

Battleship FeatureIDE

Julen Etxaniz

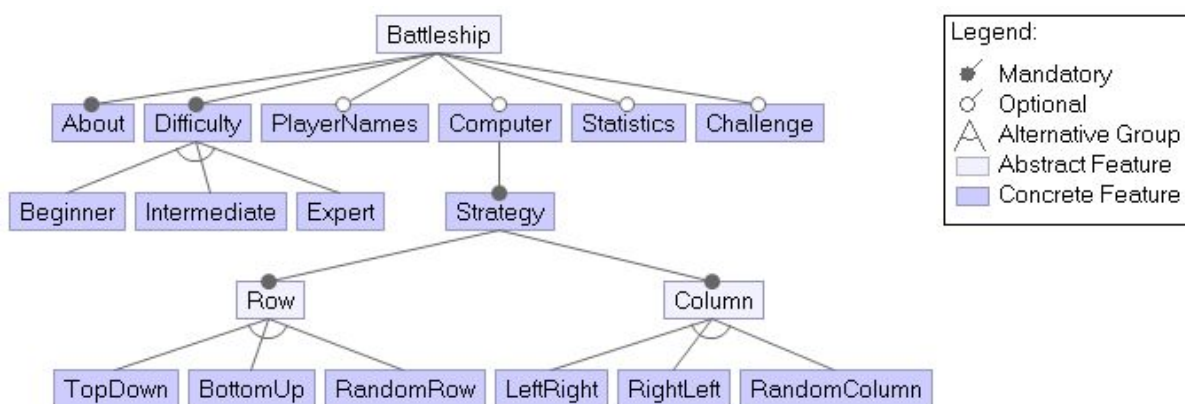
2020/06/01

Aurkibidea

Ezaugarri diagrama	2
Aldakortasun teknikak	2
Denbora: 37 ordu	4
Kodea aukeratu: 2 ordu	4
Kodea banatu: 5 ordu	5
Kodea aldatu: 5 ordu	5
Kode berria inplementatu: 20 ordu	5
Dokumentazioa: 5 ordu	6
Erreferentziak	6

Ezaugarri diagrama

Enuntziatuan zehaztutako ezaugarri guztiak implementatu dira, hautazko zatia izan ezik. Clever ezaugarria diagraman bakarrik azaltzen zenez ez da implementatu, ez zegoelako argi zer zen. Diagraman ezaugarrien ordena aldatu da aldagarritasuna errazteko. Row eta Column ezaugarriak abstraktu bihurtu dira ez zeukatelako koderik.



Aldakortasun teknikak

Taula honetan ezaugarri bakoitzeko aldakortasuna deskribatzen da, klase bakoitzean dauden metodoak eta atributuak deskribatuz.

Ezaugarria	Aldakortasuna
About	-Battleship klasean printAbout() metodoa erabili da. Ezaugarri bakoitzean hasieran original() erabili da eta beharrezko kodea gehitu da atzetik. Beheko mailan dauden ezaugarrietan bakarrik gehitu da. Adibidez, Difficulty ezaugarrian ez baina Beginnerren bai.
Difficulty	-Battleship klasean hasierako kodearen gehiengoa jarri da, ordenagailuarekin zerikusia duten zatiak izan ezik. Kodea metodotan banatu da ondorengo klasetan aldakortasuna errazago aplikatzeko. Batzuetan metodo horiek guztiz ordezkatu dira eta bestetan original() erabiliz osatu dira. -Grid klasean hasierako kodea jarri da aldaketa batzuekin. Inprimatzeko metodoan aldaketak egin dira kodea ez errepikatzeko eta aldagarritasuna hobetzeko. Beste ezaugarrietan aldatu da. -Location klasean hasierako kodea jarri da metodo eta atributu gehigarri batzuekin. Geroago ez da inongo ezaugarritan aldatu. -Player klasean hasierako kodea jarri da metodo eta atributu gehigarri batzuekin. Beste ezaugarrietan aldatu da. -Ship klasean hasierako kodea jarri da metodo eta atributu gehigarri batzuekin. Beste ezaugarrietan aldatu da.
Beginner	-Grid klasean 3 konstante gehitu dira, errenkada kopurua, zutabe

