu daiteke, eta beste objektuekin zelako erlazioak izango dituen ikusi. Eredu proiektuaren helburuak asetzeko betebeharretan oinarrituta dago eta sistemaren funtzionamendua erraztasunez ikusteko balio du.

Kasu honetan, 6. irudian, jokalaria, jokolariaren trebetasunak eta ariketak zelan elkarrenganduz ikusi daitezke.



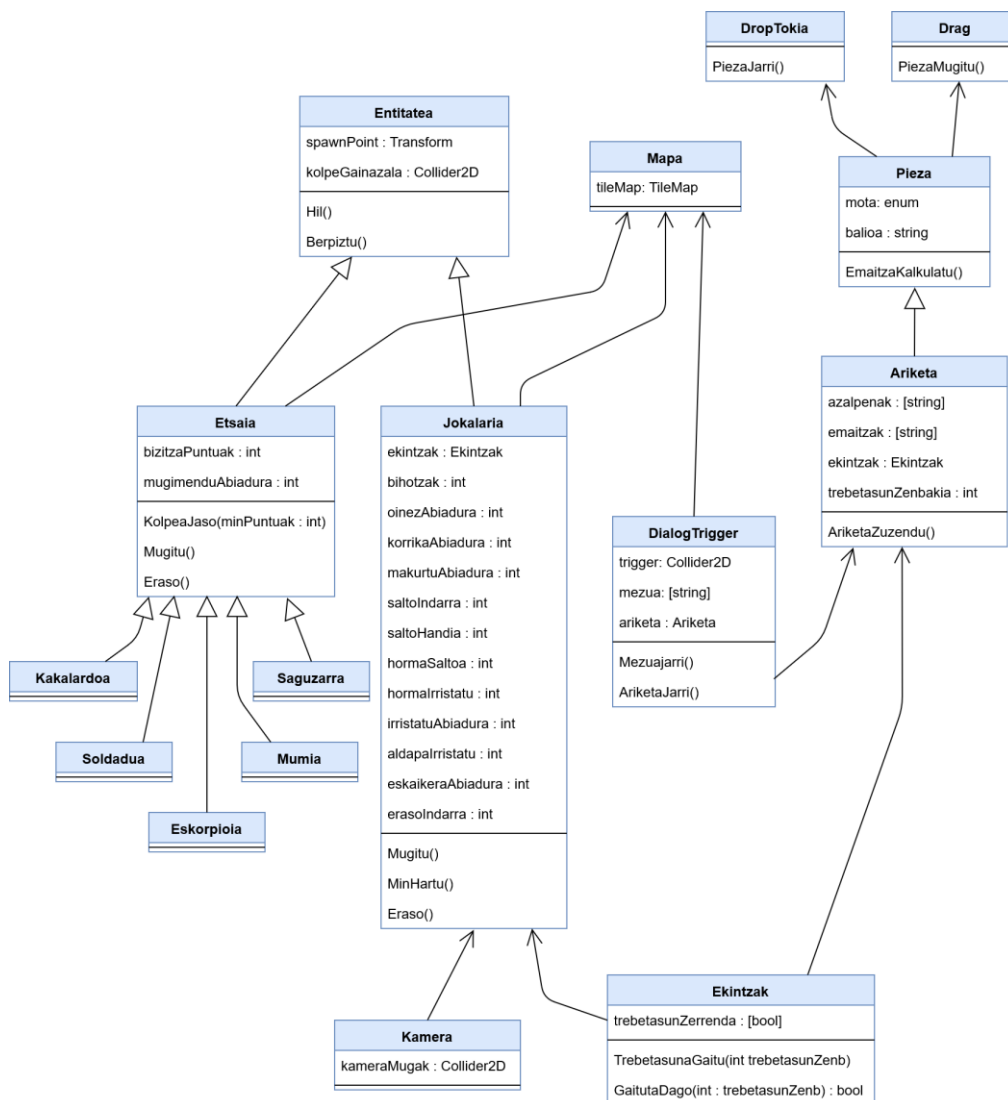
6. IRUDIA: DOMEINUEN EREDUA

- **Jokoa:** Jokalaria hiltzen bada edo aplikazioa itxi eta berriro zabaltzen bada, jokalaria agertu beharreko tokia adierazten du objektu honek. Jokalariak *checkpoint* bat zeharkatzen duenean agerpen tokia eguneratuko du.
- **Jokalaria:** Jokalariaren egoera adierazten du entitate honek: momentuko bihotz kopurua eta hartutako txanponak eta itemak.
- **Trebetasuna:** Jokalariak jokoan barrena mugitzeko izango dituen gaitasunak dira. Ariketak egin ahala abilezia berriak izango ditu eskuragarri.
- **Ariketa:** Jokoan dauden ariketak zenbaki identifikatzaile bat, azalpenak eta emaitza posibleak izango ditu. Jokalariak trebetasunak lortzeko ariketa bat edo gehiago ebatzi behar izango ditu.

4 Analisia eta diseinua

Atal honetan jokoaren barne egitura aztertuko da. Proiektua Unity-rekin garatuta dago, ondorioz, programa honen funtzionamendua dela eta, klaseen arteko erlazioak ohikoak baino konplexuagoak dira. Eszenatokiak, objektuak eta osagai guztiak dituen klase diagrama sortzea posiblea den arren, mapa konplexua eta irakurgaitza izango litzateke eta ondorioz diagrama ez litzateke erabilgarria izango.

Funtzionala den diagrama osatzeko, baliozko informazioa izango duten objektuak eta ezaugarriak erabili dira bakarrik, eta jokoaren funtzionamenduan eragin txikia izango dituztenak edo proiektuaren helburuekin erlazio gutxi izango dituztenak baztertu egin dira. Aldaketa horiek kontuan hartuta, 7. irudian klase diagramak duen egitura ikus daiteke.



7. IRUDIA: KLASA DIAGRAMA

Jokoaren gutxi gorabeherako egitura izanik, objektu bakoitzaren eginkizuna zein den azalduko da:

- **Entitatea:** Objektu honek jokoan dauden izakiak izango dituzten ezaugarriak eta ekintzak biltzen ditu. Bi atributu ditu: eraso sistemarentzako kolpe gainazala eta entitatea hil ondoren berriz agertzeko posizioa. Objektu honen metodoak *Hil()* eta *Berpiztu()* dira. Lehenak, entitateen arteko elkarreagina geldituko du. Bigarrenak, *spawnPoint* puntuan jartzen du entitatea eta aipatutako elkarreagina berriz aktibatuko du.
- **Etsaia:** Jokalariaren aurka doazen entitateak izango dira. Bizitza puntuak izango dituzte eta hauek murrizteko metodoa. Horretaz gain, mapatik mugitzeko abiadura eta metodoak izango dituzte eta jokalaria mintzeko funtzioa ere.
- **Jokalaria:** Erabiltzaileak kontrolatzen duen entitatea izango da. Mugimendu ugari egin ahal izango ditu, baina Ekintzak klaseak zeintzuk egongo diren gaituta eta zentzuk ez kudeatuko du. Mugimendu mota bakoitzak bere abiadura izango du, erasotzeko indar balio bat eta bizirik mantentzeko bihotz kopuru finkoa. Azkenik, mugitzeko, erasotzeko eta bihotzak murrizteko metodoak izango ditu.
- **Kamera:** Jokalaria jarraituko du eta jokoan zein mapa zati ikusiko den kudeatzen du. Kamera mugitu daitekeen azalera definituko da marraztu gabeko tokiak pantailan ez agertzeko.
- **Mapa:** Objektu honetan lurra eta beste irudiak kokatuko dira. *TileMap*⁴ bitartez marraztuko da. Jokoan dauden objektuak mapan jarriko dira.
- **DialogTrigger:** Ariketak jartzeko erabiliko da objektu hau eta jokalaria honen gainazala ukitzen duenean eragingo da. Ariketa jarri aurretik mezu bat agertuko da pantailan jokalaria aurrean duen oztopoa gainditzeko trebetasun berria behar duela esanez.
- **Pieza:** Ariketak egiteko objektuak izango dira. Mota ezberdinetan sailkatuko dira: ekintzak, baldintzak, aldagaiak, zenbakiak, string-ak eta abar. Motarekin piezak zein tokitan jarri daitezke jakingo da. Horretaz gain, pieza bakoitzak balio bat izango du, eta pieza batzuk bata bestearen barruan jarri ahal izango dira (mota egokia denean). *EmaitzaKalkulatu()* funtzioak sortutako pieza konbinaketaren balioa kalkulatu du.
- **Ariketa:** Pieza handi bat izango da zeinetan piezak barruan jarri ahal izango dira baina ezin izango da mugitu. Ariketak azalpenak, emaitza zuzen posible guztiak eta aktibatzen duen trebetasunaren zenbakia izango ditu. Ariketa zuzentzeko Pieza klasetik heredatuko duen *EmaitzaKalkulatu()* metodoaren erantzuna eta emaitza zerrenda konparatuko dira.
- **Drag:** Piezak mugitzeko erabiliko da.

⁴ **TileMap eta TileSet:** Jokoetan eszenatokia marrazteko teknika berezia da. Eszenatokia bata bestearen ondoan jartzen diren irudi karratuekin marrazten da (irudi ezberdin guztien multzoa *TileSet-a* da). Bukaerako emaitzan, irudiak errepikatzen direla ikusten da, baina irudietako bakoitza non hasten den eta non bukatzen den ezin da hauteman. *TileMap-a* laukietan banatutako eszenatokia da zeinetan laukietako bakoitzean irudi bat jarri daiteke.

- **Drop:** Piezak jartzeko erabiliko da.
- **Ekintzak:** Jokalariaren trebetasun zerrenda izango da. Ariketak egin ahala trebetasun gehiago aktibatuko dira. Erabiltzailearen input-a erabilia eta aktibatutako trebetasunak kontuan hartuta mugimendu bat edo beste bat burutuko da.

5 Unity

Atal honetan Unity motore grafikoaren funtzionamendua eta egitura ulertzeko azalpenak eskainiko dira.

5.1 Objektuak

Objektuak joko bat sortzeko ezinbesteko elementuak dira, programazioan erabiltzen diren objektuen antzera funtzionatzen dute. Jokalaria, etsaiak, itemak, beste objektuak pantailaratzeko kamera edo soinuak jartzen dituen elementua objektuak dira eta guztien multzoak jokoa osatzen du. Objektuak 3 ezaugarri berezi dituzte:

- Osagai eta aldagaiak dituzte funtzio ezberdinak burutzeko.
- Funtzio horiek jokoaren fotograma bakoitzean aktibatu daitezke.
- Objektu batek beste objektu batzuk izan ditzake seme bezala.

Azken ezaugarria oso erabilgarria da. Ekintza ugari egiten dituen objektu bati seme bat jarri ahal zaio eginkizun bakoitzeko. Horrela, objektu gurasoaren lana seme diren objektuetan banakatu egiten da lana erraztuz. Horretaz gain, gurasoak seme guztiak eta seme bakoitzaren osagai eta aldagaiak eskuragarri dituenez, funtzionalitate konplexuak burutu daitezke.

5.2 Eszenatokia

Eszenatokia 3 dimentsioko mundu birtual baten errepresentazioa da. Objektuak eszenatokian jartzen dira eta kamera osagaia erabilia eszenatokia pantailaraten da. Eszenatokia 3 dimentsioko esparrua denez, agertzen diren objektu guztiak ezinbestekoak diren hiru osagai izan behar dituzte: espazioan, x, y eta z ardatzetan, okupatzen duten **posizioa**, **angelua** eta **eskala**.

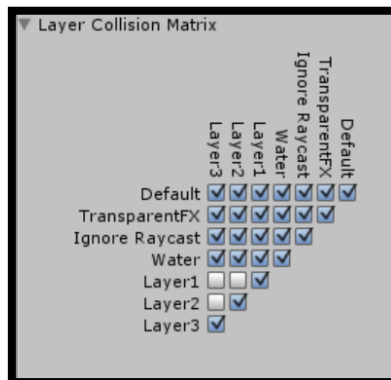
Objektu guztiak ez dute zertan eszenatokian egon behar. Objektu batzuk instantziatu eta suntsitu egiten dira jokoaren funtzionamenduagatik edo jokalariaren eraginagatik. Adibidez, Space Invaders bezalako joko baten, jokalariaren eta etsaien jaurtigaiak ez daude momentu oro pantailan, jokalaria edo estralurtar baten instantziaren ondorioz sortu eta suntsitzen dira.

5.3 Collider eta Rigidbody

Collider eta *Rigidbody* motore grafikoaren osagirik garrantzitsuenak dira. Izan ere, osagai hauen aldagaien arabera fisika legeak eragin ezberdinak izango dute.

- Collider*. Objektuaren forma definitzen duen osagaia da. Osagai hau ez dauka objektuaren irudiarekin (*SpriteRenderer* osagaia) zerikusirik, baina irudiaren antzeko forma badauka talkak benetakoak direlako irudipena sortuko da. Bi objektuen *Collider* osagaiak talka egiten badute, kode bitartez bi objektuak eta hauen osagaiak lortu daitezke. Hortaz, talka bat gertatzean nahi den funtzionalitatea garatu daiteke. Azkenik, *Collider*-ak *trigger* bezala ere erabili daitezke. *Trigger*-ak gertatzen diren talkak hautematen ditu, baina ez ditu fisika-legeak jarraituko. Beraz, *trigger*-ak kode bitartez funtzionalitateak garatzeko erabiltzen dira. Adibidez gela baten sartzean argia kentzeko eta beldur soinu bat jartzeko erabili daitezke edo jokalaria jaso ahal dituen itemak ere *trigger*-ak dira.
- Rigidbody*. Objektuak duen *Collider* osagaia erabilia fisika legeak simulatzeko erabiltzen da. Talkak eta ondorioz gertatzen diren mugimenduak motore grafikok kalkulatu eta aplikatzen ditu. Adibidez, pilota bat aldapa baten jartzen bada (pilotak *Collider* eta *Rigidbody* osagaiak behar ditu eta aldapak *Collider* osagaia), pilota aldapa behera jauziko da, erabiltzailearen eraginik gabe eta kode bitartez portaera erabaki gabe. *Rigidbody* osagaiak duen eragina murriztu daiteke eta kode bitartez honek duen portaera aldatu daiteke.

Objektu denak ez dute beste objektu guztiarekin zertan talka egin behar. Adibidez, Mario jokoan etsaiak ez dituzte perretxikoak hartzen. Nahi diren talkak ezgaitzeko, objektuei geruzak esleitzen zaizkie eta kolpe matrizean geruzen arteko portaera erabakitzen da. Matrizean desaktibatzen diren hutsuneak bi geruzen arteko talkak ezgaitzen ditu. 8. Irudian geruza matrize baten adibidea ikus daiteke. Lehen geruzak ez du bigarren eta hirugarren geruzarekin talka egiten eta bigarren geruzak ez du hirugarrenarekin talka egiten.



	Default	TransparentFX	Ignore Raycast	Water	Layer1	Layer2	Layer3
Default	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
TransparentFX	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Ignore Raycast	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Water	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Layer1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Layer2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Layer3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

8. IRUDIA: TALKA MATRIZEAREN ADIBIDEA

5.4 Script

Script bat objektu bati lotutako osagaia da eta kode bitartez objektuaren funtzionalitatea aldatu daiteke. Scriptak objektuak dituen osagai guztiak ditu eskuragarri *GetComponent<OsagaiarenIzena>()* funtzioa erabilia. Horrela, osagaien ezaugarriak eta balioak aldatzeko eta hedatzeko ahalmena dauka. Adibidez, script baten bitartez *RigidBody* osagaiaren masa aldagaia aldatu daiteke grabitate gutxiko funtzionalitatea sortzeko. Horretaz gain, objektu batek, script bitartez, bere semeak eta hauen osagai guztiak eskuragarri ditu funtzioak garatzeko. Eta baita alderantziz ere, seme batek gurasoaren osagaiak erabili ditzake.

Kodearen exekuzio ordenaren inguruan, Unity-n hiru funtzio garrantzitsu daude:

- *Start()*. Jokoa hasi eta berehala exekutatzen da. Funtzio honetan aldagaien hasieraketa egiten da eta erabiliko diren osagaiak esleitzen dira.
- *Awake()*. Jokoa hasi aurretik (*Start()* funtzioaren aurretik) exekutatzen da. Adibidez, *Singleton* patroia funtzio honetan egiten da, horrela, beste klase batek *Start()* funtzioan instantzia behar badu eskuragarri egongo da.
- *Update()*. Aurreko bi funtzioen ondoren exekutatzen da eta etengabe egiten du. Funtzio honetan etengabe exekutatu beharreko funtzionalitateak egon behar dute. Adibidez, erabiltzailearen inputa eta jokalaria mugimendua funtzio honen barruan egon behar dute.

6 Garapena eta ebaluazioa

Proiektua osatzeko jarraitutako urratsak azalduko dira ondorengo lerroetan. Horretarako, prototipoetako bakoitzean jarraitutako prozedurei buruz eta funtzionamendua egokia dela ziurtatzeko egindako probei buruz xehetasuna emango da.

6.1 Jokalari mugimendua

Prototipo honetan jokoaren oinarriko funtzionalitatea, jokalariaren mugimendua, burutu da. Jokalariaren mugimendua proiektuaren zati ludikoaren funtsa da, ondorioz, ahalik eta zehatzena eta intuitiboena izatea saiatu da.

Unity-k fisika legeak interpretatzeko eta jokalaria mugitzeko erraminta (*RigidBody* osagaia) eskaintzen du. Honen funtzionamendua egokia den arren, ez da nahi beste zehatza eta espero ez diren portaerak gertatzen dira, batez ere aldapekin lan egitean.

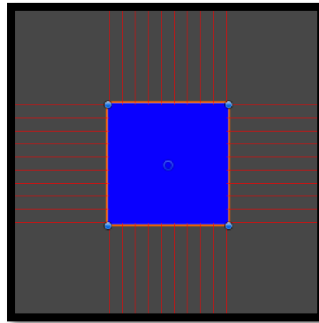
Jokalariaren mugimendu egokirako 3 gauza kontuan izan behar dira: lur-eremua edo maparekin gertatzen diren talkak, erabiltzailearen input-a eta aurreko bien konbinaketak eragindako ekintzak edo jokalariaren mugimendua. Horretaz gain, prototipo honetan jokalariaren animazioak eta kamera prestatu dira.

6.1.1 Talkak hautematea

Lehen esan bezala, ez da Unity-ren *RigidBody* tresna erabiliko. Ondorioz, egin beharreko lehenengo atazak jokalaria eszenatokiarekin izango dituen kolpeak ohartzeko modua garatzea izango da.

Talkak hautemateko modu zehatzena jokalariaren lau aldeetatik izpiak jaurtitzea da. Izpiak horizontalean eta bertikalean jaurtiko dira jokalaria mugitzen ari den noranzkoan. Izpiak oztopo bat aurkitzen duenean oztopora heldu arte dagoen distantzia neurtuko du. Distantzia hori oso txikia denean jokalariaren abiadura horizontala edo bertikala geldituko da.

Izpiak *Collider* osagaiak definitutako mugetatik jaurtiko dira. *Collider*-aren geruzaren tamaina ez da definitiboa, ondorioz, jaurti nahi diren izpi kopurua zehaztu beharrean izpien arteko distantzia maximoa definituko da eta izpi kopurua kode bitartez kalkulatu da. 9. Irudian, jokalaritik (jokalaria momentuz karratu urdin bat besterik ez da) izpiak ateratzen direla eta izpien arteko tartea berdina dela alde guztietan ikus daiteke. Jokalaria geldi dagoenez alde guztietara igorri dira izpiak.



