

SOFTWARE INGENIARITZA

TAMAGOTCHI



Partaideak:

Asier Monge

Gorka Piedra

Jon Ander Duque

Asier Faraig

Andoni Ortiz De Zarate

Aurkibidea

1. SARRERA.....	3
2. ANTOLAKETA.....	4
3. DISEINUA.....	5
4. GARAPENA.....	8
5. ONDORIOAK.....	13

1. SARRERA

Softwarearen ingeniari-tza irakasgaiko proiektu finalean, Tamagotchi jokoaren aldaera bat ezarri dugu. Eguneroko arazoak (gaixotasuna, asperdura, gosea, higienarik eza...) gainditu behar dituen maskota birtuala da Tamagotchi jokoaren aldaera. Ohiko jokoen antzekoa da, eta honako puntu hauek laburbildu daitezke:

1. Hasierako pantailan sailkapen-taula bat agertuko da, aurreko partiden puntuazioekin. Gainera, partida berriak hasteko eta jokalaria berrien hasierakoak sartzeko aukera ematen du. Inizialak sartu ondoren, "Hasi" sakatzen badugu, jokoa hasi eta jokoaren pantaila nagusira sartuko gara.

2. Hasierako pantailan maskotentzako "arrautza" agertuko da lehenik. Lau bihotz daude ezkerrean, zure gogo-aldarteak adierazten dutenak; lau katilu zopa eskuinean, gosea adierazten dutenak. Gainera, egungo puntuazioa eta egoera ere testu gisa agertzen dira. Denborarekin, "arrautza" egoeratik "heldu" egoerara (Kuchipatchi) pasatzen da, eta, gero, hurrengo egoerara (Mimitchi).

3. Katilu eta Bihotzeko kontagailuak: kontagailu horiek 0 baliora iristen badira, Tamagotchia hil eta jokoa amaitu egiten da. Hori saihesteko, piruleta eta koilarakadak ematen badizugu, bihotz kopurua eta zopa-katiluen kopurua handituko ditugu.

Gainera, maskota elektronikoa gaixotu edo kaka egin dezake; beraz, tratatu eta garbitu egin behar dugu. Zenbat eta denbora gehiago egon gaixorik eta/edo zikin, orduan eta kalte handiagoa egingo dio tamagotchiari, eta katilu eta bihotz-kontagailuak ere kaltetuko egingo dira.