	<p>kopurua eta tiro kopurua.</p> <p>-Player klasean itsasontzien luzerak zehaztu dira. Itsasontzi mota bakoitzeko hainbat itsasontzi jartzea erabaki da, bestela gutxiegi daudelako.</p> <p>-Ship klasean itsasontzien mota enumeratua gehitu da. Metodo eraikitzailean luzeraren arabera itsasontzi mota zehaztu da.</p>
Intermediate	<p>-Grid klasean 3 konstante gehitu dira, errenkada kopurua, zutabe kopurua eta tiro kopurua.</p> <p>-Player klasean itsasontzien luzerak zehaztu dira. Itsasontzi mota bakoitzeko hainbat itsasontzi jartzea erabaki da, bestela gutxiegi daudelako.</p> <p>-Ship klasean itsasontzien mota enumeratua gehitu da. Metodo eraikitzailean luzeraren arabera itsasontzi mota zehaztu da.</p>
Expert	<p>-Grid klasean 3 konstante gehitu dira, errenkada kopurua, zutabe kopurua eta tiro kopurua.</p> <p>-Player klasean itsasontzien luzerak zehaztu dira. Itsasontzi mota bakoitzeko hainbat itsasontzi jartzea erabaki da, bestela gutxiegi daudelako.</p> <p>-Ship klasean itsasontzien mota enumeratua gehitu da. Metodo eraikitzailean luzeraren arabera itsasontzi mota zehaztu da.</p>
PlayerNames	<p>-Battleship klasean jokalariei izena ezartzeko metodoak aldatu dira, erabiltzaileak idatzitako izena esleitzeko. Ez dago aldaketa gehiago egin beharrik nahiz eta ezaugarri hau ez aukeratu jokolariaren izena inprimatzen delako. Aukeratu ez bada, defektuzko izenak Player1, Player2 eta Computer dira.</p>
Computer	<p>-Battleship klasean bigarren jokalariren setup metodoak ordezkatu dira, jokolariaren izena computer izateko eta itsasontziak ausaz kokatzeko errorerik gabe.</p> <p>-Randomizer klasean hasierako kodea utzi da, nahiz eta metodo gehienak behar ez izan.</p>
Strategy	<p>-Battleship klasean playGame eta makeGuess metodoak ordezkatu dira ordenagailuak automatikoki errenkada eta zutabeak kalkula ditzan txanda bakoitzean. Kalkulatzeko erabilitako calculateRow() eta calculateCol() metodoak azpiko ezaugarritan zehaztutakoak izango dira</p>
TopDown	<p>-Battleship klasean row atributua -1 balioarekin eta calculateRow() metodoa gehitu dira. Metodo honek row atributua inkrementatu eta itzultzen du. Mugara iristean berriro hasieratzen da.</p>
BottomUp	<p>-Battleship klasean row atributua errenkada kopuruaren balioarekin eta calculateRow() metodoa gehitu dira. Metodo honek row atributua dekrementatu eta itzultzen du. Mugara iristean berriro hasieratzen da.</p>
RandomRow	<p>-Battleship klasean calculateRow() metodoa gehitu dira. Metodo honek ausazko balio bat itzultzen du, 0 eta errenkada kopuruaren artekoa.</p>

LeftRight	-Battleship klasean col atributua -1 balioarekin eta calculateCol() metodoa gehitu dira. Metodo honek col atributua inkrementatu eta itzultzen du. Mugara iristean berriro hasieratzen da.
RightLeft	-Battleship klasean col atributua zutabe kopuruaren balioarekin eta calculateCol() metodoa gehitu dira. Metodo honek col atributua dekrementatu eta itzultzen du. Mugara iristean berriro hasieratzen da.
RandomColumn	-Battleship klasean calculateCol() metodoa gehitu dira. Metodo honek ausazko balio bat itzultzen du, 0 eta zutabe kopuruaren artekoa.
Statistics	-Battleship klasean playGamePlayer(Player player1, Player player2) metodoak hedatu dira estatistikak inprimatzeko, hasieran original(player1, player2) erabiliz. -Grid klasean asmatutako eta huts egindako tiro kopurua kontatzeko atributuak gehitu dira getter-ekin batera. Hauen balioa aldatzen joateko markHit(row, col) eta markMiss(row, col) metodoak hedatu dira, lehenengoa ordezkatzuz eta bigarrena hasieran original(row, col) erabiliz. -Player klasean printStatistics(Player opp) metodoa gehitu da, jokariaren estatistikak inprimatzeko. Lehenengo lerroan egindako, asmatutako eta huts egindako tiro kopurua. Bigarren lerroan jotako, hondoratutako eta aurkitu beharreko itsasontzi kopurua. Hurrengo lerroetan itsasontzi mota bakoitzeko jotako, hondoratutako eta aurkitu beharreko kopurua.
Challenge	-Battleship klasean playGame() metodoa ordezkatu da, tiro kopurua eta puntuak kontuan hartzeko. Puntu gehien egin duena irabazlea da. Berdinketaren kasuan, tiro gehien asmatu dituen izango da irabazlea, eta bestela parra egin dutela idatziko da. Jokalarietako batek itsasontzi guztiak urperatzen baditu, beste jokalaria beste horrenbeste tiro botatzen utziko zaio. markResult() metodoa hedatu da hasieran original(p, opp, row, col, oldRow, oldCol) erabiliz, puntuak eta falta diren tiro kopurua inprimatzeko. markHit() eta markMiss() metodoak ordezkatu dira tiro bakoitzarekin puntuak gehitu eta inprimatzeko.