9. IRUDIA: IZPIAK IGORTZEN DITUEN LAUKIA

Izpiak *Collider*-aren alde batetik bestera igortzen direnez, erpinen posizioa beti ezaguna izatea komeni da. Gainera, jokalaria tokiz mugituko denez, erpinen posizio berriak kalkulatu behar dira. Erpinen posizioa kalkulatzeko kodea 10. irudian ikusten da eta *Update()* metodoaren barruan (etengabe exekutatzen den metodoan) jarriko da.

```

public void IzpiJatorriaEguneratu()
{
    Bounds mugak = boxCollider2d.bounds;

    izpiJatorria.bottomLeft = new Vector2(mugak.min.x, mugak.min.y);
    izpiJatorria.bottomRight = new Vector2(mugak.max.x, mugak.min.y);
    izpiJatorria.topLeft = new Vector2(mugak.min.x, mugak.max.y);
    izpiJatorria.topRight = new Vector2(mugak.max.x, mugak.max.y);
}
  
```

10. IRUDIA: IZPIEN JATORRIA KALKULATZEKO FUNTZIOA

Behin izpiak ditugula talka eta egoera bakoitzaren informazioa gordetzeko datu egitura sortuko da. Datu hauek abiadurak eraldatzeko informazioa eskaintzeaz gain animazio zuzena jartzeko ere balio du. 11. Irudian gordeko den informazioa ikus daiteke. Kolpeen balioak gezurra izango dira (0 aldapa angeluaren kasuan eta (0,0) aldaparen normalaren kasuan) eta jokalaria eszenatokitik mugitzean balio hauek aldatuko ditu.

```

public struct KolpeInfo
{
    public bool ganean, azpian;
    public bool eskuma, ezkerra;

    public bool aldapaIgotzen;
    public bool aldapaJaisten;
    public bool aldapaIrristatu;
    public bool paretaitatsi;

    public float aldapaAngelu;
    public Vector2 normala;
}
  
```

11. IRUDIA: KOLPE INFORMAZIOA JAKITEKO DATU EGITURA

6.1.2 Erabiltzailearen input-a

Erabiltzaileak jokalaria eszenatokitik mugitu dezan teklatutik aginduak jaso behar dira. Erabiltzailearen input-ak jokalaria ekintza bat egiteko agindua emango du (mugitu, korrika, salto, makurtu...). Agindu horrek jokalaria abiadura bat esleituko dio.

Ekintza burutzeko baldintza batzuk bete behar dira. Adibidez, salto egiteko jokalaria derrigorrez lurrian egon behar du. Ekintza bakoitzerako baldintzak egokiak diren jakiteko aurretik egindako kolpe informazioa begiratu da. Baldintzak egokiak badira, jokalaria abiadura bat (horizontala x ardatzean eta bertikala y ardatzean) esleituko zaio. 12. Irudian, jokalaria gainean eragiten duen grabitate indarra, erabiltzailearen mugimendua eta salto aginduak zelan kudeatzen diren ikus daiteke.

```
if (kudeatzailea.kolpeak.gainean || kudeatzailea.kolpeak.azpian)
{
    abiadura.y = 0;
}
else
{
    abiadura.y += Vector2.down * grabitate;
}

aginduHorizontala = Input.GetAxisRaw("Horizontal");
if (kudeatzailea.kolpeak.ezkerra || kudeatzailea.kolpeak.eskuma)
{
    abiadura.x = 0;
}
else
{
    abiadura.x = aginduHorizontala * oinezAbiadura;
}

if (kudeatzailea.kolpeak.azpian && Input.GetButton("Jump")){
    abiadura.y = saltoIndarra;
}

kudeatzailea.Mugitu(abiadura * Time.deltaTime);
```

12. IRUDIA: ERABILTZAILEAREN INPUT-A JASOTZEKO METODOA

6.1.3 Jokalaria mugitu

Hasieran aipatu bezala, izpiak soilik jokalaria mugitzen ari den noranzkoan jaurtiko dira. Orain, jokalaria abiadura bat esleituta dauka. Abiadura horrekin eta eszenatokiaren aurka gertatzen diren talkak kontuan hartuta, abiadura eraldatuko da mugimendua ahalik eta zehatzen izateko.

Jokalaria mugimendu abiadura erabilia, izpiak jokalaria mugimenduaren noranzkoan jaurtiko dira bakarrik. Gainera, jokalaria mugitzen denez 11. irudian agertzen den talka informazioaren datu egitura eraldatu dezakegu eta jasotako abiadura eraldatu. Unity-k eskaintzen duen egiturak duen arazo nagusia aldapak igo eta jaisterako orduan duen portaera da.

Aldapa igotzen saiatzean oztopoa dagoela pentsatzen du eta jokalaria gelditu egiten da edo oso motel mugitzen da. Garatutako sistema honekin, izpiak erabiltzen, aldaparen malda jakin daiteke. Aldaparen angeluarekin eta erabiltzailearen aginduekin jasotako abiadurarekin, trigonometria formulak erabili daitezke aldapa normal igotzeko (abiadura horizontala abiadura bertikal eta horizontal bihurtuz).

Aldapa behera mugitzen bagara, jokalariaren mugimendua horizontala denez, aldapara itsatsita egon beharrean jokalaria airean egongo da, eta airean egotea ekintza asko egin ahal ez izatea esan nahi du. Grabitate indarra beti jokalarian eragiten duenez, izpiak beherantz igorriko dira eta aldapa jaitsi nahi dela hauteman daiteke. Aldaparen angelua kalkulatu ondoren aldapa igotzeko erabili den prozedura berdina jarraituko da.

13. Irudian prozedura zelan jarraitzen den ikus daiteke: Funtzioak abiadura bat jasotzen du, noranzkoa erabiltzen du izpia eskuman edo ezkerrean igortzeko. Izpiak talka egiten badu, aldaparen angelua kalkulatu da, eta jokalaria aldapa behera mugitzen ari dela konprobatu ondoren abiadura moldatu egiten da eta kolpeak hautemateko datu egitura aldatu egiten da.

```

// Sebastian Lague youtuben egindako tutorial sortan oinarrituta
void AldapaJaitsi(ref Vector2 abiadura)
{
    // abiaduraren ikurra erabilia jokalariaren noranzkoa jakin daiteke
    float xNoranzkoa = Mathf.Sign(abiadura.x);
    Vector2 izpia = (xNoranzkoa == -1) ? izpiJatorria.bottomRight : izpiJatorria.bottomLeft;
    // izpia igortzen da beherantz
    RaycastHit2D kolpea = Physics2D.Raycast(izpia, Vector2.down, Mathf.Infinity);
    if (kolpea)
    {
        float aldapaAngelua = Vector2.Angle(kolpea.normal, Vector2.up);
        // aldapa behera goazela ziurtatu aldaparen normala eta abiaduraren noranzkoa konparatuz
        if (Mathf.Sign(kolpea.normal.x) == xNoranzkoa)
        {
            float distantzia = Mathf.Abs(abiadura.x);
            // abiadura horizontala bertikalean eta horizontalean banatzen da aldapan
            abiadura.x = Mathf.Cos(aldapaAngelua * Mathf.Deg2Rad) * distantzia * Mathf.Sign(abiadura.x);
            abiadura.y -= Mathf.Sin(aldapaAngelua * Mathf.Deg2Rad) * distantzia;

            kolpeak.azpian = true;
            kolpeak.aldapaJaisten = true;
            kolpeak.aldapaAngelu = aldapaAngelua;
            kolpeak.normala = kolpea.normal;
        }
    }
}

```

13. IRUDIA: JOKALARIAK ALDAPAK JAISTEKO FUNTZIOA [10]

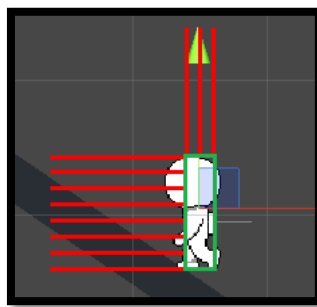
6.1.4 Animazioak

Jokalariaren mugimendu ekintzak sortu dira, baina momentuz jokalaria lauki urdin bat besterik ez da, eta ekintzak ez dira guztiz bereizgarriak. Jokalariak egoera edo mugimendu jakin bat egiten dagoela adierazteko animazioak marraztu dira. Jokalariak egin dezakeen ekintza bakoitzeko animazio bat sortu da.

Animazio kudeatzaileak animazio batetik bestera noiz aldatu behar duen jakiteko parametro batzuk sortu behar dira. Gero, parametroekin animazioak aldatzeko arauak sortu dira animazio grafo bat sortuz. Animazio kudeatzailean parametroak sortzeko 11. irudiko datu

egitura erabili da beste batzuen artean. Adibidez, datu egiturako ‘azpian’ aldagaia gezurra denean, jokalariaren salto animazioa jartzeko araua jarri da. Sortutako grafoa jokalariaren mugimenduen kateaketa kontuan hartuta bete behar da. Hau da, jokalariak bi mugimendu bata besteren segidan egin ahal baditu, grafoan ere bi mugimenduen animazioak lotuta egon behar dute, bestela, bigarren mugimenduaren animazioa ez da ikusiko.

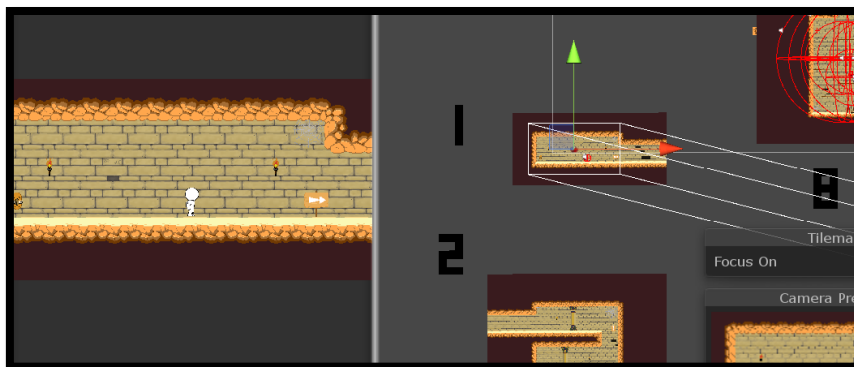
Azkenik, animazioak marraztu ondoren jokalariaren tamaina zelakoa den dakigunez, *Collider* aldagaiaren tamaina zehaztu da. Prototipoan egindako aldaketak 14. irudian ikus daitezke. Izpiak jokalariaren talka geruzatik ateratzen dira. Talkak hautemateko jaurtitzen diren izpiak jokalariaren noranzkoa jarraitzen dute: gora eta ezkerredera. Jokalaria eszenatokian dauden objektuen gainetik mugitzen da, oinez mugitzeko animazioa dauka eta *Collider*-a, lauki berdea, jokalariaren tamaina berdina du gutxi gora behera.



14. IRUDIA: ANIMATUTAKO JOKALARIA ALDAPA IGOTZEN

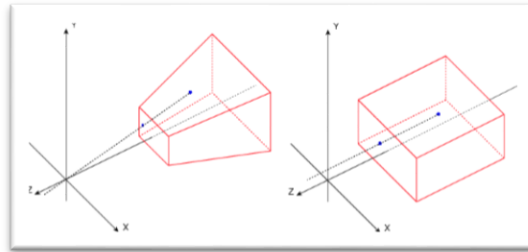
6.1.5 Kamera

Kamera osagaia eszenatokian dauden objektuak pantailan agertzeko erabiltzen da. Eszenatokia birtualki espazio infinitua denez, ezin da dena erakutsi. Gainera, gero eta espazio gehiago erakutsi orduan eta txikiagoak ikusiko dira bertan dauden objektuak. Kamera linterna bat bezala funtzionatzen du, pantailan soilik ‘argiak’ islatzen duen eszenatokiko zatia ikusten da. 15. Irudiko pantaila leihoan, ezkerreko leihoan, soilik kamerak adierazten duen zatia, laukizuzen zuri batekin marraztutako zatia, agertzen dela ikus daiteke.



15. IRUDIA: KAMERAREN ERABILERA ADIBIDEA

Unity-k bi kamera ezberdin eskaintzen ditu: perspektiba kamera eta kamera ortografikoa. Jokoa 2 dimentsiotako plataforma jokoa denez eta ez denez sakontasuna erabiliko, kamera ortografikoa erabili da. 16. Irudian perspektiba kameraren (ezkerrean) eta kamera ortografikoaren (eskuman) desberdintasuna ikus daiteke.

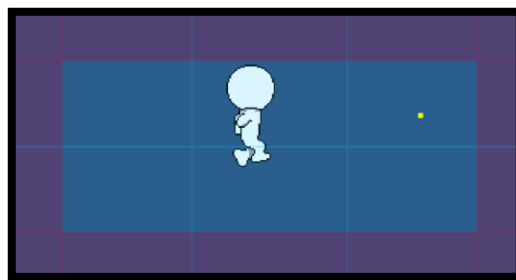


16. IRUDIA: PERSPEKTIBA KAMERA ETA KAMERA ORTOGRAFIKOA

Kamerak eszenatokiko objektu bat jarraitzeko funtzioa dauka, horrela, objektu horren inguruan dauden gauzak pantailaratzeko gai da. Jokoa garatzeko nahikoa da funtzionalitate horrekin, baina, Cinemachine osagaiak ezaugarri oso baliagarriak eskaintzen ditu. Softwarea doakoa denez eta kamera arrunta bezala funtzionatzen duenez jokoan inplementatu da.

Cinemachine-ren ezaugarri nagusia kameraren mugimendua pantailan leuna eta atsegina egitea da. Jarraitzen ari den objektua, kasu honetan jokalaria, nora mugitzen den aurreikusi dezake. Horrela, pantailaren erdian jokalaria egon beharrean, jokalaria mugitu ostean egongo den tokia ikusiko da. Horretaz gain, kameraren mugimendua kudeatzeko pantaila bi zatitan banatzen da: barruko zatia eta kanpoko. Jokalaria barruko zatian agertuko da beti, ondorioz, abiadura aldatzen duenean eta kanpoko mugara gerturatzen denean kamera ere bizkorrago mugituko da.

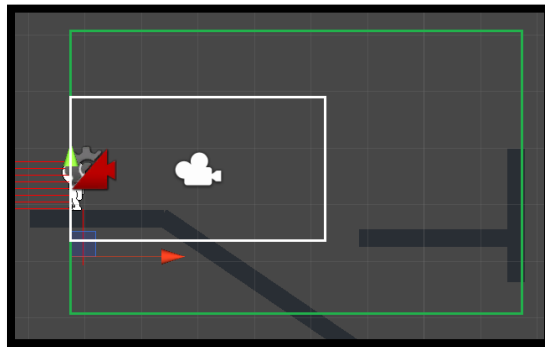
17. Irudian jokalaria salto egin duela ikusten da. Kamerak jokalaria puntu horira doala kalkulatzen du, ondorioz, jokalaria ez dago erdian, baizik eta ezkerrera baztertuta eta beti bezala barruko eremu urdinaren barnean. Horretaz gain, bi lerro bertikal ikus daitezke. Puntu horia, jokalaria mugituko den tokia, bi lerroen artean dagoen bitartean kamera ez da mugituko. Cinemacine osagaia erabiltzeko eremu urdin eta gorriaren tamainak eta lerro urdinen posizioak erabaki behar dira.



17. IRUDIA: KAMERAREN MUGIMENDUA KUDEATZEKO ESPARRUAK

Cinemachine-k eskaintzen duen beste ezaugarri bat kamerari mugak jartzea da. Funtzionalitate hau marraztu gabeko eszenatokia ez erakusteko, jokalaria maparen mugara heldu dela adierazteko edo eszenatoki zati bat izkutatzeke erabiltzen da. Adibidez, Super Mario bezalako joko baten, protagonista zulo baten jausten denean kamera ez doa zuloan behera, goian gelditzen da. Naiz eta jokoak oraindik mapa marraztuta ez duen, kamera mugak aurrerago erabiltzeko prestatu dira.

18. Irudian kameraren mugak ikusten dira. Kamera, kolore zuriz adierazita, kolore berdez marraztutako muga ezin duenez zeharkatu, ezin du jokalaria jarraitu. Ondorioz, jokalaria pantailatik kanpo irteten da.



18. IRUDIA: KAMERA MUGITU DAITEKEEN MUGA

6.1.6 Probak

Atal honetan jokariaren mugimendua, animazioak eta kameraren funtzionamendua zuzenak direla ziurtatzeko probak burutuko dira.

6.1.6.1 Talkak hautematea

Kolpeak erregistratzen dituen datu egiturak talkak eguneratzen dituela konprobatzeko jokariaren mugimendua simulatu da. Probak errazteko, izpiak noranzko bakarrean jaurti dira. Proben emaitzak 9. eta 10. tauletan ikus daitezke.

9. TAULA: TALKAK HAUTEMATEKO PROBEN EMAITZAK

PROBA	ESPERO DEN EMAITZA	LORTUTAKO EMAITZA
Eskumara mugitu	Eskuma balioa egia bihurtzen da hormara nahikoa gerturatu ondoren eta gezurra bihurtzen da noranzkoa ezkerre aldatzean.	OK
Ezkerre mugitu	Ezkerre balioa egia bihurtzen da hormara nahikoa gerturatu ondoren eta gezurra bihurtzen da noranzkoa eskumara aldatzean.	OK
Gora mugitu	Gainean balioa egia bihurtzen da sabaiaren aurka talka egitean eta gezurra bihurtzen da noranzkoa behera aldatzean.	OK
Behera mugitu	Azpian balioa egia bihurtzen da zoruaren aurka talka egitean eta gezurra bihurtzen da noranzkoa gora aldatzean.	OK
Aldapa igo horizontalean mugituz	Maldara gerturatzean aldapa igo balioa egia bihurtzen da eta eskuma/ezkerre balioa gezurra da. ⁵	ERROR1
Aldapa jaitsi horizontalean mugituz	Maldara gerturatzean aldapa jaitsi balioa egia bihurtzen da.	OK
Pareta itsatsi gorantz mugituz	Pareta itsatsi balioa gezurra da hormatik gertu zein urrun mugitu.	OK
Pareta itsatsi beherantz mugituz	Pareta itsatsi balioa egia da hormatik gertu mugitzean eta gezurra hormatik urrun mugitzean.	OK
Aldapa angelua horizontalean mugituz	Aldaparen angeluaren balioa gordetzen da. Aldapa erdian angelua aldatzen bada balioa ere eguneratu egiten da. ⁶	OK
Aldapa angelua beherantz mugituz	Aldaparen angeluaren balioa gordetzen da.	ERROR2

- ERROR1: Jokalaria aldaparen aurka talka egiten du eta aldapa igo balioa egia bihurtzen da, baina talka horizontala gertatzen da ere.
 Arazoa ekiditeko, talka horizontalak bakarrik aldapa igotzen ez dagoenean konprobatuko da, eta jokalaria aldapa igotzen egongo da soilik lehen izpia, lurraren parean dagoena, aldapa dagoela hautematen badu.
- ERROR2: Maldaren angelua gorde egiten da, baina, 'V' itxura duen bi malden artean jaustean zoratu egiten da.
 Arazoa ebazteko, ezkerreko eta eskumako erpinak jaurtitzeko dituzten izpiekin XOR eragigaitza erabiliko da, bi angelu topatzen badira ez da angelurik gordeko.

Akatsak konpondu ondoren probak errepikatu dira. 10. Taulan errepikatutako proben emaitzak ikusten dira.

⁵ Aldapa irrastatu aldagaia egia izateko erabilizailearen *input*-a behar da. Momentuz aldapa igo eta aldapa jaitsi aldagaia funtzionatzea nahikoa da.

⁶ Aldaparen normala angeluarekin lortu daiteke, hortaz, ez dira probak errepikatuko.

10. TAULA: TALKAK HAUTEMATEKO ERROREA EMAN DUTEN PROBEN ERREPIKAPENA

PROBA	ESPERO DEN EMAITZA	LORTUTAKO EMAITZA
Aldapa igo horizontalean mugituz	Maldara gerturatzean aldapa igo balioa egia bihurtzen da eta eskuma/ezkerra balioa gezurra da.	OK
Aldapa angelua beherantz mugituz	Aldaparen angeluaren balioa gordetzen da.	OK

Proba hauek talkak dagokion bezala hautematen direla konprobatzeko dira. Aurrerago, jokalariaren mugimenduaren beharrianak asetzeko talken egoerak moldatuko dira. Adibidez, jokalaria aldapa igotzen edo jaisten badago, naiz eta izpiak lurra ukitzen ez egon 'azpian' atributuaren balioa egia dela jarriko da salto mugimendua ahalbidetzeko.

6.1.6.2 Erabiltzailearen input-a eta jokalariaren mugimendua

Erabiltzailearen aginduak eta jokalariaren mugimendua batera garatu dira, trebetasun bakoitzeko aginduak eta aginduen erantzuna inplementatuz. Gainera, garatutako ekintza berriak aurreko ekintzetan eragina dute. Ondorioz, probak egitean, aurreko probak ere errepikatu dira, trebetasun berriak aurretik egindako trebetasunak zapaltzen ez dutela konprobatzeko eta ekintza guztien funtzionamendua egokia dela ziurtatzeko. Proben emaitzak 11. 12. eta 13. tauletan ikus daitezke.