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

2. ANTOLAKETA

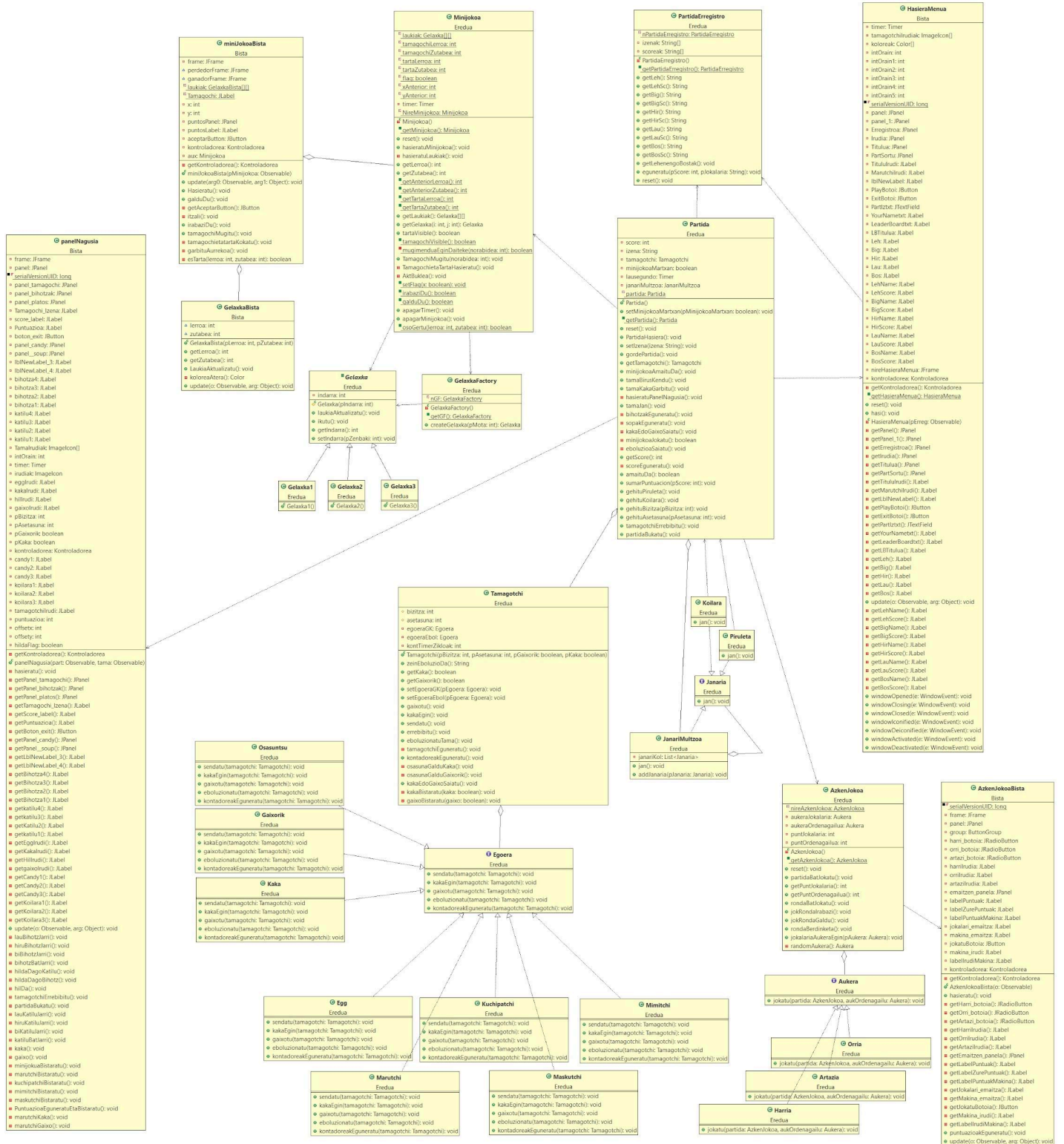
Lehenik eta behin, diagrama klasea egin genuen, klase bakoitza bere atributuekin eta metodoekin eta ondo egindako klaseen arteko erlazioekin. Ondoren, proiektuaren zati garrantzitsuenetako bat kodea bi zatitan antolatzea zen, ikuspegia eta eredua. Horretarako, MVC aplikatuko dugu, aurrerago azalduko duguna.

Atal nagusi horiek taldekideei esleitzen zaizkie eta aurrerapena aplikazio desberdinak erabiltzen dituzten lankide guztiekin partekatzen da. Alde batetik, WhatsApp taldeetan egunero sortzen diren aldaketak edo kezkak aipatzen dira. Gainera, bilerak bitan egiten dira, Discord aplikazioaren bidez. Aplikazioak aukera ematen dizu ordenagailuaren audioa eta pantaila partekatzeko, eta hori oso baliagarria da horrelako lanetarako.

Gure proiektuaren kode guztia landu da Visual Studion. Tresna erabilgarria da hainbat ordenagailutatik hainbat pertsonen arteko kodea editatzeko, eta, horretarako, baita GitHub ere erabili da, denok kodea eskuragarri izateko.

Proiektua hiru sprintetan banatuta egon da; sprint bakoitzeko, eginkizun batzuk bete behar izan ditugu, eta gero irakaslearekin elkarriketa bat egiten zen, proiektuak jaso beharreko aldaketen inguruan aholkuak jasotzeko: lehenengo sprinteko eginkizunak joko osoa kodetzea MVC patroia ondo aplikatuta, bigarrenekoak beste patroir batzuk aplikatzea(State, Factory eta Composite) eta hugarrenean hedapen bat proposatu eta implementatzea zen.

Irudia entregarekin batera gehitu dugu ere, kalitate hobean ikusi ahal izateko.