Denbora: 37 ordu

Hemen agertzen diren denborak gutxi gorabeherakoak dira, ez baita denboraren erregistrorik egin. Deskribapenekin ulertu daiteke atal bakoitzean zenbateko lana egin behar izan den. Orokorrean, pentsatzen nuena baino askoz denbora gehiago behar izan dut. Izan ere, hasieran pentsatzen nuen kode berria inplementatzeko lan gutxiago egongo zela.

Kodea aukeratu: 2 ordu

Kodea aukeratzeko enuntziatuan agertzen diren bi aukerak analizatu eta probatu dira. Lehenengoa askoz osatuagoa dagoela ikusi da eta beraz hori izan da lanean hasteko kodea. Eclipse Java proiektu bat sortu da hasierako kodearekin eta hori ez da aldatu,

aldaketak FeatureIDE proiektuan bakarrik egin dira. Horrek kodea pixkanaka hartzen eta konparatzen joatea ahalbidetu du.

Kodea banatu: 5 ordu

Kodea metodotan banatu da, aldagarritasuna errazteko eta errepikapenak sahiesteko. Kode zati bakoitza non jarri behar den erabakitzea ez da horren zaila izan. Baina, banatzerakoan aldakortasuna inplementatzea zailagoa izan da, eta horretarako kodea metodotan banatzeak lagundu du. Kasu batzuetan aldakortasuna original() bidez inplementatu da eta bestetan berriz metodoa ordezkatzuz. Batzuetan ezaugarrien ordenarekin ere arazoak egon dira eta azkenean ordena aldatzea erabaki dut.

Kodea aldatu: 5 ordu

Nahiz eta lan honen helburua ez izan kodea hobetu da ulergarritasuna eta aldagarritasuna hobetzeko. Adibidez, metodo laguntzaileak aldatu dira, switch batekin kasu guztiak kontuan hartu ordez, kasu guztietarako balio duen kalkulua egin da. Horrela, kodea askoz motzagoa da eta gainera ez dago aldagarritasunagatik kasuak gehitu edo kendu beharrik.

Aldagarritsunarekin zerikusia duen beste hobekuntza zenbakiak klaseetan definitutako kostanteekin ordezkatzeko izan da. Adibidez, Grid-en errenkada eta zutabe kopuruaren konstanteak erabili dira zenbakien ordez. Horrela, nahikoa da leku bakar batean aldaketa egitea eta lan gehigarria eta akatsak sahiesten dira.

Beste hobekuntza txiki bat Randomizer klaseko metodoak modu estatikoan deitzea izan da, bestela warning bat agertzen baita.

Gainera, erabiltzaileak itsasontzien errenkada, zutabe eta norabidea sartzerakoan salbuespenen tratamendua gehitu da erroareak sahiesteko. Izan ere, gerta zitekeen partida erdian akatsen bat egitea eta partida berriro hasi behar izatea.

Jokatzerakoan agertzen diren mezuak eta horien ordena ere aldatu da, jokaldien ordena argiago gera dadin. Gainera, horrela ordenagailuaren eta beste jokalarari baten aurka jokatzerakoan antzeko esperientzia izatea lortzen da.

Kode berria inplementatu: 20 ordu

Nahiz eta jokoaren oinarria eginda egon, ezaugarri askotan kodea berria inplementatu behar izan da eta honek lan asko ekarri du. Adibidez, Statistics ezaugarrian dena inplementatu gabe zegoen. Computer ezaugarrian, berriz, oinarria jadanik inplementatuta zegoen baina aldakortasuna falta zen. Itsasontziak ez ikutzearena hemen bakarrik aipatzen zen arren jokalarari guztientzako araua dela suposatu da. Funtzionalitate batzuk inplementatzeko aurretik zegoen kodean ere aldaketa batzuk egin behar izan dira.

Dokumentazioa: 5 ordu

Kodean komentarioak gehitu dira metodo bakoitzaren deskribapenarekin.

Dokumentu honetan ezaugarri diagrama, ezaugarri bakoitzeko aldakortasuna, denbora eta erreferentziak zehaztu dira.

Erreferentziak

Jatorrizko kodea hemendik hartu da:

<https://github.com/ymarcus93/Java-Battleship>

Beste erreferentziak, jokoaren inguruko ideiak hartzeko:

[https://en.wikipedia.org/wiki/Battleship_\(game\)](https://en.wikipedia.org/wiki/Battleship_(game))

<https://en.wikipedia.org/wiki/Warship#Types>

[https://battleship.fandom.com/wiki/Battleship_\(game\)](https://battleship.fandom.com/wiki/Battleship_(game))

<https://www.ultraboardgames.com/battleship/variations.php>