11. TAULA: JOKALARIAREN MUGIMENDU EGOKIRAKO PROBEN EMAITZAK

PROBA	ESPERO DEN EMAITZA	LORTUTAKO EMAITZA
Oinez mugitu	Eskuma eta ezkerra geziak zapaltzean jokalaria noranzko egokian mugitzen da eta hormarekin talka egitean gelditzen da.	OK
Salto egin	Salto botoia sakatzean jokalaria gora mugitzen da. Sabaiaren aurka talka egiean abiadura 0 bihurtzen da eta grabitateak jokalaria behera bidaltzen du berriro. Zoruaren aurka talka egitean jokalaria gelditzen da.	OK
Aldapa igo oztoporik gabe	Jokalariak aldapa igotzen du abiadura konstante mantenduz.	OK
Aldapa igo oztopoa gainean	Jokalaria aldapa igotzen hasten da eta oztopoa aurkitzean gelditu egiten da.	ERROR1
Aldapa igo oztopoa eskuman/ezkerrean	Jokalaria aldapa igotzen hasten da eta oztopoa aurkitzean gelditu egiten da.	ERROR2
Aldapa igo eta salto egin	Jokalaria aldapa igotzen dagoela salto egin dezake.	ERROR3
Aldapa jaitsi oztoporik gabe	Jokalaria aldapa jaisten du abiadura konstante mantenduz eta lurretik altxatu gabe.	OK

PROBA	ESPERO DEN EMAITZA	LORTUTAKO EMAITZA
Aldapa jaitsi oztopoa eskuman/ezkerrean	Jokalaria aldapa jaisten hasten da eta oztopoa aurkitzean gelditu egiten da.	ERROR4
Aldapa jaitsi eta salto egin.	Jokalaria aldapa igotzen dagoela salto egin dezake.	OK
Hormara itsatsi	Jokalaria motelago erortzen da.	OK
Hormara itsatsi eta salto egin	Jokalaria hormaren kontrako noranzkoan salto egiten du.	OK
Korrika egin	Korrika botoia sakatzean jokalaria oinez baino bizkorrago mugitzen da. Oztopo batekin talka egitean gelditu egiten da.	OK
Korrika egin eta salto egin	Jokalaria airean dagoela arinago mugitzen da x ardatzean.	OK
Aldapak korrika igo/jaitsi	Jokalariai aldapak oinez baino bizkorrago igotzen/jaisten ditu.	OK
Korrika egin eta hormaren aurka salto egin eta itsatsi ondoren berriz salto egin	Jokalariai hormaren aurka salto egitean kontrako noranzkoan salto egin daiteke berriz. Abiadura horizontala ohikoa baino handiago da.	OK
Makurtu	Makurtzeko botoia sakatzean jokalariaen altuera txikitzen da eta motelago mugitzen da.	OK
Altxatu	Makurtzeko botoia askatzean jokalariai altuera berreskuratzen du. Gainean sabaia badago jokalaria ez da altxatuko.	ERROR5
Makurtu eta korrika botoiak sakatu	Jokalaria korrika badoa abiadura bizkorrean egingo du eta makurtu egiten denean motel mugituko da.	ERROR6
Makurtu eta salto egin	Jokalariai soilik egingo du salto lehenengo altxatu ahal bada.	OK
Irristatu (korrika egiten dagoela makurtu)	Jokalariaen tamaina murriztu egiten da eta abiadura pixkanaka murrizten da.	OK
Aldapa irristatu (aldapan dagoela makurtu)	Jokalariaen tamaina murriztuko da eta maldaren arabera irristatzen den abiadura bizkorragoa edo motelagoa izango da.	ERROR7
Aldapa irristatu eta salto egin	Jokalariai hasierako tamaina berreskuratuko du aldaparen noranzkoan salto egingo du.	ERROR8
Plataforma mugikorrek jokalaria gora mugitu	Jokalaria gorantz mugitzen da. Eskuma/eskerra mugimenduak behar bezala funtzionatzen dute.	OK
Plataforma mugikorrek jokalaria behera mugitu	Jokalaria beherantz mugitzen da. Eskuma/eskerra mugimenduak behar bezala funtzionatzen dute.	ERROR9
Plataforma mugikorrek jokalaria eskumara/ezkerrera mugitu	Plataformak jokalaria eskumara/eskerrera mugitzen du eta jokalaria plataforma gainera mugitu daiteke.	OK
Plataforma mugikorrek jokalaria gora mugitu eta salto egin	Jokalaria gorantz mugitzen da eta salto egin dezake.	ERROR10
Plataforma mugikorrek jokalaria behera mugitu eta salto egin.	Jokalaria beherantz mugitzen da eta salto egin dezake.	OK

PROBA	ESPERO DEN EMAITZA	LORTUTAKO EMAITZA
Plataforma mugikorrek beste oztopoekin bezala talkak hautematen dira	Jokalaria gelditu egiten da plataforma mugikorrarekin talka egitean eta baita plataforma mugikorraren gainean badago beste oztopo baten aurka talka egiten badu.	OK
Salto egitean plataforma zeharkagarria zeharkatu	Jokalariak gainean daukan plataforma zeharkatu dezake salto egitean. Sabaiaren aurka talka ez da erregistratzen.	OK
Makurtzean plataforma zeharkagarria beherantz zeharkatu	Jokalariak azpian duen plataforma zeharkatu egiten du. Azpian aldagaia gezurra da zorua berriz zapaldu arte.	OK
Eskailerak igo	Jokalariak eskailerak igotzen ditu.	OK
Eskailerak jaitsi	Jokalariak eskailerak jaisten ditu.	OK
Kutxa mugitu	Jokalariak kutxa mugitzen du abiadura berdinarekin. Ez da eskuman/ ezkerrean talkarik hautematen.	OK
Kutxa mugitu eta salto egin	Jokalariak salto egiten du eta kutxa toki berdinean mantentzen da.	OK
Kutxa mugitu eta korrika egin	Jokalariak kutxa pixka bat arinago mugitzen du.	OK
Kutxa mugitu eta makurtu	Jokalariak kutxa askatu eta makurtu egiten da. Jokalariak kutxaren aurka talka egiten du.	OK
Atea zeharkatu	Jokalaria ate baten sartzen da eta beste ate baten irteten da.	OK

- ERROR1: Talka bertikalak konprobatzean, abiadura bertikala 0 jartzen da, baina, mugimendu horizontalean abiadura bertikala aldatzen da berriro ere eta jokalaria oztopoan gora eta behera mugitzen da.
 Arazoa ekiditeko, aldapa igotzeko kodea baino lehen gainean aldagaiaren balioa gezurra dela konprobatzen da.
- ERROR2: Aurreko errorearekin gertatzen den bezala, oztopoa aurkitzen du eta abiadura horizontala gelditu egiten da. Baina, jokalaria aldapa igotzen dagoenez, abiadura bertikala ere badago eta jokalaria gora eta behera mugitzen da.
 Arazoa ekiditeko, aldapa igotzeko kodea baino lehen eskuma edo ezkerreko balioa gezurra dela konprobatzen da.
- ERROR3: Jokalaria aldapa gora doanez, izpiak gora jaurtitzen dira behera jaurti beharrean. Ondorioz, azpian aldagaia gezurra izango da beti eta jokalariak ezingo du salto egin.
 Arazoa konpontzea oso erraza da. Jokalaria aldapa igotzen badago azpian aldagaiaren balioa egia izatea besterik ez da jarri behar.
- ERROR4: Lehen bi arazoekin bezala gertatzen da. Oztopoa aurkitzean abiadura horizontala gelditzen da baina ez bertikala eta jokalaria behera mugitzen saiatzen da.
 Arazoa konpontzeko aldapa jaisteko kodea exekutatu baino lehen talka horizontalak ez gotea konprobatzen da.

- ERROR5: Makurtu botoia askatzean, jokalaria ganean sabaia badauka ez du tamaina berreskuratzen, baina, makurtu botoia berriz sakatzean tamaina berriro murrizten da. Gainera, makurtu botoia askatzen duenean ganean sabaia badauka eta ganean duen oztopotik alhora mugitzen bada, ez du abiadura normala berreskuratzen. Arazoa ebazteko, makurtu aldagaia sortzen da. Jokalaria makurtzean aldagaia egia bihurtzen da eta tamaina behin bakarrik murrizten da. Makurtzeko botoia askatzean jokalaria altxatzeko zain gelditzen da eta abiadura eta tamaina normalak berreskuratzean aldagaia gezurra bihurtzen da berriro.
- ERROR6: Korrika eta makurtu botoiak batera sakatzean abiadurak nahastu egiten dira eta ohikoak baino abiadura bizkorragoak edo motelagoak gertatzen dira. Arazoa ebazteko Korrika aldagaia sortzen da. Makurtu aldagaia egia bada korrika balioa ezin da gezurra izan. Jokalaria korrika badoa eta makurtu botoia sakatzen bada jokalaria makurtu egingo da.
- ERROR7: Jokalaria oso azkar irristatzen da beti, aldaparen malda zein den axola izan gabe. Arazoa ebazteko, jokoan aldapa denak 45º izango dutenez, irristatzeko abiadura zehaztuko da kalkulua egin beharrean.
- ERROR8: Jokalaria tamaina berreskuratzen du, baina aldapa jaisteko abiadura eskuz aldatu dugunez, jokalaria gorantz egiten du salto aldapa irristatzeko abiadura kontuan izan gabe. Arazoa konpontzeko, salto egitean aldapa irristatu aldagaiaren balioa konprobatuko da. Aldagaia egia bada, aldaparen normala erabilita abiadura horizontala ere aldatuko da.
- ERROR9: Jokalaria ez da plataformarekin batera beherantz mugitzen. Arazoa grabitatearen abiadura plataformaren abiadura baino bizkorragoa delako gertatzen da. Plataformaren aurka talka gertatzean abiadura 0 bihurtzen da, baina, plataforma behera mugitzen denez, grabitatea berriro aplikatzen da. Arazoa ebazteko, plataforma ganean kolpe informazioa datu egituran jarriko da. Plataforma ganean egia denean, abiadura bertikala ez da 0 izango.
- ERROR10: Jokalaria gorantz mugitzen denez, izpiak ere gora jaurtitzen dira eta azpian aldagai gezurra da. Arazoa ebazteko, plataforma ganean aldagaia egia denean azpian aldagaia ere egia izango da.

Akatsak konpondu ondoren probak errepikatu dira. [12](#). Taulan errepikatutako proben emaitzak ikusten dira.

12. TAULA: JOKALARIAREN MUGIMENDUAN ERROREA EMAN DUTEN PROBEN ERREPIKAPENA

PROBA	ESPERO DEN EMAITZA	LORTUTAKO EMAITZA
Aldapa igo oztopoa gainean	Jokalaria aldapa igotzen hasten da eta oztopoa aurkitzean gelditu egiten da.	OK
Aldapa igo oztopoa eskuman/ezkerrean	Jokalaria aldapa igotzen hasten da eta oztopoa aurkitzean gelditu egiten da.	OK
Aldapa igo eta salto egin	Jokalaria aldapa igotzen dagoela salto egin dezake.	OK
Aldapa jaitsi oztopoa eskuman/ezkerrean	Jokalaria aldapa jaisten hasten da eta oztopoa aurkitzean gelditu egiten da.	OK
Altxatu	Makurtzeko botoia askatzean jokalaria altuera berreskuratzen du. Gainean sabaia badago jokalaria ez da altxatuko.	OK
Makurtu eta korrika botoiak sakatu	Jokalaria korrika badoa abiadura bizkorrean egingo du eta makurtu egiten denean motel mugituko da.	OK
Aldapa irristatu (aldapan dagoela makurtu)	Jokalariaren tamaina murriztuko da eta maldaren arabera irristatzen den abiadura bizkorragoa edo motelagoa izango da.	ERROR11
Aldapa irristatu eta salto egin	Jokalaria hasierako tamaina berreskuratuko du aldaparen noranzkoan salto egingo du.	OK
Plataforma mugikorak jokalaria behera mugitu	Jokalaria beherantz mugitzen da. Eskuma/eskerra mugimenduak behar bezala funtzionatzen dute.	OK
Plataforma mugikorak jokalaria gora mugitu eta salto egin	Jokalaria gorantz mugitzen da eta salto egin dezake.	OK

- ERROR11: Aldapa jaisten ari dela, momentu txiki batzuetan airean dagoela pentsatzen du. Ondorioz, aldapa irristatu aldagaia eta abiadurak kendu egiten dira. Arazo hau ebazteko, aldapa irristatu aldagaia gezurra izateko airean segundo erdia baino denbora gehiago egon behar du.

Errore berria konpondu ondoren proba errepikatu da. 13. Taulan errepikatutako probaren emaitza ikusten da.

13. TAULA: JOKALARIAREN MUGIMENDUAN ERROREA EMAN DUEN PROBAREN ERREPIKAPENA

PROBA	ESPERO DEN EMAITZA	LORTUTAKO EMAITZA
Aldapa irristatu (aldapan dagoela makurtu)	Jokalariaren tamaina murriztuko da eta maldaren arabera irristatzen den abiadura bizkorragoa edo motelagoa izango da.	OK

6.1.6.3 Animazioak

Jokalariaren mugimenduarekin gertatzen den bezala, animazioentzat arauak sortzen diren heinean aurreko animazioak zapaltzen ez direla eta guztien funtzionamendua zuzena dela konprobatu behar da. Egindako probak eta emaitzak 14. eta 15. tauletan agertzen dira.

14. TAULA: JOKALARIAREN ANIMAZIO PROBEN EMAITZAK

PROBA	ESPERO DEN EMAITZA	LORTUTAKO EMAITZA
Geldi animazioa	Jokalaria mugitzen ez dagoenean geldi dagoen animazioa ikusten da.	OK
Oinez animazioa	Jokalaria ezkerrera/eskumara mugitzean oinez animazioa ikusten da. Animazioaren noranzkoa mugimenduarekin zuzena da.	OK
Salto egin animazioa	Jokalariak salto egitean salto animazioa ikusten da.	OK
Hormara itsatsi animazioa	Jokalariak paretara itsastean horma itsatsi animazioa ikusten da. Hormatik urruntzean salto animazioa jarraitzen du.	ERROR1
Korrika egin animazioa	Jokalaria korrika egitean korrika egin animazioa ikusten da.	OK
Makurtu animazioa	Jokalaria makurtzean makurtu animazioa ikusten da. Jokalaria mugitzen bada, makurtuta mugitu animazioa ikusten da. Altxatzean geldu animazioa edo oinez animazioa ikusten da.	OK
Irristatu animazioa	Korrika egin ondoren jokalaria makurtu egiten bada irristatu animazioa ikusten da. Abiadura nahikoa murriztu ondoren makurtu animazioa ikusten da.	OK
Aldapan irristatu animazioa	Aldapan makurtzean aldapa irristatu animazioa ikusten da. Aldapa bukaerara heltzean makurtu animazioa ikusten da. Aldapa bukaerara heldu baino lehen altxatzen bada geldu edo oinez animazioa ikusten da.	ERROR2
Korrika plataforma mugikorraren gainean	Jokalariak korrika egiten duenean korrika animazioa ikusten da.	OK
Makurtu plataforma mugikorraren gainean	Jokalaria makurtzen duenean makurtu animazioa ikusten da.	OK
Korrika plataforma zeharkagarriaren gainean	Jokalariak korrika egiten duenean korrika animazioa ikusten da.	OK
Makurtu plataforma zeharkagarriaren gainean	Jokalaria makurtzen bada plataforma zeharkatzen du. Ez da makurtu animazioa ikusten.	ERROR3
Eskailera igo	Gora gezia zapaltzean eskailera igotzeko animazioa ikusten da. Gezia eskailera erdian altxatzen bada, animazioa geldu mantentzen da.	OK

PROBA	ESPERO DEN EMAITZA	LORTUTAKO EMAITZA
Eskailerak jaitsi	Behera gezia zapaltzean eskailera jaisteko animazioa ikusten da. Gezia eskailera erdian altxatzen bada, animazioa gelditu mantentzen da.	ERROR4
Kutxa mugitu	Kutxara gerturatzean kutxa mugitzeko animazioa ikusten da. Jokalaria gelditzen denean animazioa gelditu mantentzen da.	ERROR5
Atea zeharkatu	Ate aurrean behera gezia zapaltzean atean sartzeko animazioa ikusten da. Segundo bat pasatu ondoren beste ate batetan atetik ateratzeko animazioa ikusten da.	ERROR6

- ERROR1: Horma itsatsi animazioa ikusten da pantailan, baina, noranzkoa okerra da. Arazoa konpontzeko horma itsatsi animazioarentzat irudia alderantziz ikuskatuko da.
- ERROR2: Aldapa jaisten dagoela, jokoak jokalaria momentu txiki baten airean dagoela pentsatzen du. Arazoa ekiditeko, animazio ez da kenduko airean segundo erdia baino gehiago egon arte.
- ERROR3: Plataforma zeharkatu baino lehen, makurtu animazioa jartzen saiatzen da eta mugimendu arraro bat agertzen da. Arazoa ebazteko, azpian dagoen plataforma zeharkagarria bada makurtu ekintza ezgaituko da.
- ERROR4: Animazioa behar bezala funtzionatzen du, baina, jokalaria eskailera azpian dagoenean, behera gezia zapaltzen bada, ez da zorua hautematen eta lur azpian gelditzen da jokalaria. Arazoa ebazteko, eskailera objektuaren irudia lurrera heltzen dela ikusiko da, baina *Collider*-a altuera jakin batetik aurrera hasiko da. Horrela, eskailera igo daiteke, baina, behar bezala ez ingo du zorua zeharkatu.
- ERROR5: Animazioaren funtzionamendua egokia da, baina, jokalaria kutxa mugitzen dagoela kontrako noranzkoan pixka bat mugitzen bada, animazioa ikusten jarraitzen da, alderantziz. Arazoa ebazteko, kutxa bultzatu animazioa jartzeko, jaurtitzen diren izpiak kutxa batekin egin behar dute talka.
- ERROR6: Atea zeharkatzeko animazioa jarri baino lehen jokalaria makurtu egiten da. Portaera hau ebazteko, jokalaria ezin izango du makurtu ekintza egin ate baten aurrean dagoenean.

Akatsak konpondu ondoren probak errepikatu dira. 15. Taulan errepikatutako proben emaitzak ikusten dira.

15. TAULA: JOKALARIA ANIMATZEKO ERROREA EMAN DUTEN PROBEN ERREPIKAPENA

PROBA	ESPERO DEN EMAITZA	LORTUTAKO EMAITZA
Hormara itsatsi animazioa	Jokalariak paretara itsastean horma itsatsi animazioa ikusten da. Hormatik urruntzean salto animazioa jarraitzen du.	OK
Aldapan irristatu animazioa	Aldapan makurtzean aldapa irristatu animazioa ikusten da. Aldapa bukaerara heltzean makurtu animazioa ikusten da. Aldapa bukaerara heldu baino lehen altxatzen bada geldu edo oinez animazioa ikusten da.	OK
Makurtu plataforma zeharkagarriaren gainean	Jokalaria makurtzen bada plataforma zeharkatzen du. Ez da makurtu animazioa ikusten.	OK
Eskailerak jaitsi	Behera gezia zapaltzean eskailera jaisteko animazioa ikusten da. Gezia eskailera erdian altxatzen bada, animazioa geldu mantentzen da.	OK
Kutxa mugitu	Kutxara gerturatzean kutxa mugitzeko animazioa ikusten da. Jokalaria gelditzen denean animazioa geldu mantentzen da.	OK
Atea zeharkatu	Ate aurrean behera gezia zapaltzean atean sartzeko animazioa ikusten da. Segundo bat pasatu ondoren beste ate batetan atetik ateratzeko animazioa ikusten da.	OK

6.1.6.4 Kamera

Kameraren funtzionamendua zuzena da, baina jarritako propietate berriak funtzionatzen dutela konprobatu behar da. Egindako probak eta emaitzak 16. taulan daude.

16. TAULA: KAMERA PROBEN EMAITZAK

PROBA	ESPERO DEN EMAITZA	LORTUTAKO EMAITZA
Kamera jokalaria jarraitu	Kamera jokalaria jarraitzen du adierazitako eremuetatik irteten ez den bitartean.	OK
Atea zeharkatu eta kamera muga aldatu	Jokalaria agertzen den puntu berrian beste kamera muga bat badago aldatu egiten da.	OK

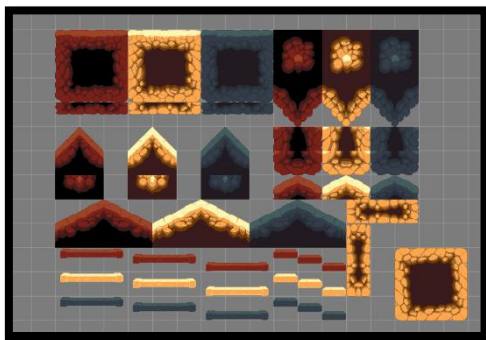
6.2 Mapa eta bizitzak

Prototipo honetan jokalaria mugituko den lur-eremua prestatuko da, jokalaria bizi sistema bat jarriko zaio, hauek galtzeko tranpa pare bat garatuko dira eta bihotz denak galtzen baditu berriro agertzeko *checkpoint*-a ere inplementatuko da.