Kodearen diseinuari eta klaseen antolaketari dagokionez, esan daiteke Partida klasea hartu dugula nukleotzat, eta haren inguruan eraiki dugula kode guztia. Jokoa hastean Hasiera Menuaren bista da agertzen den lehen elementua eta hartatik Partida Erregistroak leaderboarda erakusten du, baina play botoia sakatzean Partida sortzen da, honek Panel Nagusia eraikitzen du, eta Partida klaseak kudeatzen du gainerakoa. Alde batetik, Partidak Tamagotchi bat du atribututzat, eta Tamagotchiak berak kudeatzen zein erabakitzen ditu bere bizitza, ahetasuna, eboluzioaren egoera, kakarik duen, gaixotzen den (hiru hauek State patroia aplikatuz) eta hiltzen den, guzti hauek Tamagotchiaren ezaugarriak baitira. Beste alde batetik, Partidak ere kudeatzen du TamaDigOut eta Harri-orri-ar minijokoak eraikitzeke erabakiak eta deiak, eta haietatik bertatik, ondoren, dagozkien bistak. TamaDigOut minijokoan Factory patroia erabiltzen da gogortasun maila ezberdineko gelaxkak sortzeko, baita MVC patroia ere, bistako gelaxkek detektatzen dituzten gertaerak eredu ke gelaxkei pasatzeko. Harri-orri-ar minijokoan, ordea, State erabili dugu, jokalarien aukera kudeatzeko modu errazagoan. Azkenik, Partida klaseak Partida Erregistroa eguneratzen du jokatu den partidaren emaitzarekin, eta leaderboarda kargatzen den hurrengo aldian, lortutako azken emaitza agertu egingo da, 5 hoberenen artean baldin badago.

MVC patroia

Gure programa bi zatitan banatuta dago, eredu zatia eta bista zatia. Eredu aldean zati logikoa aurki dezakegu, hala nola, kalkuluak, minijokoaren zati logikoaren garapena... Bestalde, bista zatiaren kodean pantailan ikusiko dena inplementatuta dago, adibidez, irudiak, bizitza, tamagotchiaren momentuaren eboluzioa...

Lehen esan dugunez bezala, bi alde daude, Eredu zatian hainbat klase ditugu, baina haietako lau Observable-ak dira, eta Bista aldekoak Observer-ak dira. Observable objektu batek Observer den objektu bati egoeraren aldaketan berri ematen die eta Observer-ak bistaratzen ditu.

Gure kasuan, Observable-ak dira hurrengo klaseak: *Minijokoa*, *Partida*, *Tamagotchi* eta *PartidaErregistro* klaseak. Klase hauetako bat aldageta bat egiten duenean *notifyObservers/setchanged* erabiltzen da, konkretuki, *Partida* klasean erabiltzen da, bihotzak eta koilarak eguneratzeko eta eboluzio bat egoten bada, hau da, Tamagotchi mota aldatzen bada. Beste aldetik, *Minijokoa* klasean, Tamagotchi-ren posizioa, matrizearen(laukiak) egoera eguneratzeko eta irabazi edo galdu duen esateko erabiltzen dugu *notifyObservers/setchanged*.

Bestetik, hiru Observer klase ditugu: *HasieraMenu*, *panelNagusia* eta *minijokoaBista* klaseak. Klase hauek *update()* metodoa erabiltzen dute Observable klaseen notifikazioak jasotzeko eta notifikazio bakoitzarako egin behar duen metodoa. Metodo hauen betekizuna irudiak eta bista eguneratzea da.

Listener kontutik, Listenerrek bi klaseetan erabiltzen ditugu: *HasieraMenua* eta *miniJokoaBista* klaseetan. Observer klaseetan erabiltzen da notifikatzeko eredu klaseetan aldaketa bat egon behar delako. Adibidez, Minijokoa amaitzean mezu bat agertzen da eta leiho hori “Aceptar” botoia sakatu eta gero itxi behar da eta hau Listenerrek egiten du.

Patroien azalpena

Proiektu honetan Software Ingeniaritza ikasgaian ikasitako hainbat patroia erabili dugu.

Factory

Factory patroia matrizeko gelaxketan erabili dugu. Patroi sortzaile bat da eta gelaxkak sortzeko erabiltzen dugu. 3 azpi klase daude, indarraren arabera, GelaxkaFactory klasearekin gelaxkak sortzen dira.

State

Sprint honetan State patroia inplementatu dugu gure jokuaren hiru zatitan; alde batetik, kaka eta gaixorik jartzeko mekaniketan. Horretarako Kaka, Gaixorik eta Osasuntsu klase berriak sortu ditugu, mekanika honek eskatzen dituen aldaketak egiteko. Beste alde batetik eboluzioetan ere aplikatu dugu patroia hau, honetan, lehen azaldu dugun moduan, tamagotchiren eboluzioetako bakoitza egiteko hainbat klase berri sortu egin ditugu. Bi kontu hauek kudeatzeko Egoera deitu dugun interfaze baten metodoak inplementatu ditugu. Eta azkenik, harri-orri-ar jokoan jokalariairen aukerak kudeatzeko (harria, orria edo artazia) state patroia erabili dugu ere, Aukera interfazeko metodoak inplementatuz.

Composite

Composite patroia erabili dugu Tamagotchiri piruletak eta koilarak emateko. Javako Composite patroiak objektuak zuhaitz gisa egituratzeko balio du. Objektu individualak nahiz objektu-konposizioak modu berean tratatzeko aukera ematen du, eta horrek aldi berean piruletak eta koilarak ematea ahalbidetzen digu.

Observer

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

4. GARAPENA

1. Sprint

Lehenengo sprintaren elkarriketa egiterakoan gure jokoa ez zuen ondo funtzionatzen, pantaila aldatzeko momentuan ez zen ondo egiten, eta tutoretza baten ondoren ikusi genuen hainbat partida sortzen zirela zela arazoa, arazo hau konpondu eta gero aurreratu genuen proiektuan.

Atazen dokumentazioa

Sprint	Ataza	Arduraduna	Denbora planif.	Denbora erreal	Oharrak
1	HasieraMenua Klasea	Jon Ander	2 egun	2 egun	
1	miniJokoaBista Klasea	Andoni eta Gorka	2 egun	2 egun	Zailtasunak egon ziren MVC patroiak inplementatzeko baina tutoretza baten laguntzaren esker ondo bideratu genuen
1	panelNagusia Klasea	Asier Monge	2 egun	3 egun	
1	Tamagochi mota desberdinak	Asier Faraig	1 egun	1 egun	
1	Minijokoa Klasea	Andoni eta Gorka	2 egun	2-3 egun	Klasea inplementatzen ari ginen bitartean zailtasunak izan genituen baina banan-banan konponduz eta dagokien ordenan ondo egin ahal izan genuen
1	Partida Klasea	Asier Monge	1 egun	2 egun	Klase honetan hainbat zailtasun eduki genituen hasiera. Izan ere, aski kostatu zitzaigun

					desberdintzea zein zati zijoan bistara eta zein kodera. Baina behin hori desberdinduta errazago egin zitzaigun. Baina bidean hainbat zailtasun eduki ditugu.
1	Klase diagrama	Jon Ander eta Asier Faraig	1 egun	1 egun	
1	Dokumentazioa	Gorka	2 ordu	3 ordu	Denon artean ondo koordinatutak egotea garrantzitzua izan da.
1	Pantaila/panel desberdinak konektatzea	Jon Ander, Gorka eta Asier Monge	2 ordu	4 ordu	
1	Irabazi eta Galdu panelak	Andoni	1 ordu	2 ordu	
1	Irudien inplementazioa	Jon Ander, Andoni eta Asier Monge	3 ordu	4 ordu	
1	Metodo txikiak	Guztiak	1 ordu	1 ordu	Bigarren mailako metodoak

2. Sprint

Sprint honetan ikasgelan ikasitako patroiak aplikatzen hasi ginen, hala nola, *Factory*, *State* edo *Composite*. Gainera, minijokoaren matrizea egiteko modua aldatu behar izan genuen, gelaxka klasea sortu eta mota horretako matrizea egin.