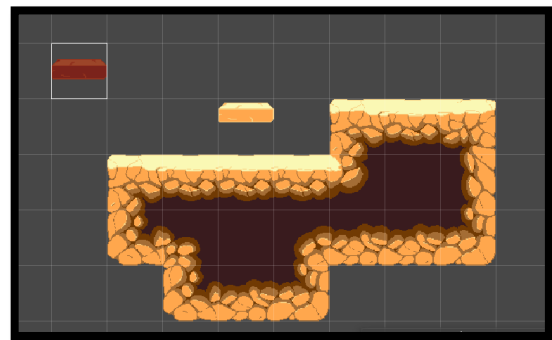
6.2.1 Lur-eremua

Jokalariaren mugimendua inplementatuta dagoela, jokalaria mugitzeko lur-eremua prestatuko da. *TileMap* osagaia erabilia eszenatokia irudiz bete da eta jokalaria irudi hauen gainean mugitu ahal izateko *Collider* osagaia jarri zaie irudiei.

Unity-ren *Tilemap* osagaia erabilia eszenatokia laukietan banatzen da eta sortutako lauki bakoitzean *Tile* bat jarri daiteke. Irudi guztien multzoa edo *TileSet*-a definitu ondoren, eszenatokiko lauki bakoitzean *Tile* bat jarri daiteke, mosaikoak bezala, eta nahi den mapa sortu daiteke. 19. irudian erabilitako *TileSet*-a ikus daiteke eta 20. irudian mapa txiki baten adibidea.

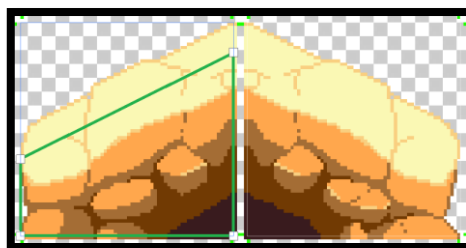


19. IRUDIA: *TILESET*-A EDO IRUDI SORTA



20. IRUDIA: *TILEMAP* MAPA ADIBIDEA

Tilemap-ari *Collider* osagaia jartzean, jokalaria irudien gainetik mugitzeko gai izango da. Unity-k *Tile* bakoitzaren *Collider*-aren forma eta tamaina automatikoki kalkulatzen du irudia erabilia. Sortzen duen talka geruza egokia ez denez, *Tile* bakoitzaren *Collider*-a eskuz definitu da, 21. irudian ikus daitekeen bezala.



21. IRUDIA: *TILE* BATEN *COLLIDER* OSAGAIA

22. Irudian, eszenatokia apaintzen duten atzeko horma eta zutabeak ere *TileMap* erabilia marraztu dira. Irudi hauei *Collider* osagaia jartzen bazaie jokalaria trabatuta geratuko da. Aurrerago arazoak ez izateko, irudiak funtzioaren arabera sailkatu dira (oztopoak, apaingarriak, mugikorrak eta zeharkagarriak) eta multzo bakoitzari geruza bat esleitu zaio. Gero, izpiak jaurtitzeko funtzioan zeintzuk geruza antzeman behar diren adierazi da.



22. IRUDIA: MAPA ADIBIDEA

23. Irudian, geruza jakin bateko talkak soilik hautemateko erabili den kodea zelakoa den ikus daiteke. Kasu honetan, soilik 'kolpeGainazalak' parametroak adierazten duen geruza multzoa hautemango da.

```
RaycastHit2D kolpea = Physics2D.Raycast(izpia, Vector2.down, Mathf.Infinity, kolpeGainazalak);
```

23. IRUDIA: KOLPEGAINAZALAK GERUZA MULTZOA SOILIK ANTZEMATEKO FUNTZIOA.

6.2.2 Bizitza sistema

Jokoan erronka pixka bat jartzeko, jokalariai bihotzetan oinarritutako bizitza sistema bat jarri zaio. Gero, jokalaria minduko duten elementuak eta bihotzak berreskuratzeko edo kopurua areagotzeko itemak sortu dira.

Jokalariai 3 bihotz ditu eta min hartzen duen bakoitzean bihotz erdia galduko du. 24. Irudian bihotz erdia galdu ondoren interfazea zelakoa den ikus daiteke. Jokoan zehar zailtasuna areagotuz joango denez, jokalariai bihotz kopurua areagotzeko aukera eskainiko zaio. Lortu daitezkeen bihotz kopuru maximoa 6 bihotz dira.



24. IRUDIA: JOKALARIAREN BIZITZA INTERFAZEA

Jokalariak dituen bihotz kopurua, bihotz horietatik zenbat dituen osorik eta bihotz kopuru maximoa 'Inbentario' klasean biltzen dira. Jokalariak min hartzen duenean bihotz erdia soilik galtzen duenez, zenbaketa bihotz erdietan egitea erabaki da. Horrela, interfazea eguneratzea askoz errazagoa izango da. Klaseak bihotzak galtzeko, bihotzak berreskuratzeko eta bihotzak areagotzeko metodoak ditu eta hauetako bakoitza exekutatzen denean interfazea eguneratzeko agindua bidaltzen da. 'Inbentario' klasea bihotzen balioak kudeatzen dituen bakarra izatea nahi da. Horretarako, 25. irudian ikus daitekenez, klaseak *Singleton* patroia izango du.

```
public static Inbentarioa instantzia;  
  
private void Awake()  
{  
    if(instantzia != null)  
    {  
        Debug.LogWarning("SingletonError: Inbentario bat baino gehiago aurkitu da");  
        return;  
    }  
    instantzia = this;  
}
```

25. IRUDIA: INBENTARIOAREN SINGLETON PATROIA

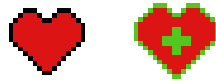
Sortutako mehatxuak, pintxoak, amildegiak edo beste tranpak, minutako objektua jokalaria dela jakin behar dute. Horretarako, jokalaria 'Player' geruzan sailkatuko da. Jokalaria nor den jakinda eta inbentarioa erabilia bihotz kopurua murriztu daiteke. 26. Irudian jokalaria ukitzean mintzeko funtzioak zelako itxura duen ikus daiteke.

```
private void OnTriggerEnter2D(Collider2D collision)  
{  
    if (collision.tag == "Player")  
    {  
        inbentarioa.KolpeaJaso(bihotzKopurua);  
    }  
}
```

26. IRUDIA: JOKALARIA KONTAKTUAN MINTZEKO FUNTZIOA

Bihotzak berreskuratzeko bi item ezberdin sortu dira: bihotza, bihotz oso bat berreskuratzeko eta bihotz berdea, galdutako bihotz denak berreskuratzeko eta bihotz kopuru

maximoa areagotzeko. Tranpekin bezala, jokalaria itemaren *trigger* barruan sartzen denean aktibatzen da eta inbentarioa erabiliz dagokion efektua aplikatzen da. Itemak erabilera bakarra dute, ondorioz, jokalaria hartzen dituzenean desagertu egiten dira. 27. Irudian, eskerretik eskumara, bihotzak berreskuratzeko itema eta bihotzak areagotzeko itema agertzen dira.



27. IRUDIA: BIHOTZAK BERRESKURATZEKO ITEMAK

6.2.3 CheckPoint-ak

Jokoetan jokalaria bihotzik gabe gelditzen denean partida galtzen du eta *respawn*⁷ egiten duenean azkeneko zatia berriro errepikatu behar du. Jokalariaren hil eta berpiztu egoerak eta *checkpoint* funtzionalitatea inplementatzeko 'JokalariKudeatzailea' klasea eta objektua sortuko da.

Checkpoint-a joko osoa ez errepikatzeko balio duten elementuak dira. Jokalaria hil ondoren *respawn* egiten duenean, zeharkatutako azken *checkpoint*-ean egingo du. Jokoan dauden *checkpoint* guztiak 'JokalariKudeatzailea' objektuaren 'semeak' izango dira. Horrela, klaseak *checkpoint* denak eskuragarri izango ditu zerranda baten bezala. Klasearen atributu nagusia 'gordePuntua' da. Atributu honek jokalaria zeharkatu duen azken *checkpoint*-a zein den adierazten du. Klasearen *Start()* metodoan (jokoa hasi bezain laster exekutatu da) 'gordePuntua' atributuaren balioa lehenengo semearekin eguneratzen da.

Tranpak eta itemak bezala, *checkpoint*-a *trigger* batekin funtzionatzen du eta jokalaria bakarrik eragiteko prestatuta dago. Honek, *trigger*-a eragiten duenean, 'gurasoaren' 'JokalariKudeatzailea' klaseko 'gordePuntua' atributuaren balioa bere buruarekin ordezkatu du 28. irudian ikusten den bezala.

```

private void OnTriggerEnter2D(Collider2D collision)
{
    if (collision.tag == "Player")
    {
        JokalariKudeatzailea jokalariKud = GetComponentInParent<JokalariKudeatzailea>();
        if (jokalariKud != null)
        {
            jokalariKud.gordePuntua = gameObject;
        }
    }
}

```

28. IRUDIA: CHECKPOINT-A ALDATZEKO KODEA

⁷ **Respawn**: Joko barruko entitatea hil ostean berriro agertzeko ekintzari *respawn* egitea deritzo.

Azkenik, hil eta berpiztu egoerak jarri dira. Jokalaria bihotzik gabe gelditzen denean hil egiten da eta hiltzeko animazioa ikusten da. Jokalaria hil denez, mugimendua ezgaitu egiten da. Ondoren, jokalaria azken gorde puntura mugitzen da⁸ *respawn* egiteko eta inbentarioa erabilita bihotz denak berreskuratzen ditu. Momentu horretan, berpizteko animazioa jartzen da eta bukatu ondoren jokalariaren mugimendua berriro ahalbidetzen da.

6.2.4 Probak

Atal honetan irudikatutako mapa, jokalariaren bizitza-sistema eta *checkpoint*-ak behar bezala funtzionatzen dutela ziurtatuko da.

6.2.4.1 Lur eremua

Jokalariak marraztutako irudien gainean behar bezala mugitu daitekeela konprobatu da. Ekintza guztiak ez dira zertan konprobatu behar, irudiak aldatu diren arren *Collider*-n gainazalaren forma berdinak dira.

6.2.4.2 Bizitza sistema

Inbentarioaren funtzionamendua zuzena dela konprobatzeko sortutako itemak eta tranpak inbentarioan datuak aldatzen dituztela eta interfazea inbentarioko datuak pantailaratzen dituela ziurtatu da. Proben emaitzak 17. taulan daude.

⁸ Unity-n, objektu guztiak posizio bat dutenez, jokalaria 'gordePuntua' objektuaren posiziora mugitu daiteke.

17. TAULA: BIZITZA SISTEMAREN PROBEN EMAITZAK

PROBA	ESPERO DEN EMAITZA	LORTUTAKO EMAITZA
Pintxoak zapaldu 3 bihotzekin	Inbentarioan bihotz kopurua 5 da ⁹ . Interfazean bi bihotz eta erdi ikusten dira.	OK
Pintxoak zapaldu bihotz erdiarekin	Inbentarioan bihotz kopurua 0 da. Interfazean ez da bihotzik ikusten.	OK
Pintxoak zapaldu bihotzik gabe	Inbentarioan bihotz kopurua 0 da. Interfazean ez da bihotzik ikusten.	OK
Amildegian jauzi 3 bihotzekin	Inbentarioan bihotz kopurua 0 da. Interfazean ez da bihotzik ikusten.	OK
Bihotzak berreskuratzeko itema hartu 3 bihotzekin	Inbentarioan bihotz kopurua 6 da. Interfazea ez da aldatzen. Itema ez da hartzen.	OK
Bihotzak berreskuratzeko itema hartu bihotz erdiarekin	Inbentarioan bihotz kopurua 3 da. Interfazean bihotz bat eta erdi ikusten dira. Itema hartzen da.	OK
Bihotzak berreskuratzeko itema hartu bi bihotz eta erdiarekin	Inbentarioan bihotz kopurua 6 da. Interfazean hiru bihotz ikusten dira. Itema hartzen da.	OK
Bihotzak areagotzeko itema hartu 3 bihotzekin	Inbentarioan bihotz kopurua 8 da. Interfazean 4 bihotz ikusten dira. Itema hartzen da.	OK
Bihotzak areagotzeko itema hartu bihotz erdiarekin	Inbentarioan bihotz kopurua 8 da. Interfazean 4 bihotz ikusten dira. Itema hartzen da.	OK
Bihotzak areagotzeko itema hartu sei bihotzekin	Inbentarioan bihotz kopurua 12 da. Interfazea ez da aldatzen. Itema ez da hartzen.	OK

6.2.4.3 Checkpoint-ak

Jokalariak jokoan aurrera doala gorde puntua eguneratzen dela eta bere funtzionamendua zuzena dela ziurtatzeko 18. taulan dauden probak burutu dira.

⁹ Inbentarioan bizitza bihotz erdietan kontaktzen da interfazea eguneratzeko errazagoa delako.

18. TAULA: CHECKPOINT-EN PROBEN EMAITZAK

PROBA	ESPERO DEN EMAITZA	LORTUTAKO EMAITZA
Jokalariak bihotz denak galtzen ditu	Hil animazioa jartzen da eta input-a ez dabil. Jarraian berpiztu animazioa jartzen da eta input-a berriro erabilgarri dago.	OK
Jokalaria partida hasi bezain laster bihotzak galtzen ditu	Jokalaria lehenengo <i>checkpoint</i> -ean agertzen da. Bihotz guztiak berreskuratzen ditu.	OK
Jokalaria <i>checkpoint</i> bat zeharkatzen du bihotzak galdu baino lehen	Jokalariak zeharkatutako <i>checkpoint</i> -ean agertzen da.	OK
Etsai bat <i>checkpoint</i> bat zeharkatzen du	Jokalariaren 'gordePuntua' ez da eguneratzen.	OK

6.3 Eraso sistema

Jokoan erronka gehiago jartzeko eta alde ludikoarekin bukatzeko, jokalariari eraso ekintzak jarri zaizkio eta mugimendu eta portaera ezberdinak dituzten etsaiak inplementatu dira.

6.3.1 Jokalaria

Jokalariak eszenatokian aurkituko dituen etsaiak garaitzeko eraso mugimenduak sortu dira. Jokalariak etsai bat kolpatzen badu etsaiari bere bizitza puntuak murriztuko zaizkio eta punturik gabe gelditzen denean eszenatokitik desagertuko da.

Jokalariaren eraso ekintzak dagokion itema lortu ondoren ahalbidetzen dira. Item horiek lastargia, ezpata eta arkuak dira eta bakoitzarentzako animazio bat sortu da. Lastargiak etsai txikiekin bukatzeko balio du, baina eginkizun nagusia iluntasunean ikustea da. Ezpatak lastargia baino min gehiago ematen du eta urrunago iristen da. Arkuarekin urrunetik eraso daiteke, baina eraso bakoitzak gezi bat erabiltzen du eta gezirik gabe gelditzean arkuak ez du erasotzeko balio. Geziak berreskuratzeko gezi itema sortu da. Jokalariak hiru itemak izan arren, ezin dira denak aldi berean erabili. Horrek esan nahi du item erabilgarri bat egongo dela eta erabiltzaileak hiruren artean zein erabili nahi duen erabaki behar duela. Item erabilgarria edozein momentuan aldatu daiteke, baldin eta inbentarioan item bat baino gehiago badago.

Inbentarioan item hauen informazio dena gordetzen da eta bihotzekin bezala, aldaketak egiten direnean interfaze grafikoak pantailaratu egiten ditu. 29. Irudian erasotzeko itemak duen interfazea ikusi daiteke. Bertan, momentuan erabili daitekeen itema, itemak zelan aldatu eta arkuari gelditzen zaizkion gezi kopurua adierazten da.



29. IRUDIA: ITEM INTERFAZEA

6.3.2 Etsaiak

Etsaiak jokalariaren aurka doazen entitateak dira. Etsai mota bakoitzak bizitza puntu eta erasotzeko portaera ezberdinak ditu. Etsaien eraso ekintzaz gain, jokalariak etsai bat ukitzen duenean ere bihotz erdia galduko du. Hauek dira jokoan dauden etsaiak keta portaerak:

- **Saguzarra.** Etsai honek bizitza puntu gutxi ditu, jokalariak eraso batekin garaitu dezake. Saguzarrak hegan eginez mugituko da jokalariaren noranzkoan eta noizbehinka abiadura areagotuko du haserre balego bezala. Ez dauka eraso berezirik, jokalaria harrapatzen saiatzen da bakarrik. Bi mota daude: txikia, eszenatokiko objektuak edo jokalaria ukitzean suntsitzen dena eta handia, jokalariaren erasoak bakarrik garaitu dezakeena. 30. Irudian saguzar txikiak eta handiak duten itxura ikusten da.



30. IRUDIA: SAGUZAR ETSIAIK

- **Momia eta momia besoa.** Momia jokalariaren noranzkoan mugitzen da, oinez eta oso geldo. Nahikoa gerturatzen denean zaplazteko bat ematen saiatuko da. Bizitza puntu nahikoak ditu 2-3 kolpe jasateko eta garaitu ostean besoa erortzen zaio. Besoak portaera berdina du, baina pixka bat bizkorragoa da. 31. Irudian momiak eta momia besoak duten itxura ikusten da.



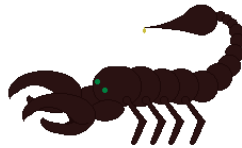
31. IRUDIA: MOMIA ETSAIA

- **Kakalardoa.** Etsai honek portaera pasiboa dauka. Ez du jokalariaren presentzia hautematen, soilik kaka bolak prestatu eta jaurtitzen ditu. Bola hau eszenatokitik biratuz doa apurtu egiten den arte. Jokalariak gaineratik salto edo apurtu egin beharko ditu minik hartu nahi ez badu. Kakalardoa eta honek botatzen dituen kaka bolak eraso batekin suntsitu daitezke. 32. Irudian kakalardoak duen itxura ikusten da.



32. IRUDIA: KAKALARDOA

- **Eskorpioia.** Eskorpioia ez da ia mugitzen, baina jokalari oso urrunetik ikusi dezake eta pozoia jaurtiko dio. Momia bezala 2-3 kolpe behar dira eskorpioia gainditzeko. 33. Irudian eskorpioiak duen itxura ikusten da.



33. IRUDIA: ESKORPIOI ETSAIA

- **Soldadua.** Jokalariaren antzeko etsaia da. Guztietatik mehatxu handiena da eta bizitza puntu gehien duena: 6-8 kolpe behar dira soldadua garaitzeko. Eszenatokitik mugitzen da ezkerretik eskumara mugituz. Jokalaria ikusten badu 3 eraso erabili ditzake: Korrika sprintean mugitu lantzarekin erasotzen, lantza aurrera jaurti eta bi aldeetara kolpatu. 34. Irudian soldaduak duen itxura ikusten da.



34. IRUDIA: SOLDADU ETSAIA

Etsaien mugimendua implementatzeko Unity-k eskaintzen duen fisika sistema erabili da garapen lana ahalik eta gehien errazteko. *RigidBody* osagaia erabilia, ez da grabitate indarra kalkulatu behar eta talkekin erreakzioak ere Unity-k kudeatzen ditu. Azkenik, etsaien artean gertatzen diren talkak kentzeko, etsai denak talka geruzan baten sailkatu dira eta kolpe gainazalen matrizean etsaiak etsaiekin talka ezgaitu da.

6.3.3 Probak

Atal honetan etsaien mugimendua, eraso patroia, jokalariaren ekintza berriak eta orokorrean eraso-sistemaren funtzionamendua zuzena dela konprobatuko da.

6.3.3.1 Jokalaria

Atal honetan jokalariaren ekintza berriak funtzionamendu egokia dutela, inbentarioan agertzen diren datuak egokiak direla, animazio berriak behar den momentuan ikusten direla eta interfazeak datu egokiak erakusten dituela konprobatu da. Proben emaitzak 19. eta 20. tauletan aurkitzen dira

19. TAULA: JOKALARIAREN ERASOEN PROBEN EMAITZAK

PROBA	ESPERO DEN EMAITZA	LORTUTAKO EMAITZA
Jokalariak ez duen itema hartzen du	Jasotako itema inbentarioan gordetzen da eta interfazeak pantailaratu egiten du	OK
Jokalariak dagoeneko daukan itema hartzen du	Itema lurrean gelditzen da. Inbentarioan eta interfazean ez da baliorik eguneratzen.	OK
Itemak aldatu	Jokalariak itema erabilgarria aldatzen du. Item bakoitzarekin eraso ezberdina egiten du. Interfazean item erabilgarria momentu bakoitzean zein den adierazten da.	OK
Lastargiarekin eraso	Etsaia kolpatzen denean bizitza puntuak murrizten zaizkio. Lastargiaren eraso animazioa ikusten da.	ERROR1
Ezpatarekin eraso	Etsaia kolpatzen denean bizitza puntuak murrizten zaizkio. Ezpataren eraso animazioa ikusten da	OK
Arkuarekin eraso	Geziak etsaia kolpatzen badu bizitza puntuak murrizten zaizkio. Arkuaren eraso animazioa ikusten da. Gezi kopurua murrizten da.	OK
Arkuarekin eraso gezirik izan gabe	Ez da gezirik jaurtitzen. Gezi kopurua 0 izaten jarraitzen du. Ez da eraso animazioa jartzen. Etsaiak ez du kolperik jasotzen.	OK
Itema aldatu item bakarrarekin	Jokalariak ez du itema aldatzen. Eraso ez da aldatzen eta interfazea ez da aldatzen.	OK
Lastargia iluntasunean	Jokalariaren inguruan eszenatokia ikusten da. Eraso ondoren ikusten den distantzia denbora batez handitzen da.	OK

- ERROR1: Etsaia mindu egiten da eta animazioa jartzen da, baina jokalaria alboetara mugitu daiteke eta salto egin dezake. Ondorioz animazioa ez da ondo ikusten. Arazoa konpontzeko, eraso animazioa irauten duen bitartean mugimendua ezgaituko da.

Errorea konpondu ondoren proba errepikatu da. 20. Taulan errepikatutako probaren emaitza ikusten da.

20. TAULA: JOKALARIAREN ERASORAKO ERROREA EMAN DUEN PROBAREN ERREPIKAPENA

PROBA	ESPERO DEN EMAITZA	LORTUTAKO EMAITZA
Lastargiarekin eraso	Etsaia kolpatzen denean bizitza puntuak murrizten zaizkio. Lastargiaren eraso animazioa ikusten da.	OK

6.3.3.2 Etsaiak

Etsai bakoitzaren mugimendu eta eraso ekintzak eta ekintza bakoitzaren animazioak egokiak direla konprobatzeko 21. eta 22. tauletan agertzen diren testak burutu dira.

21. TAULA: ETSAIEN MUGIMENDU ETA ERASOEN PROBEN EMAITZAK

PROBA	ESPERO DEN EMAITZA	LORTUTAKO EMAITZA
Etsaiak jokalaria mintzen du	Jokalariak bihotz erdia galtzen du.	ERROR1
Etsaiak jokalaria jarraitzen du.	Etsaiak jokalaria jarraitzen du ikusten duen bitartean. Jokalaria ez dagoenean normal mugitzen da.	ERROR2
Saguzarra jokalaria jarraitu	Saguzarra jokalaria jarraitzen du. Saguzar txikia eszenatokiarekin edo jokalariarekin suntsitzen da. Hegan egiteko animazioa ikusten da.	OK
Saguzarra jokalaria eraso	Saguzarrak erasotzen duenean abiadura areagotzen du eta animazioa bizkorrago ikusten da. Segundo batzuen ostean egoera normalera itzultzen da.	OK
Momia jokalaria jarraitu	Momia jokalaria jarraitzen du eszenatokitik oinez mugituz. Oinez animazioa ikusten da.	OK
Momia jokalaria eraso	Jokalaria momiatik gertu dagoenean eraso egiten du. Eraso animazioa ikusten da.	OK
Jokalariak momia suntsitu	Momia suntsitzen denean besoa erortzen zaio. Besoak momiaren portaera berdina du, baina bere animazioekin.	OK
Kakalardoak kaka bolak sortu eta jaurti egiten ditu.	Kakalardoaren eraso animazioa ikusten da. Kaka bolak eszenatokitik biratuz mugitzen da.	OK

PROBA	ESPERO DEN EMAITZA	LORTUTAKO EMAITZA
Eskorpioia mugitu	Eskorpioia aurrera eta atzera mugitzen da eszenatoki gainean. Mugitzeko animazioak ikusten dira.	OK
Eskorpioia eraso	Eskorpioiak pozoia jaurtitzen du jokalariaren noranzkoan. Pozoia jokalaria kolpatzean edo eszenatokiaren aurka kolpatzean suntsitzen da. Eraso animazioa ikusten da.	OK
Soldadua mugitu	Soldadua dagoen gelako bi puntuen artean mugitzen da. Mugitzeko animazioa ikusten da.	OK
Soldadua sprint erasoa	Soldadua jokalariaren noranzkoan mugitzen da oso azkar. Sprint erasoa animazioa ikusten da.	OK
Soldadua lantza jaurti erasoa	Soldadua lantza jaurtitzen du jokalariaren noranzkoan eta lerro zuzenean. Lantza jaurtitzeko animazioa ikusten da.	OK
Soldadua lantzarekin kolpatu erasoa	Soldadua jokalariari gerturatzen da lantzarekin kolpatzen saiatzeko. Lantza erasoa animazioa ikusten da.	OK

- **ERROR1:** Jokalariak kolpeak oso azkar jasotzen ditu eta bihotz erdia baino gehiago galtzen du.

 Arazoa ekiditeko, jokalariari garaitezkin egoera eta animazioa jarri zaio kolpea jaso eta ondorengo segundoan. Gainera, min hartzen duenean kolpea jaso duen kontrako zentsuan mugituko da.
- **ERROR2:** Etsaiak jokalaria jarraitzen du. Jokalaria desagertzen denean toki berri baten aurkitzen da etsaia. Toki berri horretan, aldapa baten edo bi gelaren artean adibidez, mugimendua ez da guztiz egokia eta dauden gelatik ihes egin dezakete.

 Portaera arazoak ekiditeko, jokalariak gela aldatzen duenean etsaiak ezgaitu egingo dira. Gela horretan berriz sartzean gelako etsaiak hasierako tokian agertuko dira, garaitutakoak barne.

Akatsak konpondu ondoren probak errepikatu dira. **22.** Taulan errepikatutako proben emaitzak ikusten dira.

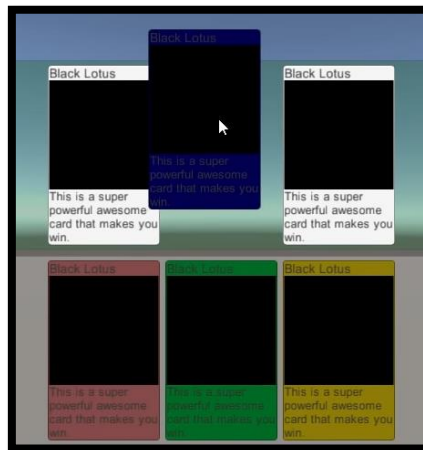
22. TAULA: ETSAIEN MUGIMENDU ETA ERASOAN ERROREA EMAN DUTEN PROBEN ERREPIKAPENA

PROBA	ESPERO DEN EMAITZA	LORTUTAKO EMAITZA
Etsaiak jokalaria mintzen du	Jokalariak bihotz erdia galtzen du.	OK
Etsaiak jokalaria jarraitzen du.	Etsaiak jokalaria jarraitzen du ikusten duen bitartean. Jokalaria ez dagoenean normal mugitzen da.	OK

6.4 Drag eta Drop funtzionalitateak

Ariketak egiteko, jokalariai behar dituen kode lerroak eskainiko zaizkio. Lerro kodeak kolore eta forma ezberdineko piezetan idatzita egongo dira eta piezak ordena eta toki zuzenean jarri beharko dira ariketa ebazteko. Jokoak konbinazio ezberdinak bereizteko gai izango da erabiltzailearen emaitza zuzena den jakiteko eta pieza bakoitzak zer egiten duen jakiteko funtzionalitatea ere izango du.

Piezak mugitzeko portaera, karta jokoetan oinarrituta dago [11]. Kartak eskutik mahaira eta alderantziz mugitzen dira. Karta bat altxatzean, hutsune bat sortzen da karta zegoen tokian. Karta pantailatik mugitzean hutsunea ere mugitzen da karta non jarriko den adierazteko. 35. Irudian, karta urdina mugitu da eta sortutako hutsuneak karta zuriak alboratu ditu. Horrela, karta urdina bi karta zurien artean jarriko dela ikusten da.



35. IRUDIA: KARTA JOKOA

Piezak mugitzeko 'Drag' klasea sortu da. Klase honek, saguarekin objektua pantailatik mugitzea ahalbidetzen du eta ariketako pieza guztietan egongo da. Horretaz gain, ariketa berriz hasteko eta mugimendu oker baten ondorioz pieza hasierako posiziora itzultzeko funtzionalitateak jarri dira. Lehenengo funtziorako, exekuzio hasieran, piezaren 'guraso' eta posizioak gordetzen dira. Ariketa berriz hasteko deia jasotzean pieza guztiak hasierako posiziora itzultzen dira. Bigarren funtziorako, pieza bat mugitu aurretik duen 'guraso' eta posizioak gordetzen ditu. Pieza bat maiz tokiz aldatu daiteke, ondorioz, bi balioak aldagarriak dira, lehen kasuan ez bezala.

'Drag' klasearekin hartutako piezak zein tokitan jarri daitezkeen adierazteko 'Drop' klasea sortu da. Klase hau ariketa leihoan, agindu guztiak dauden leihoan edo beste pieza batzuen barnean dago soilik. 'Drag' klaseak piezaren mugitu aurreko posizioa gordeta duenez, pieza bat 'Drop' klasea ez duen toki baten uztean, aurreko posiziora itzultzen da. Horretaz gain, 'Drop' klasea duen tokian pieza bakarra izateko funtzionalitatea ere inplementatu da. Horrela, pieza bat dagoeneko okupaturik dagoen tokian jartzen bada, posizioak aldatu egingo dira:

mugitzen ari den pieza hutsunean jarriko da eta hutsunean zegoen pieza mugitzen zegoen piezaren hasierako tokira joango da.

36. Irudian, ariketa bateko piezak ikusten dira. ‘Drag’ osagaia duten elementuak, mugitzeko funtzioa dutenak, karratu gorriarekin inguratuta daude eta ‘Drop’ osagaia duten elementuak, piezak jarri daitezkeen tokiak, karratu urdinarekin inguratuta daude.



36. IRUDIA: ARIKETAKO DRAG ETA DROP OSAGIAIK

Pieza bat ‘Drop’ osagaia duen tokian jarri arren, programazio arauak jarraituta, agindu konbinaketa batzuk ez dira zuzenak. Horregatik, ‘Drag’ eta ‘Drop’ osagaiak mota ezberdinetan sailkatu dira funtzioaren arabera eta kolore eta forma ezberdinak eman zaizkie. Horrela, pieza bat tokiz aldatzeko, piezaren eta hutsunearen mota berdina izan behar dute, bestela, pieza toki okerrean jarri dela esan nahi du eta hasierako posiziora itzuliko da.

Pieza bat non jarriko den adierazteko, objektu huts bat sortzen da piezaren tamaina berdina duena. Objektu adierazgarri hau, saguaren posizioa kontuan izanda, gertuen dagoen eta ‘Drop’ klasea duen hutsunean jarriko da. Pieza soltatzenean, pieza objektu hutsa dagoen posizioan geratzen da eta objektu hutsa desagertu egiten da. Gainera, piezaren ‘Drag’ eta hutsunearen ‘Drop’ mota erabiliz, objektu adierazgarriak mugimendua zuzena edo okerra den ere jakin dezake. Pieza bat mota okerreko hutsunean jarri nahi bada, objektu adierazgarria gorria izango da eta zuzena bada piezaren kolore berdineko izango da 37. eta 38. irudietan ikus daitezkeen bezala.



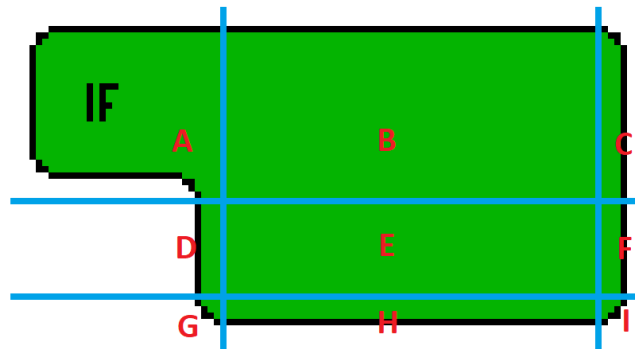
37. IRUDIA: AGINDUA MOTA OKERREKO HUTSUNEAN



38. IRUDIA: AGINDUA MOTA ZUZENENKO HUTSUNEAN

Horretaz gain, hutsuneak dituzten piezak tamaina aldatu behar dute hutsune horietan beste pieza bat jartzen denean. Tamaina aldatetaren kalkulua 'TamainaAldaketa' klaseak egiten du. Altuera, bere burua eta hutsunean dauden pieza guztien (hutsuneko 'seme' guztien) batura da. Zabalera, bere burua eta hutsunean dagoen piezarik zabalena batura da.

Piezen irudia handitzean edo txikitzean forma eta xehetasuna ez galtzeko 9-slice izeneko teknika erabili da. Teknika hau, irudia 9 zatitan banatzen du, 39. irudian ikusten den bezala. Erpinetako zatien tamainak, A, C, G eta I letra dutenak, ez dira inoiz aldatzen. Tamaina aldateta ardatz horizontalean bada, B eta H letrak aldatuko dira, eta aldateta ardatz bertikalean bada, D eta F letrak. Erdiko zatia, E letra duena, tamaina aldateta ardatz horizontalean zein bertikalean denean aldatuko da.



39. IRUDIA: 9-SLICE TEKNIKA PIEZA BATEKIN

Ariketaren emaitza ezagutzeko, 'Emaitzalrakurri' izeneko klasea erabili da. Klase honen parametro nagusia string motako identifikadorea da eta ariketa baten emaitza ariketa leihoan dauden pieza guztien identifikadoreen segida izango da.

Pieza bat beste baten barnean jartzen denean bere 'semea' bihurtzen da. Propietate hori erabilita pieza baten 'semearen' identifikadorea jakin daiteke. Funtzio errekursibo bat erabilita pieza baten identifikadorea eta seme guztien identifikadoreak lortu daitezke (emaitza = identifikadorea + semaren emaitza). Ariketa baten, 'Drag' eta 'Drop' osagaiak duten objektu guztiak script hau daukate, ariketa leihoa barne. Horrela, ariketa baten emaitza ezagutzeko, ariketa leihoari emaitzaren balioa eskatu behar zaio. 40. Irudian emaitza kalkulatzeko funtzioa zelan funtzionatzen duen ikus daiteke.

```
public string EmaiztaItzuli()
{
    emaintza = identifikadorea;
    // semeak baldin baditu
    if (transform.childCount > 0)
    {
        // seme bakoitzeko
        for (int i = 0; i < transform.childCount; i++)
        {
            // emaitza = emaitza + semearen emaitza
            emaintza += transform.GetChild(i).GetComponent<EmaiztaIrakurri>().EmaiztaItzuli();
        }
    }
    return emaintza;
}
```

40. IRUDIA: ARIKETAREN EMAITZA KALKULATZEKO FUNTZIOA

Azkenik, pieza bakoitzak zer egiten duen jakiteko 'PiezaInfo' klasea sortu da. Klasearen atributu nagusia piezaren deskribapena edo informazioa da eta ariketaren argibideak agertzen diren leihoa erabiliko du informazio hori erakusteko. Piezen informazioa erakusteko, Unity-ren *OnPointerEnter()* funtzioa erabili da. Funtzioa sagua objektu baten barruan sartzen denean exekututzen da eta sartu den objektu horrek 'PiezaInfo' klasea badu, argibide leihoan deskribapena jarriko da, bestela, ariketa argibideak jarriko dira. Ariketako pieza guztiak 'PiezaInfo' klasea dute.

41. Irudian ariketa baten adibidea ikusi daiteke. Interfazearen azpialdean ariketaren argibideak jarriko dira. Behin argibideak irakurri ondoren, erabiltzaileak piezak mugitu ditzake eta argibideen tokian leihatila txiki bat geldituko da ariketaren laburpenarekin. Interfazearen ezkerrean ariketa leihoa dago. Leiho horretan dauden piezak erabiltzailearen ariketako erantzuna izango dira. Interfazearen eskumaldean ariketa ebatzteko behar diren piezak daude. Azkenik, interfazeko beheko eta eskumako ertzean, goitik behera, argibideak berriz ikusteko, ariketa berriz hasteko eta ariketa zuzentzeko botoiak daude.



41. IRUDIA: ARIKETA ADIBIDEA

6.4.1 Probak

Ariketak jokoaren funtsa dira, ondorioz funtzio guztien portaera zuzena dela ziurtatu behar da. Horretarako, prototipo honetan egindako probak 23. eta 24. tauletan agertzen dira.

23. TAULA: DRAG AND DROP PROBEN EMAITZAK

PROBA	ESPERO DEN EMAITZA	LORTUTAKO EMAITZA
Pieza mugitu	Saguarekin klik eginez objektua pantailatik mugitu daiteke.	OK
Piezak mugitu eta berriz hasi	Pieza guztiak hasierako posiziora itzultzen dira.	OK
Pieza 'drop' eremuan jarri	Pieza adierazitako 'drop' eremuan jartzen da, piezak ordenaturik daude eta ez daude bata bestearen gainean.	OK
Pieza 'drop' eremutik kanpo jarri	Pieza hasieran zegoen tokira itzultzen da.	OK
Pieza libre dagoen hutsune bakarreko tokian jarri	Pieza hutsunean jartzen da.	ERROR1
Pieza okupaturik dagoen hutsune bakarreko tokian jarri	Piezak tokiz aldatzen dira.	ERROR2
Pieza mota okerreko hutsunean jarri	Pieza aurretik zegoen tokira itzultzen da.	OK
Pieza mota okerreko eta libre dagoen hutsune bakarreko tokian jarri	Pieza aurretik zegoen tokira itzultzen da.	OK
Pieza mota okerreko eta okupaturik dagoen hutsune bakarreko tokian jarri	Hutsunean zegoen pieza geldi mantentzen da eta oker jarri nahi den pieza aurreko tokira itzultzen da.	ERROR3
Ariketaren emaitza irakurri ¹⁰	Emaitza ariketako pieza guztien identifikadore sekuentzia da, agerpen ordenan jarrita.	OK
Sagua pieza baten gainean jarri	Leihatilan ariketaren argibideak kendu eta piezan agertzen den aginduaren informazioa agertzen da.	OK
Sagua pieza baten gainetik kendu	Leihatilan piezaren informazioa kendu eta ariketaren argibideak agertzen dira.	OK
Sagua barneko pieza batetik kanpoko pieza batera mugitu	Leihatilan barneko piezako infirmazioa kendu eta kanpoko piezaren informazioa jartzen da.	ERROR4

¹⁰ Funtzioa zuzena dela ziurtatzeko antzeko test asko egin dira eta proba denak ondo irten dira.

- ERROR1: Hutsune bakarreko tokian piezarik dagoen konprobatzean objektu adierazgarria aurkitzen da. Pieza objektu adierazgarriarekin aldatzen da eta bigarren hau ez da inoiz suntsitzen.
Arazoa konpontzeko, pieza bakarreko hutsunean posizio aldaketa egin baino lehen, aurkitzen duen objektua 'Drag' klasea izatea konprobatuko da. Sortzen den objektua hutsik dagoenez ez dauka klase hau eta ez da posizio aldaketa egiten. Mugimendua amaitzean objektu hutsa desagertu egiten da.
- ERROR2: Hutsunean pieza bat dago, baina lehenengo errorearen soluzioa jarraituz, 'Drag' klasea duen konprobatzean objektu hutsa aurkitzen da. Ondorioz, hutsune bakarreko tokian bi pieza bukatzen dute.
Problema konpontzeko, hutsunean bi objektu badaude (pieza bat eta objektu hutsa), piezak aldatu egiten dira eta objektu adierazgarria suntsitu egiten da.
- ERROR3: Naiz eta pieza toki okerrean jartzen saiatu, bi piezak, hutsunean zegoen pieza eta erabiltzaileak mugitu duena hasierako posizioetara mugitzen dira.
Arazo hau oso erraz konpontzen da, baldintzen ordena aldatuz: lehenengo pieza berria jartzen da, gero, lehengo zatia posible bada, bazegoen pieza mugituko da.
- ERROR4: Sagua zein piezaren gainean dagoen jakiteko Unity-ko *OnPointerEnter()* funtzioa soilik erabili denez, pieza batetik irtetea ez da ezer gertatzen.
Arazoa ekiditeko, *OnPointerExit()* funtzioa erabili da. Sagua pieza baten gainetik kentzen denean funtzioa exekutatu da. Funtzioan, saguaren azpian dauden objektu guztiak zerrendatuko ditu. Zerrendatik altuen dagoen 'PiezaInfo' klasea bilatuko da eta horren informazioa jarriko da leihoan. 'PiezaInfo' klasea aurkitzen ez bada, ariketaren argibideak jarriko dira berriro.