Atazen dokumentazioa

Sprint	Ataza	Arduraduna	Denbora planif.	Denbora erreala	Oharrak
2	Aurreko sprintean	Guztiak	5 egun	6 egun	Tutoretza batera

	izan genituen arazoak konpontzea.				joan ginen eta han ikusi genituen zeintzuk ziren arazoak.
2	Gelaxka klasea eta Factory patroia.	Gorka	2 ordu	2 ordu	
2	State patroia.	Asier Faraig eta Jon Ander	2 egun	3 egun	
2	Dokumentazioa.	Andoni eta Asier Monge	1 egun	1 egun	
2	Metodo txikiak.	Guztiak	1 ordu	1 ordu	Bigarren mailako metodoak.

3. Sprint

Sprint honetan jokoaren hedapen bat proposatu behar genuen eta gure proposamena beste minijoko bat gehitzea jokoaren amaieran zen. Tamagotchia hiltzen denean, jokolaria aukera bat du berpizteko, horretarako harri-orri-ar jokoan irabazi behar du.

Atazen dokumentazioa

Sprint	Ataza	Arduraduna	Denbora planif.	Denbora erreal	Oharrak
3	Harri-orri-ar minijokoa	Guztiak	2 egun	3 egun	
3	Composite patroia.	Gorka	2 ordu	2 ordu	
3	Observer patroia ondo jarri gelaxketan	Asier Faraig eta Jon Ander	1 egun	1 egun	
3	Aurreko elkarrizketan agertutako arazoak	Guztiak	2 egun	2 egun	
3	Dokumentazioa.	Andoni eta Asier Monge	1 egun	1 egun	
3	Metodo txikiak.	Guztiak	1 ordu	1 ordu	Bigarren mailako metodoak.

Proben dokumentazioa

ID	Helburua	Sarrera	Exekuzio baldintzak	Esperotako emaitza	Lortutako emaitza	Bukatuta
1	Eboluzioa frogatu	Denbora eta bizi	Bihotz-puntuak positiboak izan	Hurrengo eboluziora pasatu	Ebaoluzionatu du	BAI
2	Minijokoa frogatzea	Minijokoa saltatzea	Portzentaia egokia izatea	Minijokoa irabazi edo galdu	Irabazi edo galdu egitea	BAI
3	Gelaxken indarra ondo kudeatu	Gelaxkak indarra izatea(1tik 3ra)	Gelaxka guztiak indar zenbaki bat izatea	Gelaxken guztien indarra 0 izatea	Gelaxkak guztiak 0 indarrarekin geratu al dira	BAI
4	Tamagotchi jatea	Bizitza eta koilarak	Piruleta eta koilarak jartzea	Jaten duenen bizitza eta koilarak igotzea	Puntuak igotzen dira	BAI
5	Harri-orri-ar minijokoa	Tamagotchia hiltzen da	Tamagotchia hil behar da	Partida irabaztea edo galtzea	Irabazi edo galdu egitea	BAI
6	LeaderBoard frogatu	LeaderBoard hasieratuta egotea	Partida amaitzea	Jokatutako partida LeaderBoard -ean agertzea puntuarekin	Partida agertzen da puntuarekin	BAI
7	Panelak batetik batera pasatzea	Panelak batetik batera pasatzea	Hainbat panel egotea	Panelen arteko lotura ondo izatea eta behar den momentuan aldatzea	Panelak ondo aldatzen dira	BAI
8	Puntuak frogatu	Denbora eta bizi	Bihotz-puntuak positiboak izan	Puntuak eguneratzea	Puntuak eguneratzen dira	BAI

5. ONDORIOAK

Proiektu honetan denbora asko eman ondoren, asko ikasi genuen. Horien artean, talde-lanak lana banatzea eta elkarri laguntzea esan nahi du, arazoak daudenean. Ondo lan egiten dugu gure artean, guztiek beren trebetasunak parteka ditzaten eta beren ahuleziak gogora ditzaten. Gure lanaren zati zuzena egin dugula uste dugu, eta beti prest gaude elkarri laguntzeko.

Proiektuaren hasieran, 1. sprintean, jokoak ez zuen irakasleak nahi bezala funtzionatzen, baina konpontzeko nahikoa denbora inbertitu genuen. Gaur egun dugun ezagutzaren arabera, esan dezakegu proiektu hau egokia dela eta nahikoa ikasten dela. Datorren urtean edo etorkizunean programa honetara itzultzen bagara, ziur gaude oraindik asko dugula hobetzeko.