Akatsak konpondu ondoren probak errepikatu dira. [24](#). Taulan errepikatutako proben emaitzak ikusten dira.

24. TAULA: DRAG AND DROP SISTEMAN ERROREA EMAN DUTEN PROBEN ERREPIKAPENA

PROBA	ESPERO DEN EMAITZA	LORTUTAKO EMAITZA
Pieza libre dagoen hutsune bakarreko tokian jarri	Pieza hutsunean jartzen da.	OK
Pieza okupaturik dagoen hutsune bakarreko tokian jarri	Piezak tokiz aldatzen dira.	OK
Pieza mota okerreko eta okupaturik dagoen hutsune bakarreko tokian jarri	Hutsunean zegoen pieza geldi mantentzen da eta oker jarri nahi den pieza aurreko tokira itzultzen da.	OK
Sagua barneko pieza batetik kanpoko pieza batera mugitu	Leihatilan barneko piezako infirmazioa kendu eta kanpoko piezaren informazioa jartzen da.	OK

6.5 Ariketak

Prototipo honetan jokalariak egin ahal dituen ekintzekin (mugitu, salto egin, bihotzak galdu, eraso...) programazio ariketak sortu dira. Gainera, partida gordetzeko eta kargatzeko funtzioak sortu dira eta hasiera menua eta joko gelditzen duen *pause* menua ere implementatu dira.

6.5.1 Ariketa kudeatzailea

Jokalariaren ekintzekin ariketak egiteko, ariketa bakoitzeko pieza guztiak (agindu lerroak), ariketen enuntziatua, pieza bakoitzaren argibideak eta ariketa bakoitzaren emaitza posible denak jarri dira. Ariketa denak prestatu ondoren, zailtasunaren arabera ordenatu dira eta 'Ekintzak' klasea sortu da zeintzuk ariketa ebatzi diren jakiteko. Ondoren, jokalariaren mugimenduak ezgaitu dira eta ebatzitako ariketen arabera mugimenduak ahalbidetzeko funtzionalitatea garatu da.

Ariketa bakoitza interfaze leiho baten egin dira eta 'PuzzleManager' klasea leiho horien guztien 'gurasoa' da. Horrela, klase honek ariketa guztiak eskuragarri ditu zerrenda baten bezala eta hauetako bakoitza jartzeko, kentzeko eta emaitza konprobatzeko gai da.

6.5.2 Gorde/Kargatu funtzioak

Erabiltzaileak bere partidan egiten dituen aurreraketak ondorengo saioan mantentzea nahi da. Horretarako, jokalariaren inbentarioaren egoera, zeharkatutako azken *checkpoint*-a, ariketa zerrenda eta eszenatokian gelditzen diren itemak (jokalaria itemak hartu ondoren desagertu egiten dira) gorde behar dira. Datu horiek serializatu daitekeen klase baten gorde dira. Klasea 42. irudian agertzen da.

```
[Serializable]
public class Data
{
    public int checkpointZenbakia;
    public bool[] ariketak;
    public int txanponKopurua;
    public bool suArgia;
    public bool ezpata;
    public bool arkuak;
    public int geziKopurua, geziKopuruMax;
    public int bizitzaPuntuak, bizitzaPuntuMax;
    public bool[] mapaItemak;
}
```

42. IRUDIA: PARTIDAREN EGOERA GORDETEZKO KLEASEA

Jokoa martxan jartzen denean, fitxategia aurkitzen bada, datuak kargatuko dira eta bestela partida hasieratik jolastu beharko da. Fitxategiko datuak kargatzean, jokalaria 'checkpointZenbakia' *checkpoint*-era mugituko da, 'ariketak' zerrendarekin ebatzi diren ariketak ezgaituko dira (honek aldi berean jokalariaren mugimenduetako batzuk aktibatuko ditu), inbentarioa txanpon, item, gezi eta bihotzekin eguneratuko da eta azkenik, mapan dauden itemak gaitu edo ezgaituko dira 'mapaltemak' zerrendak adierazten duen bezala¹¹.

Datuak gordetzeko, bigarren prototipoan sortutako *checkpoint*-ak erabili dira. Jokalariak hauetako bat zeharkatzen duenean eta 'JokalariKudeatzailea' klaseko 'gordePuntuak' atributuaren balioa eguneratzen badu, partida gordeko da.

6.5.3 Menuak

Aplikazioa zabaltzean zuzenean jokoan ez hasteko eta erabiltzaileari partida hasieratik jolasteko aukera emateko hasierako menua sortu da. Menu hau jolasteko edo irteteko aukerak eta jolastu nahi bada, eta partida bat gordeta badago, hau jarraitzeko edo berriz hasteko aukera eskaintzen du.

Hasiera menuan, datuak gordeta daudenean karga prozesua gertatzen da eta daturik ez badago eta partida berri bada, datuak gordetzeko fitxategia sortzen da. Hauentzako karga denbora txiki bat behar denez, trantsizioa egiteko klasea sortu da. 'FadeManager' klaseak,

¹¹ *Checkpoint*-ekin eta ariketako piezekin egin den bezala, item guztiak guraso berdina dute, ondorioz guztiak zerrenda baten bezala bilduta daude.

pantaila iluntzen du eta dena kargatu ondoren berriz ere argitzen du. Klase hau, joko eta ariketen artean aldaketa egiteko, menuak jarri eta kentzeko, joko barruko gelen artean mugitzeko eta jokalaria hil ondoren *checkpoint*-era mugitzeko erabili da. Horrela kamera mugimenduak edo interfazearen agerpena txukuna eta atsegina izango da.

Behin partida hasita dagoela, jokoa gelditzeko aukera eskaintzen duen *pause* menua garatu da. Menua 'P' edo 'ESC' tekla zapaltzen zabaltzen da. Menuan, partidako denbora gelditzeaz gain, hasierako menura itzultzeko aukera eta ariketen argibideak ikusteko aukera eskaintzen da. Menua ixteko berriz ere 'P' edo 'ESC' tekla zapaldu daiteke edo 'JARRAITU' jartzen duen botoian klik egin daiteke.

Interfaze guztiak garatu ondoren, erresoluzio ezberdinetan ikusgarriak izateko prestatu dira. Unity-k, erresoluzio anitzeko interfazeak izateko egin behar diren aldaketak erakusten ditu bere webgunean [12]. Erresoluzioa aldatzean bi arazo gertatzen dira: interfazeko elementu batzuk pantailatik irteten dira edo bata betearen gainean agertzen dira. Elementuak pantailatik kanpo ez agertzeko, pantailako erdigunera edo erpinetara ainguratu behar dira. Adibidez, bihotz kopurua adierazten duen interfazea goiko eta ezkerreko erpinean ainguratuta dago. Bigarren arazoa konpontzeko, interfazeko osagaiak pantailaren tamainarekin aldatzea aukera jarri behar da. Interfazetik kanpo erresoluzio aldaketekin arazoak saihesteko, Unity-ko dendako Auto-letterbox doako softwarea instalatu da. Honek, eszenatokiaren tamaina aldatu beharrean marra horizontalak edo bertikalak jartzen ditu 'soberan' dagoen pantaila zatia ezkutatzeko.

6.5.4 Probak

Atal honetan ariketen funtzionamendua, partida gorde eta kargatzeko funtzioak eta menuen portaera egokia dela konprobatuko da.

6.5.4.1 Ariketa kudeatzailea

Ariketa guztien funtzionamendua zuzena dela, ariketak trebetasunak gaitzen dituztela eta orokorrean ariketen portaera zuzena dela ziurtatzeko, 25. taulan dauden probak egin dira.

25. TAULA: ARIKETEN PROBEN EMAITZAK

PROBA	ESPERO DEN EMAITZA	LORTUTAKO EMAITZA
Ariketak jarri/kendu	Ariketaren zenbakia erabilia ariketak jarri eta kendu daitezke. Emaiza ariketa zenbakia erabilia konprobatzen da ere.	OK
Ariketa guztiak ebatzi	Pieza denak behar bezala funtzionatzen dute. Ariketa bakoitzeko argibideak agertzen dira. Pieza bakoitzak zer egiten duen ikusi daiteke. Ariketa berriro hasteko botoiak ariketan egindako aldaketak kentzen ditu. Emaiza konprobatzeko botoiak ariketa zuzena den konprobatzen du.	OK
Jokalariaren mugimenduak konprobatu ariketak ebatzi gabe	Jokalariak ezin da mugitu	OK
Jokalariaren mugimenduak konprobatu ariketak ebatzi ondoren	Ariketa bakoitzak aurretik ezgaituta dagoen jokalariaren ekintza bat ahalbidetzen du.	OK
Ariketa bat ebatzi	'Ekintzak' klaseko ariketa zerrenda eguneratu egiten da. Ariketa ebatzi aurretik ezgaituta zegoen jokalariaren mugimendu bat ahalbidetzen da.	OK

6.5.4.2 Gorde/Kargatu funtzioak

Partidaren egoera eta datuak gordetzeko eta berriz kargatzeko funtzioak zuzenak direla konprobatzeko egindako probak 26. taulan agertzen dira.

26. TAULA: PARTIDA GORDE/KARGATU PROBEN EMAITZAK

PROBA	ESPERO DEN EMAITZA	LORTUTAKO EMAITZA
Partida gorde	Partidaren egoera fitxategi baten gordetzen da.	OK
Partida kargatu. Partidaren fitxategia existitzen da	Fitxategia aurkitzen bada datuak kargatzen dira eta jokalariak partida jarraitu dezake gordetako datu berdinekin. Fitxategia ez bada aurkitzen edo zabaldu ezin bada partida berria hasten da.	OK
Partida kargatu. Partidaren fitxategia ez da existitzen	Partida berria hasten da.	OK

6.5.4.3 Menuak

Sortutako menu berriak dituzten aukerak zuzenak direla konprobatzeko, pantaila aldaketak egiteko trantsizio funtzioa zuzen dabilela eta jokoan dauden interfazeak erresoluzio ezberdinetan funtzionatzen dutela konprobatzeko 27. taulan agertzen diren probak egin dira.

27. TAULA: MENUEN PROBEN EMAITZAK

PROBA	ESPERO DEN EMAITZA	LORTUTAKO EMAITZA
Hasiera menuan 'Irten' sakatu	Aplikazioa itxi egiten da.	OK
Hasiera menuan 'Jolastu' sakatu	Partidarik gordeta ez badago partida berria hasten da, bestela partida jarraitzeko edo berriz hasteko aukera eskaintzen da.	OK
Hasiera menuan 'Jarraitu' sakatu	Partida martxan jartzen da. Gordetako datuak kargatzen dira.	OK
Hasiera menuan 'Hasi' sakatu	Partida martxan jartzen da. Partida berria hasten da.	OK
Jokoaren hasiera menuan lehen karga gertatzen da	Hasiera menua ilundu egiten da. Jokoa prest dagoenean pantaila berriz argitzen da.	OK
Jokalaria gelaz aldatzen da	Pantaila ilundu egiten da gelatik irtetean. Gela berrian sartzen dagoenean berriz argitzen da. Gelak bata bestearen ondoan daudela dirudi.	OK
Jokalaria hil egiten da	Jokalariaren hil animazioa ikusi ondoren pantaila ilundu egiten da. Pantaila argitzen denean jokalaria <i>checkpoint</i> -ean dago eta berpiztu animazioa ikusten da.	OK
Ariketa bat hasten da	Pantaila ilundu egiten da eta jokoa gelditu egiten da. Ariketa interfazea jarri ondoren pantaila berriz argitzen da. Pantaila guztiz argituta dagoenean ariketaren argibideak jartzen dira.	OK
Ariketa bukatzen da	Pantaila ilundu egiten da. Interfazea kendu ondoren pantaila argitu egiten da eta jokoa berriz ere martxan jartzen da.	OK
<i>Pause</i> menua jartzen da	Jokoa gelditu egiten da eta menuaren interfazea jartzen da.	OK
<i>Pause</i> menuan jarraitu botoia sakatzen da	Jokoa berriz ere martxan jartzen da eta menuaren interfazea ixten da.	OK
<i>Pause</i> menuan irten botoia sakatzen da	Gorde ez diren datuak galduko direla ohartarazi ondoren pantaila ilundu egiten da. Hasiera menua kargatu ondoren pantaila argitzen da.	OK
<i>Pause</i> menuan hiztegia botoia sakatzen da	Ariketa guztietan emandako argibideak berriro irakurtzeko leihoa zabaltzen da.	OK
Hiztegi leihoan 'P' edo 'ESC' botoia sakatzen da	Hiztegi leihoa eta menua itxi egiten dira. Jokoa martxan jartzen da.	OK

PROBA	ESPERO DEN EMAITZA	LORTUTAKO EMAITZA
Ariketan erresoluzioak aldatu	Interfazeko eduki guztia ikusgarria da.	OK
Hasiera menuan erresoluzioak aldatu	Interfazeko eduki guztia ikusgarria da.	OK
<i>Pause</i> menuan eta hiztegia leihoan erresoluzioak aldatu	Interfazeko eduki guztia ikusgarria da.	OK

6.6 Oztupoak

Implementazioarekin bukatzeko, mapa ariketen ordenaren arabera antolatu da. Jokoari musika eta soinu efektuak ere jarri zaizkio. Jokoaren ingurumen egipitoarra lortzen laguntzeko irudi apaingarriak ere jarri dira.

6.6.1 Ariketa *trigger*-ak

Ariketak jokoaren parte izateko, jokalaria gainditu ezin duen oztupo baten aurrean dagoela ikusi behar du. Eta soilik ariketa egin ondoren eta mugimendu berri bat lortu ondoren izango da oztupoa ebazteko gai.

Mapa eraikitzean bi oztupo mota bereiztu behar dira: jokoaren entretenigarria egiten duten erronkak eta ariketa egitea behartzen duten oztupo garaiezinak.

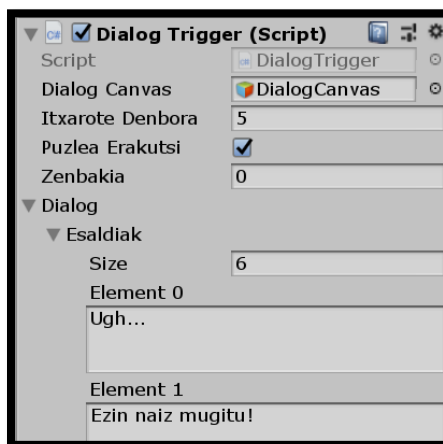
Lehen kasurako, oztupo eta erronken antolakuntza jokalariai eskuragarri dituen trebetasunekin bateragarriak izan behar dute. Jokoan aurrera egin ahala, erronken zailtasuna pixkanaka handituz joango da. Horrela, ariketen konplexutasuna ere areagotuz doanez, jokoaren zailtasun maila orokorra ere handitzen joango da. Plataforma joko bat denez, maparen erronkak diseinatzeko ‘Nintendo metodoa’ jarraitu da: “*Evolve, expand and reset*” [13]. Erronkak bi puntu seguruen artean daude beti eta oztupoa gainditzea ez da bereziki zaila. Aurrera egin ahala, oztupoen zailtasun maila handituz doaz eta erronkak konbinatuz edo puntu seguruen distantziak urrundu.

Bigarren kasuan, oztupoa gainditzea ezinezkoa dela ziurtatu behar da ariketak jartzeko aitzakia egoteko. Oztupo hauek ariketarekin bereganatutako trebetasun berriarekin soilik zeharkatu daitezke. Jokalaria ezinezko erronka batera gerturatzen denean, *trigger* bat eragingo du. Aurreko *trigger*-ak bezala, jokalariai bakarrik eragingo du. Ondoren, oztupoa zergatik ezin duen gainditu azalduko da eta behar duen trebetasuna ahalbidetzen duen ariketa jarriko da. 43. Irudian funtzioa zelakoa den ikus daiteke.

```
private void OnTriggerEnter2D(Collider2D collision)
{
    if(collision.tag == "Player")
    {
        StartDialog(dialog);
        puzzleCanvas.GetComponent<PuzzleManager>().AriketaGaitu(zenbakia);
    }
}
```

43. IRUDIA: ARIKETAK JARTZEKO FUNTZIOA

Prozesu guztia ‘DialogTrigger’ izeneko klaseak kudeatuko du. Klaseak, 43. irudiko kodean ikusten den bezala, 3 datu garrantzitsu behar ditu: oztopoaren azalpena duen mezua (‘dialog’ izena dauka), ariketen ‘gurasoa’ den objektua (‘puzzleCanvas’ izenarekin agertzen da) eta zenbatgarren ‘semea’ edo ariketa gaitu behar den jakiteko zenbakia (‘zenbakia’ izena dauka). Datuak Unity-ren interfazea erabiliz hornitzen dira, 44. irudian ikusten den bezala. Behin ariketa ebatzi dela *trigger*-a ezgaitu egiten da ariketa etengabe ez jartzeko.



44. IRUDIA: ARIKETA BATEN TRIGGER-A

6.6.2 Soinuak

Jokoari azken ukituak emateko, jokalariari, etsaiei, tranpei eta aurkitu daitezkeen beste objektu batzuei soinuak jarri zaizkio¹² [14]. Soinuak kudeatzea oso ataza erraza da: Soinua behar duen objektuari Unity-ko ‘AudioSource’ osagaia jarri behar zaio eta soinua entzuteko *Play()* funtzioari deitzen zaio. Soinuak hiru egoera ezberdinetan jarri dira:

- Ekintzak: Jokoan, gertakizun baten ondorioz (jokalariaren mugimendua, tranpa bat gaitzea, etsaiaren eraso) gertatzen diren soinuak, script bitartez jartzen dira. Adibidez, jokalariak salto egiten duenean soinua jartzeko, *saltoSoinua.Play()* kode lerroa *SaltoEgin()* funtzioaren barruan jarri behar da.

¹² Erabilitako audio pistak CC0 lizentzia daukate

- Instantzia berriak: Jokoan, objektu batzuk instantziatu egiten dira. Soinu efektua momentu horretan gertatzeko, 'AudioSource' osagaiak 'playOnAwake' atributua eskaintzen du. Horrela, soinua duen objektuaren *Awake()* funtzioa deitzen denean soinua jartzen da. Adibidez, jokalaria gezi bat jaurtitzen badu, gezia instantziatzen denean gezi soinua bakarrik jarriko da.
- Musika: Jokoaren musika jartzeko, 'AudioSource' osagaiaren 'loop' eta 'playOnAwake' atributuak gaitu dira. Horrela, musika jokoaren hasieratik eta etengabe entzutea lortu da.

Horretaz gain, *pause* menuan soinuak aktibatu edo desaktibatuzeko botoia jarri da. Botoi horrek, Unity-ko 'AudioListener' osagaia gelditzen du. 'AudioListener' osagaia jokoan dagoen mikrofonoa da. Gertatzen diren soinuak antzeman eta errepikatzen ditu, ondorioz, osagaia ezgaitzean ez da soinurik entzuten.

6.6.3 Probak

Ariketak jartzeko *trigger*-ak eta soinuak behar bezala funtzionatzen dutela konprobatzeko 28. eta 29. tauletan agertzen diren probak burutu dira.

28. TAULA: ARIKETA, TRIGGER-EN ETA SOINUEN PROBEN EMAITZAK

PROBA	ESPERO DEN EMAITZA	LORTUTAKO EMAITZA
Etsai bat <i>trigger</i> barruan sartzen da	Etsaiak ez du <i>trigger</i> -a aktibatzen	OK
Jokalaria oztopora gerturatu eta <i>trigger</i> -a aktibatzen du	Ariketa zergatik jarriko den azaltzen duen mezua agertzen da. Oztopoa gainditzeko behar den ariketa jartzen da. Ariketa ebatzi ondoren <i>trigger</i> -a ezgaitu egiten da.	OK
Jokalaria soinu efektua duen ekintza egiten du	Ekintza egiten duenean soinua entzuten da eta amaitzean / geldi dagoenean ez da entzuten.	OK
Etsai batek soinu efektua duen mugimendua egiten du	Etsaiak mugimendua egiten duenean soinua entzuten da eta geldi dagoenean ez da entzuten.	ERROR1
Soinu efektua duen soinua instantziatzen da	Objektu berria agertzean dagokion soinua entzuten da.	OK
Partida bat hasten da	Jokoaren musika etengabe entzuten da.	OK
<i>Pause</i> menuko bozgorailu botoia sakatzen da	Soinuak aktibatuta badaude desaktibatu egiten dira eta soinuak desaktibatuta badaude aktibatu egiten dira.	OK

- ERROR1: Etsaien soinu efektua zuzena da. Arazoa ondorengoa da: Etsaiak jokalaria gela baten sartzen denean aktibatzen dira. Gelak handiak direnez, etsai batzuk ez dira pantailan agertzen, baina etsai guztien soinu efektuak entzuten dira. Funtzionamendu ezegoki hau konpontzeko, etsai, tranpa eta soinua duten beste objektuei *trigger* bat jarri zaie. *Trigger* hau jokalaria aktibatzen du soilik eta soinu efektua entzuten den distantzia adierazten du, horrela, soinua soilik entzungo da jokalaria beste objektura gerturatzen denean.

Errorea konpondu ondoren proba errepikatu da. 29. Taulan errepikatutako probaren emaitza ikusten da.

29. TAULA: ARIKETA, TRIGGER ETA SOINUAN ERROREA EMAN DUEN PROBAREN ERREPIKAPENA

PROBA	ESPERO DEN EMAITZA	LORTUTAKO EMAITZA
Etsai batek soinu efektua duen mugimendua egiten du	Etsaiak mugimendua egiten duenean soinua entzuten da eta geldi dagoenean ez da entzuten.	OK

7 Ondorioak

Atal honetan garatutako lana aztertuko da: helburuen betekizunak, garapen arazoak, egin daitezkeen hobekuntzak eta baita hausnarketa pertsonala ere.

7.1 Helburuen betekizuna

Atal honetan proiektu hasieran proposatutako helburuak bete diren azalduko da:

- *Arazorik gabe funtzionatzen duen 2 dimentsioko plataforma joko funtzionala garatzea. Jokoa oztopoz betetako gelaz osatuta egotea eta oztopoak gainditzeko trebetasun ezberdinak (salto, makurtu, korrika egin, irristatu...) garatu.*
Hieroglifoak jokoari ariketa funtzionalitateak kentzen bazaizkio 2 dimentsioko plataforma joko klasiko bat da, hortaz, helburua bete egin da.
- *Programazio ariketak egiteko interfazea garatzea, kode lerroak Drag and Drop sistema erabilia mugitzea eta ariketaren emaitza ezberdinak irakurtzeko funtzionalitatea izatea.*
Jokoan dauden ariketetan, kode lerroak Drag and Drop funtzioa duten piezetan banatuta daude eta sistemak sortu daitezkeen konbinaketa desberdinak bereizten dituzenez helburua bete da.
- *Bideo-jokoak eta programazioak duten harreman estua aprobetxatuz programazio irakasgaiarekin lagungarriak izan daitezkeen ariketak garatzea, ariketen emaitzak jokoan eragina dutelarik.*
Ariketak jokoaren funtzioak programatzen gaudela emateko prestatuta daude. Ariketen zailtasuna oso erraza da eta soilik bukaeran, elementu asko batzen direnean, dira konplexuak ariketak, ondorioz, helburua bete dela esan daiteke.
- *Jokoan aurkitzen diren ariketak programazioko oinarritzko osagaien inguruan (if/else egiturak eta baldintzak, aldagaiak, eragigaiak, funtzioak, begiztak eta for egiturak) informazioa eta azalpenak eskaintzea.*
Ariketak egin aurretik, erabiltzaileari beharrezkoak dituen argibideak erakusten zaizkio. Gainera, beste ariketa batzuetako argibideak ikusteko aukera dauka, hortaz, helburua bete egin da.
- *Joko bat garatzeko beharrezkoak diren ezagutzak bereganatzea. Jokoa garatzeko erraminta (unity) zeintzuk baliabide eskaintzen dituen eta hauek erabiltzen ikasi: fisika legeak jokoan aplikatu, animazio grafoak garatu, soinuak zelan jarri...*
Helburu hau subjektiboa den arren, bete egin dela esan dezaket, proiektua burutzeko jokoekin lotutako teknika ugari ikasi behar izan ditudalako. Lortutako jokoak hasierako ideiarekin antza handia dauka eta orotarako emaitzarekin pozik nago.

- *Bideo-jokoa ikasgela baten aurkitu daitekeen ordenagailu batek erabili ahal izatea. Funtzionatzeko hardware baliabide gutxi erabili behar du eta erresoluzio grafiko gehienetan funtzionatu.*

 Ikasgela baten probatu ez dudan arren, beste gailu pare batekin probatu dut jokoa eta hasierako karga luzea den arren, jokoa funtzionamendua egokia da, ondorioz, helburua bete dela esan daiteke.

7.2 Planteamendu aldaketak

Proiektuaren hasieran, Denbora-plangintza atalean, proiektuaren iraupenaren estimazioa egin zen. Behin proiektua bukatuta dagoela, estimazioaren eta benetako iraupenaren arteko konparaketa egin daiteke. 30. taulan prozesu bakoitzaren estimazioa eta benetako iraupena ikus daitezke:

30. TAULA: PROIEKTUAREN BENETAKO IRAUPENA

PROZESUA	ESTIMAZIOA	IRAUPENA
KUDEAKETA	15 ORDU	14 ORDU
Proiektuaren helburuak erabaki	3 ordu	2 ordu
Denbora-plangintza	2 ordu	3 ordu
Bideragarritasun ekonomikoa	3 ordu	3 ordu
Arriskuen kudeaketa	4 ordu	3 ordu
Proiektuaren garapenerako erramintak aukeratu	3 ordu	3 ordu
IKERKETA/IKASKUNTZA	92 ORDU	105 ORDU
Erraminten instalakuntza	3 ordu	2 ordu
Erraminten ezaugarrien ikaskuntza	55 ordu	63 ordu
Helburuak betetzeko metodo ezberdinak aztertu	34 ordu	40 ordu
BETEKIZUNEN BILKETA	18 ORDU	22 ORDU
Funtzionalitateak zehaztu	4 ordu	6 ordu
Erabilpen kasuak finkatu	12 ordu	13 ordu
Domeinuen ereduak zehaztu	2 ordu	3 ordu
ANALISIA ETA DISEINUA	38 ORDU	40 ORDU
Klase diagrama	16 ordu	16 ordu
Sekuentzia diagramak	22 ordu	24 ordu
INPLEMENTAZIOA	320 ORDU	347 ORDU
Jokalari mugimendua	72 ordu	80 ordu
Mapa eta bizitzak	68 ordu	60 ordu
Eraso sistema	34 ordu	38 ordu
<i>Drag and Drop</i> funtzionalitateak	44 ordu	54 ordu
Ariketak	50 ordu	60 ordu
Oztopoak	52 ordu	55 ordu
GUZTIRA	483 ORDU	528 ORDU

Emaitzaren analisiarekin hasi baino lehen, proiektuarekin zerikusia ez duten baina honen garapenean eragindako oztopoak azalduko dira. Motibazio falta eta arazo pertsonalak direla eta, proiektuaren ekoizpena murriztuz joan da eta azkenik alde batera utzita gelditu da hilabete

batzuetan. Ondorioz, berriro proiektuan lan egitean denbora gehiago behar izan da dena eguneratuta izateko.

Lanaren denbora desbiderapenari dagokionez, 30. taulan argi ikusten da arazo nagusia ikerketa eta inplementazio faseetan gertatu dela. Arretaz begiraturaz gero, denbora desbiderapena hiru prototipoetan izandako arazoaren erruz dela ikusi daiteke. Izan ere, inplementazioan izandako arazoak ikerketa eta ikaskuntza fasera hedatu dira. Hauek dira inplementazio fasean agertutako arazoak eta ondorioak:

- **Jokalariaren mugimendua:** Jokalariaren mugimenduaren bukaerako prototipoa izan baino lehen beste prototipo bat egon da. Hau asko aurreratu ondoren, mugimendu batzuk garatzea oso zaila edo ezinezkoak zirela eta animazioak egiteko arazo asko azaldu ziren. Ondorioz, ikerketa sakonagoa egin behar izan da emaitza zuzena aurkitzeko eta garapenean egindako aurrerakuntza asko galdu egin ziren.
- **Drag eta Drop funtzioak.** Ariketak egiteko drag eta drop sistema garatzea oso erraza izan da, baina piezak kode lerro bezala funtzionatzeko, sortutako arauak eta piezen tamaina aldatzeko modua ez zen denbora plangintzan aurreikusi eta ondorioz garapena uste baino luzeagoa izan da.
- **Ariketak prestatu.** Behin ariketa sistemaren funtzionamendua egokia izanda eta jokalariaren zein ekintza ariketa bihurtuko ziren pentsatu ondoren, ekintza bakoitzeko ariketak sortu ziren. Ariketa bakoitza prestatzeko ordea, uste baino denbora gehiago behar izan da. Ariketa bakoitzak bere azalpena eta enuntziatua behar du, ariketako pieza bakoitza sortu egin behar da, azalpena jarri, balio bat eman eta azkenik ariketaren emaitzak gorde. Horrek, ariketa guztiak prestatzeko uste baino denbora gehiago behar izatea lortu du.

Azkenean, gertatutako ezusteekin, proiektuaren denbora desbiderapena %8,5-ekoa izan da.

7.3 Kodearen kudeaketa

Proiektuaren kode guztia egunean izateko eta bi gailurekin lan egiteko **GIT** sistema erabili da. Hieroglifoak jokoaren kodea publikoa da eta Github web orrian aurkitu daiteke edonor jaitzi, konpilatu eta erailtzeko edo eraldatzeko. Proiektua ondorengo estekan aurkitu daiteke:

<https://github.com/Uribarri95/Hieroglifikoak>

7.4 Zer ikasi dut

Hasteko, joko bat garatzeko jarraitu behar diren pausoak zeintzuk diren ikasi dut. Jokoaren ideiatik, lanaren deskonposaketara eta oinarriko funtziotik konplexuenera joateko egin beharreko bidea.

Unity softwarea zelan erabiltzen den eta atzetik dauden prozesuak zelan funtzionatzen duten ulertu. Softwareak eskaintzen dituen erraminta asko zelan funtzionatzen duten eta osagai garrantzitsuenak zelan funtzionatzen duten eta beste osagai batzuekin zelan elkar eragiten duten ikasi dut.

Proiektua C# lengoaiarekin garatuta dago eta ondorioz, honen funtzionamendua, dituen ezaugarri onak eta ezaugarri ez hain onak ezagutu ditut.

Lan egiteko metodologiari dagokionez ere asko ikasi dut. Ezezagunak diren elementuekin lehen aldiz ibili aurretik, hobe da ikasketa prozesuari behar beste denbora eskaintzea. Horrela, garapen prozesuan aurkitzen diren zailtasunak bizkorrago konpondu daitezke eta arazoaren zergatia edo oinarria ezagutu.

7.5 Zer hobetu dezaket

Proiektuaren helburuak bete diren arren, beti dago hobekuntzak egiteko aukera. Etorkizunean inplementatu daitezkeen hobekuntzak ondorengoak dira:

- **Hasierako kargaren pisua murriztea.** Jokoa eszenatoki gutxitan banatuta dago, ondorioz, karga egin behar denean denbora gehiago behar da. Denbora murrizteko, erabiltzen ez diren baliabideak ezgaitu daitezke edo jokoa eszenatoki gehiagotan banatu daitezke.
- **Istorio bat kontatu.** Jokoa entretenigarria iruditzen zaidan arren, jokoak istorio bat kontatzea gustatuko litzaidake. Horrela, istorioaren hasiera, korapiloa eta bukaera jokoaren betekizun ehunekoarekin lotuta egongo litzateke eta istorio on bat jokalaria arreta bereganatzeko ahalmena dauka.
- **Ariketaren emaitzaren simulazioa.** Ariketa egiteko interfazean, emaitza konprobatzeko botoia sakatzean, ariketako kodearen exekuzioa simulatzen duen errepresentazio grafikoa jartzea. Horrela, ariketaren emaitza okerra bada, erabiltzaileak akatsa non dagoen erras ikusi dezake.
- **Soinu aukera gehiago.** Jokoa dauden soinuak gaituta edo ezgaituta daude, ez dago bolumena aldatzeko aukerarik eta gertutasunaren arabera bolumen aldaketa edo estereo efekturik. Aukera hauek inplementatzea jokoa esperientzia hobetu dezakete.

- **Ariketak sortzeko funtzioa hobetu.** Ariketak sortzeko metodoarekin ariketako pieza bakoitza banaka sortu, piezari balio bat eman eta argibidea jarri, ariketaren enuntziatua jarri eta ariketaren emaitza posibleak jarri behar dira. Guzti hau pixka bat automatizatu daiteke, adibidez, fitxategi baten ariketaren emaitzak jarri eta ordenagailuak behar diren piezak sortzeko metodoa inplementatu daiteke. Horrela, ariketak egiteko sistema jokoan eragina duen arren, funtzionalitatea jokatik banandu daiteke eta ariketak egiteko erraminta baten bihurtu daiteke edo beste joko batekin erabili daiteke.

8 Bibliografía

- [1] Ministerio de Educación, Cultura y Deporte, «Proyectos Atenea y Mercurio. Programa de Nuevas Tecnologías de la Información y de la Comunicación (P.N.T.I.C.) - Publicaciones - Ministerio de Educación, Cultura y Deporte»:
<https://sede.educacion.gob.es/publiventa/proyectos-atenea-y-mercurio-programa-de-nuevas-tecnologias-de-la-informacion-y-de-la-comunicacion-pntic/tecnologias-de-la-informacion-y-de-las-comunicaciones/21621>. [Atzitze-data: 28 10 2019].
- [2] F. Moreno, «La programación: ¿una nueva asignatura? - Innovando en Educación»:
<http://www.innovandoeneducacion.es/la-programacion-una-nueva-asignatura/>. [Atzitze-data: 28 10 2019].
- [3] European Schoolnet, «Press Release: Results From Triseum's Year-Long Game-Based Learning Validation Study Reveal Strong Student Engagement, Motivation and Knowledge Acquisition»:
<http://www.eun.org/news/detail?articleId=1966440>. [Atzitze-data: 29 10 2019].
- [4] eConfidence, «eConfidence in Behaviour Changes through Serious Games»:
<http://www.econfidence.eu/documents/407765/497360/eConfidence-games-booklet-WEB.pdf/dea0d659-c776-4989-82ea-f490ef1e3da3>. [Atzitze-data: 29 10 2019].
- [5] European Comision, «European Commission - PRESS RELEASES - Press release - Vassiliou urges Education Ministers to help kids crack the code,»:
https://europa.eu/rapid/press-release_IP-14-887_en.htm. [Atzitze-data: 30 10 2019].
- [6] N. KROES eta A. VASSILIOU, «LetterbyNeelieKroesandAndroullaVassiliou»:
ec.europa.eu/information_society/newsroom/cf/dae/document.cfm?doc_id=6597. [Atzitze-data: 28 10 2019].
- [7] J. J. Velasco, «Niños programadores: para qué sirve la enseñanza de programación en las escuelas»:
https://www.eldiario.es/turing/Ninos-programadores-ensenanza-programacion-escuelas_0_293970921.html. [Atzitze-data: 29 10 2019].
- [8] Gamelearn Team, «Serious games para la formación: 8 beneficios que te sorprenderán»:
<https://www.game-learn.com/serious-games-formacion-8-beneficios-te-sorprenderan/>. [Atzitze-data: 29 10 2019].
- [9] M. d. e. y. s. social, «Disposición 3156 del BOE núm. 57»:
<https://www.boe.es/boe/dias/2018/03/06/pdfs/BOE-A-2018-3156.pdf>. [Atzitze-data: 14 01 2020].
- [10] S. Lague, «2D Platformer Controller (Unity5)»:
https://www.youtube.com/playlist?list=PLFt_AvWsXI0f0hqURLhyloAabKPGRsqjz. [Atzitze-data: 28 10 2019].

- [11] C. Jamal, «Unity Card Game»:
<https://www.youtube.com/playlist?list=PLbghT7MmckI42Gkp2cILkO2nRxK2M4NLo>.
[Atzitze-data: 28 10 2019].
- [12] Unity Technologies, «Unity - Manual: Designing UI for Multiple Resolutions»:
<https://docs.unity3d.com/Manual/HOWTO-UIMultiResolution.html>. [Atzitze-data: 29 10
2019].
- [13] P. Holleman, «How to Design Levels With the "Super Mario World Method"»:
<https://gamedevelopment.tutsplus.com/articles/how-to-design-levels-with-the-super-mario-world-method--cms-25177>. [Atzitze-data: 28 10 2019].
- [14] FreeSound, «Freesound - Freesound»: <https://freesound.org/>. [Atzitze-data: 28 11
2019].

9 Eranskinak

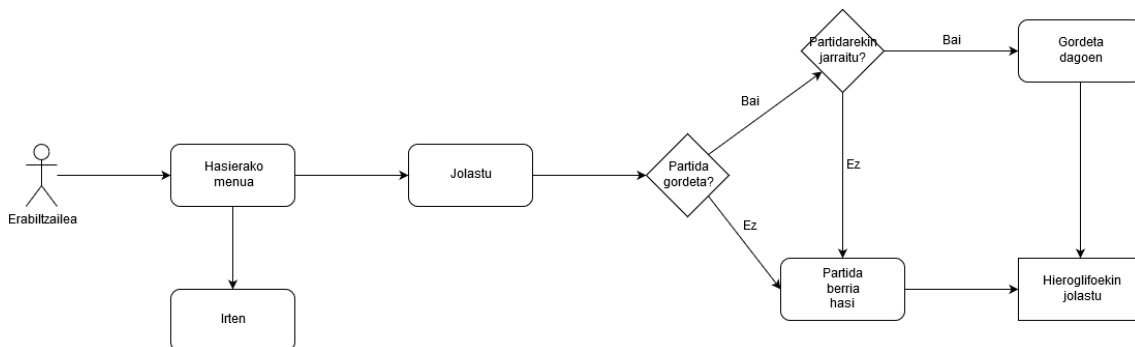
Atal honetan proiektuaren informazio osagarria eskainiko da. Erabilpen kasuak sakonago aztertuko dira eta jokoaren funtzionamendua azalduko da sekuentzia diagramen bitartez.

9.1 Erabilpen kasuen azterketa sakona

Eranskin honetan erabilpen kasuak banaka aztertuko dira. Erabilpen kasu bakoitza zehaztasunez deskribatuko da, bere fluxu-diagrama izango du eta kasuan parte hartzen duten interfazeak aurkeztuko dira.

9.1.1 Hasierako menua

Jokoa zabaltzerakoan, hasierako menua agertzen da. 45. Irudian hasierako erabilpen kasuaren adierazpen grafikoa ikus daiteke. Hasteko, jolasteko edo irteteko aukera eskaintzen da. Irten botoian sakatzen bada jokia itxi egiten da, aplikazio baten goiko x gorria zapaltzean gertatzen den bezala. Erabiltzaileak jolastu botoia sakatzen badu, bi gauza gertatu daiteke. Lehenengo kasuan, partidarik gordeta ez badago, partida berria hasiko da. Bigarren kasuan, partida bat gordeta badago, jokalaria partida jarraitzeko edo partida berri bat hasteko aukera emango zaio.



45. IRUDIA: HASIERAKO MENUA FLUXU DIAGRAMA

Naiz eta diagrama oso luzea dirudien, benetan bi pantaila besterik ez dira. Lehena, jolasteko edo irteteko aukera ematen duena. Bigarrena soilik partida bat gordeta badago agertzen da eta berriz hasteko edo jarraitzeko aukera eskaintzen du. Bi pantailak duten itxura 46. eta 47. irudietan ikusten da.



46. IRUDIA: JOLASTU/IRTEN PANTAILA



47. IRUDIA: JARRAITU/HASI PANTAILA

9.1.2 Ariketak ebazteko interfazea

Erabiltzaileak ariketak egin baino lehen argibideak irakurri behar ditu. Instrukzioak irakurri ondoren ariketetan lan egiteko kontrola izango du. 48. Irudian ikus daitezenez, interfazea 3 zatitan bananduta dago:



48. IRUDIA: ARIKETA PANTAILAREN ADIBIDEA

- Ariketak egiteko leihatila (karratu berdea). Hemen ariketak burutzeko aginduen sekuentzia dago. Hasieratik ariketa leihoan jarrita dauden aginduak normalean zuzen jarrita daude, horrela jokalariek ariketa ebazteko ariketaren zati bat eginda aurkituko du.
- Aginduen leihatila (karratu urdina). Hemen ariketa burutzeko aginduak daude. Leihatila batetik bestera aginduak mugitzeko piezak bultzatu besterik ez da egin behar.

- Argibide leihoa (karratu horia) eta botoiak. Instrukzioak irakurri ondoren ariketa ebazteko deskribapen txiki bat geratzen da argibide leihoan. Sagua kode lerro baten gainean jartzen bada, argibideak kendu eta pieza horren deskribapena agertzen da. Sagua piezarik ukitzen ez dagoenean ariketaren deskribapena agertuko da berriz.

Ariketa interfazean 3 botoi daude:

- Bonbila duen **botoi urdina** hasieran irakurritako argibide denak berriz ere ikusteko balio du.
- Berriro hasi jartzen duen **botoi horia** aginduak hasierako egoeran zeuden moduan uzten ditu.
- Badago! Jartzen duen **botoi berdea** ariketa zuzentzeko balio du. Ariketa zuzena bada, horrela adierazten duen soinu bat entzungo da eta jokora itzuli edo ondorengo ariketara joango da jokalaria. Ariketa okerra bada beste soinu bat entzungo da eta ariketa errepatatzea eta berriz saiatzea eskatuko zaio jokalariari.

Jokalariak edozein momentuan *pause* menua jarri dezake 'ESC' edo 'P' tekla erabilita. Menutik irtetean ariketa zegoen bezala egongo da.

9.1.3 *Pause* menua

Behin erabiltzaileak jokoa hasi duela, edozein momentuan *pause* menua jarri dezake. Menua jartzeko nahikoa da 'ESC' edo 'P' tekla sakatzea eta jokoa gelditu egiten da. 49. Irudian *pause* menuak duen itxura ikusten da. Menu honetan erabiltzaileak 4 botoi aurkituko ditu:

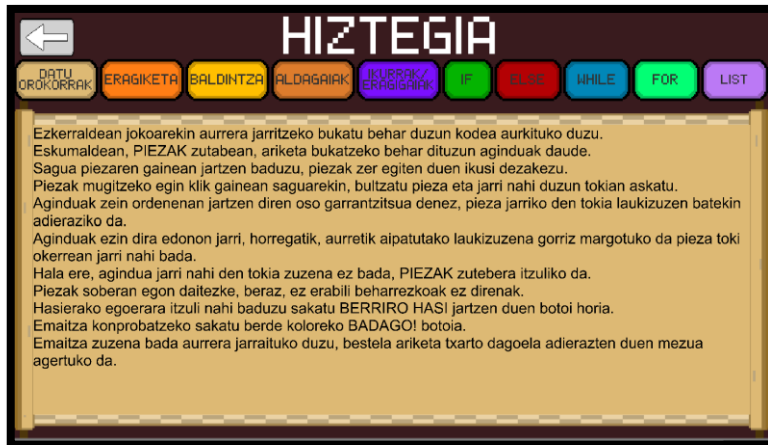


49. IRUDIA: PAUSE MENUA

- **Jarraitu botoia.** Jarraitu botoia (edo 'ESC' edo 'P' teklak) sakatuz jokoa utzitako tokian jarraituko du. Jokalaria ariketak egiten bazegoen ariketak egitera itzultzen da eta jokoa bazegoen jolasten jarraitzera itzultzen da, 5. irudian adierazten den bezala.
- **Soinua gaitu eta ezgaitzeko botoia** (bozgorailua duen botoi borobila). Bozgorailuan klik eginez soinua jarri edo kenduko da. Soinua piztuta dagoenean bozgorailua uhinekin

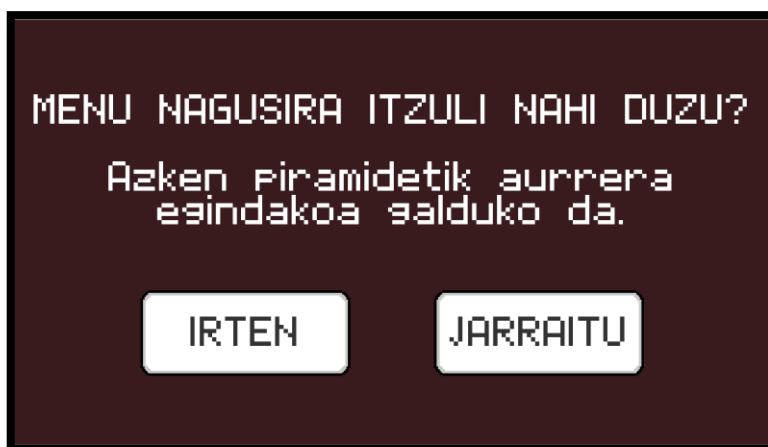
agertzen da eta itzalita dagoenean uhin gabe eta lerro gorri batez gurutzaturik (gainera musika eta botoien soinuak ez dira entzungo).

- **Hiztegia botoia.** Pieza mota bakoitzak zelan funtzionatzen duen ikusteko hiztegi txiki bat agertzen da. 50. Irudian hiztegia leihoak duen itxura ikusten da. Goian agertzen diren botoiak ariketetan duten izen eta kolore berdina dute. Botoietako bakoitzak sakatzean, dagokion informazioa erakusten du pantailan. Atzera geziarekin *pause* menura itzultzen da.



50. IRUDIA: HIZTEGIA PANTAILA

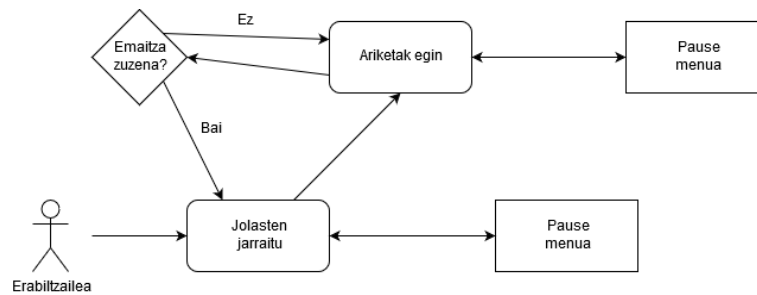
- **Menu botoia.** Hasierako menura itzultzeko balio du. Irten aurretik jokoan gorde ez diren datuak galduko direla ohartarazten da. 51. Irudian pantailak duen itxura ikusten da. Horrela, irten aurretik atzera egin ahal da eta erabiltzaileak partida jarraitzeko aukera du.



51. IRUDIA: MENUA NAGUSIRA ITZULTZEKO PANTAILA

9.1.4 Jokoaren portaera orokorra

Esan bezala, joko lineala da, hau da, partidaren hasieratik bukaera arte bide bakarra dago, eta bidean aurrera egiteko dauden oztopoak derrigorrez gainditu behar dira. Adibidez, erabiltzaileak hasieran ezin du jokalaria mugitu. Mugimendua jartzeko ariketa agertuko da pantailan, eta ariketa ebatztean jokalaria mugitu ahal izango du. Aurrerago, amildegi batekin aurkituko da. Amildegia ginetik salto egiteko beste ariketa bat ebatzi beharko du eta ariketa ebatzita dagoenean, amildegi ginetik salto egiteko ahalmena izango du eta abar. Horrela, jokalaria aurrera doan heinean oztopo berriak aurkituko ditu eta oztopoa gainditzeko ariketa egin ondoren oztopoa zeharkatuko du beste arazo batekin topatu arte. 52. Irudian jokoaren portaera erakusten duen fluxu diagrama ikus daiteke.



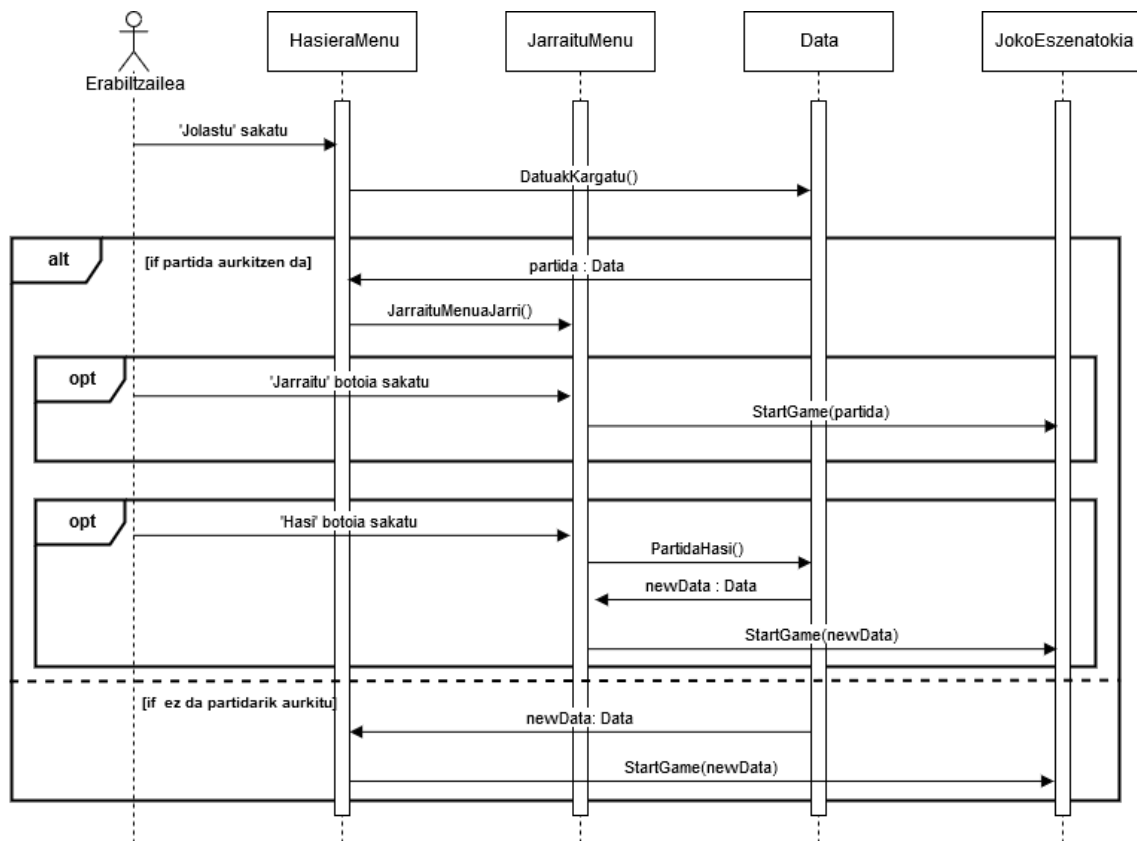
52. IRUDIA: JOKOAREN PORTAERA OROKORRA FLUXU DIAGRAMA

9.2 Sekuentzia diagramak

Eranskin honetan, Hieroglifoak jokoaren funtzio nagusien sekuentzia diagramak aztertuko dira.

9.2.1 Partida hasi

Partida hasteko diagrama sekuentzia:

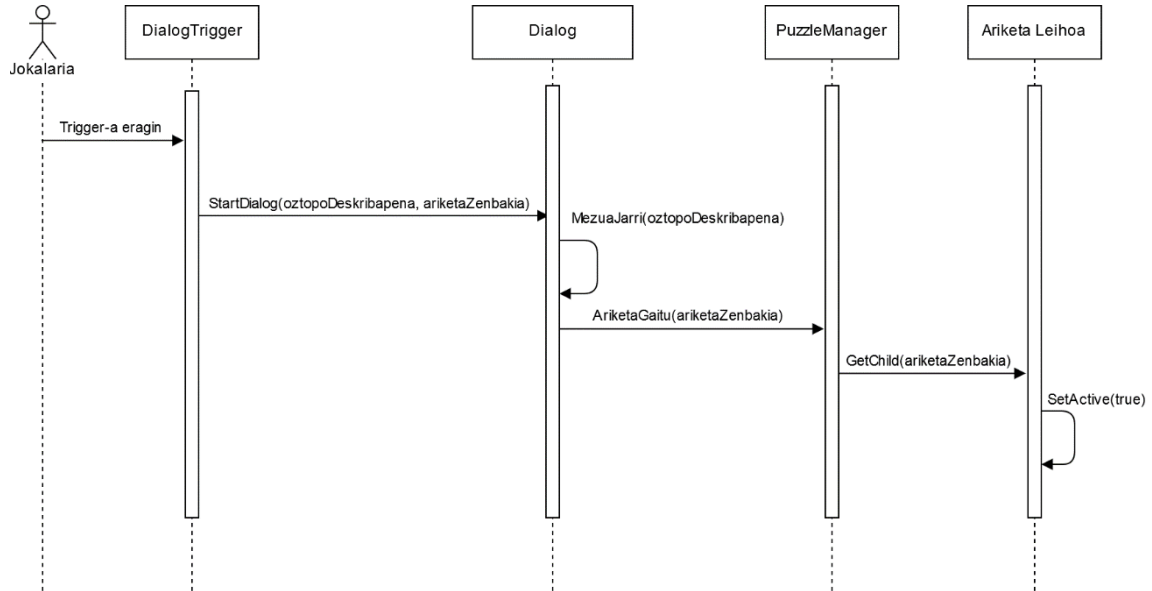


53. IRUDIA: PARTIDA HASTEKO SEKUENTZIA DIAGRAMA

Diagraman partida bat hasteko prozesua ikusten da. Partida bat gordeta dagoenean partidako datuak erabili daitezke jokia hasteko. Bestela, datu berriak sortzen dira eta partida datu horiekin hasten da.

9.2.2 Ariketa jarri

Ariketa jartzeko sekuentzia diagrama:

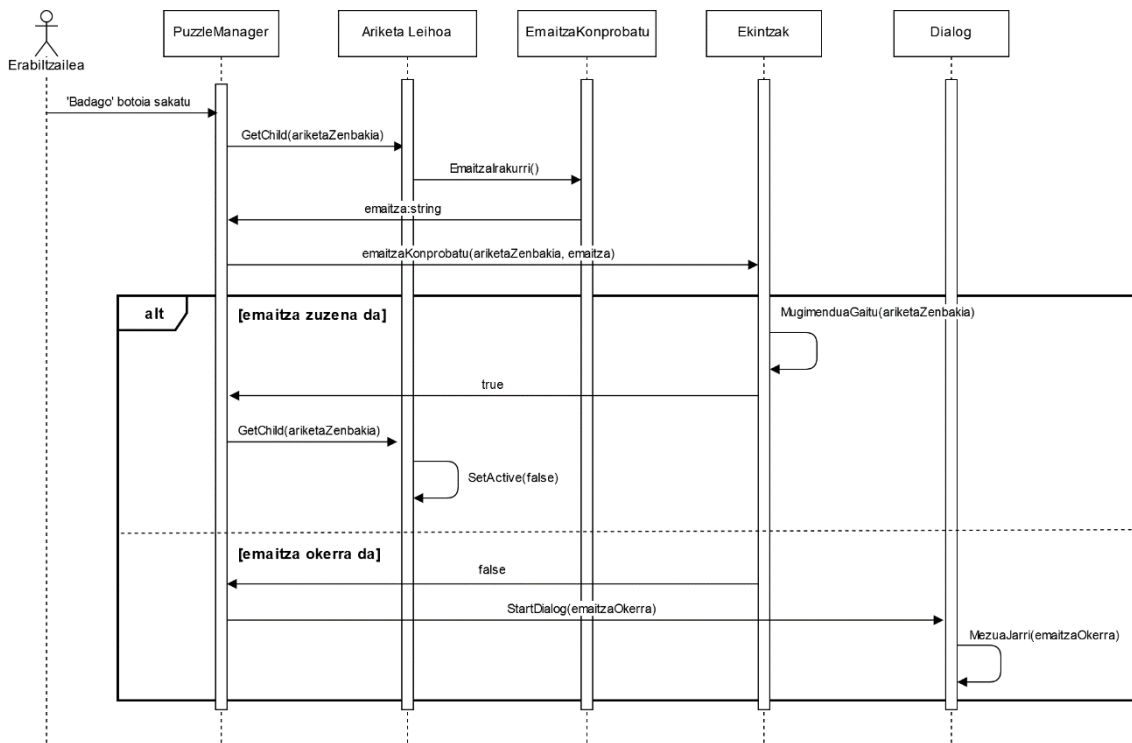


54. IRUDIA: ARIKETA JARTZEKO SEKUENTZIA DIAGRAMA

Diagraman ariketa bat zelan jartzen den ikusten da. Trigger-ak jarri behar den ariketa zenbakia dauka gordeta. Zenbaki horrekin, 'PuzzleManager' klaseak adierazitako 'semea' aktibatzen du.

9.2.3 Ariketa emaitza konprobatu

Ariketa baten emaitza konprobatzeko sekuentzia diagrama:

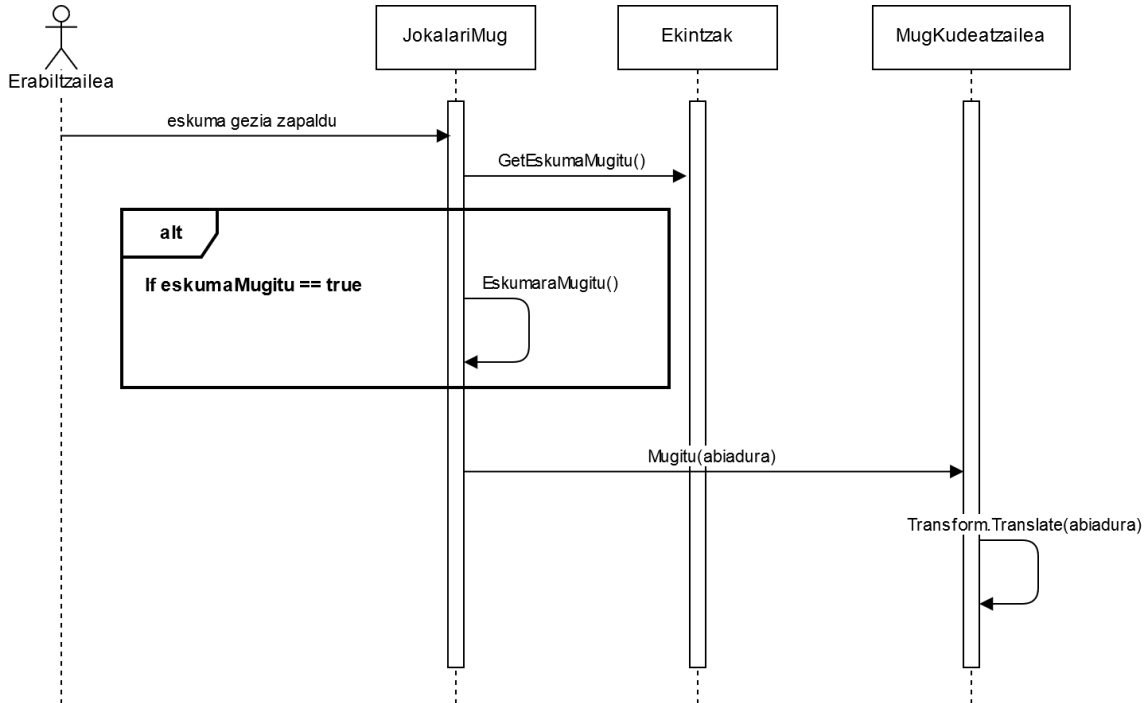


55. IRUDIA: ARIKETAREN EMAITZA KONPROBATZEKO SEKUENTZIA DIAGRAMA

Diagraman ikusten denez, ariketa zuzena denean, ariketari lotutako jokalariaren mugimendua aktibatu egiten da eta ariketa ezgaitu egiten da. Ariketa okerra denean mezu adierazgarri bat agertzen da pantailan.

9.2.4 Mugitu

Jokalaria ekintza bat egiteko (kasu honetan eskumara mugitu) sekuentzia diagrama:



56. IRUDIA: JOKALARIA ESKUMARA MUGITZEKO SEKUENTZIA DIAGRAMA

Diagraman jokalaria eskumara mugitzeko jarraitzen diren pausoak ageri dira. Ekintzak klaseak eskumara mugitzeko funtzioa ahalbidetzen badu, jokalariaren abiadura horizontala eraldatuko da. Bestela, jokalaria ez da eskumara mugituko, baina agian beste ekintza bat burutu dezake (salto egin adibidez). Azkenik, bukaerako abiadura 0 ez bada, jokalaria mugitu egiten da.