

PMOBO: Ikasgaiaren Proiektua.

Proiektuaren Helburu-dokumentua

Taldearen izena: WhiteMesa

Data:2015/05/11



Sarrera eta jokoaren deskripzioa

Blackjack, hogeitabat ere deitua, kasinoetan jokatzen den karta jokia da. Jokoaren helburua 21era heltzea da, baina pasatu gabe. Horretarako karten balioa gehituz joango da. Karten balioa daukaten zenbakiaren berdina izango da, salbuespen batzuekin. J, Q eta K kartak 10 balio dute eta A-k 11 (eskua ≤ 21 bada) edo 1. Bi kartekin 21 lortzen bada, Blackjack dela esaten da eta automatikoki irabazten da. Blackjackera jolasteko pausuak hauek dira:

- Apostuak: Apostuak egiterako orduan jokalariek nahi duten diru kopurua apostatu ahal dute, gutxienez aurrekoak apostatu duena berdinduz. Apostu guztiak berdinak ez izatekotan, apustu txikiegia daukaten jokalarietatik berriro pasatzen da ea berdindu nahi duten galdetzeko. Momenturen batean 0€ apostatzen badute edo apostua ez badute berdintzen kasinotik joan beharko dira.
- Lehenengo karta banaketa: Hasiera baten jokalaria guztiei bi karta ematen zaizkie, momentu honetan jokalaria baten eskua 21ekoa bada eta eskuan dituen kartak as (A) bat eta figura bat (J, Q, K) badira, jokalaria horrek Blackjack izango du eta automatikoki irabaziko du.
- Bigarren karta banaketa: Lehenengoan 21 heldu ez direnak karta gehiago eskatzeko eskubidea izango dute, nahi beste karta eskatu ahal dituzte 21etik pasatzen ez badira. Momentu honetan, croupierarekin jolasten badaude doblatzeko aukera ematen zaie. Doblatzea, zure apostua bikoiztea, karta bat hartzea eta plantatzea da. Doblatzen denean, croupierrak ere apostua doblatu beharko du.
- Irabazlea kalkulatu: Guztien txanda bukatu denean irabazlea kalkulatzeko momentua heldu da. Croupierrik gabe, irabazlea orden honetan kalkulatzen da: Blackjack > 21 > 20 > 19 > ... > 1. Jokalaria guztien eskua > 21 bada ez dago irabazlerik. Croupierra egoteko kasuan irabazle bat baino gehiago egon ahal dira. Goiko metodo berdina jarraitzen da, baina irabazle moduan croupierra baino esku hobeagoa daukaten jokalaria guztiak izango dira.



Proiektuaren Helburuak

Proiektuaren helburua BlackJack jokoa inplementatzea izango da. Jokoa 2 eta 7 jokalarien artean jokatuko da eta jokalaria bakoitza 500 €-rekin hasiko da beti. Partida bat hastean, hasierako jokalariek partidatik alde egin dezakete, baina inork ezin izango du partidara sartu.

Gainera, informatikako laborategietan ikasitako guztia aplikatuko dugu. Aplikatuko ditugun ezagutzen artean, EMA-ak, iteradoreak, atributu estatikoak, Singleton-patroia, Array-ak,... daude.

Bigarren mailako helburu bezala, herentzia erabiliko ditugu jokoa konplexuagoa egiteko (kasinoaren IA egiteko eta kartetan erabiltzeko) eta hainbat “*Easter Egg*”, adibidez kontsolan pantailaratzen diren emotikono guztiak bistaratzea.

Azkenean, hondorengo listan dauden gauzak egin ditugu:

- Lehenengo mailako helburuak
 - Jokalarien arteko blackjacka inplementatzea
- Bigarren mailako helburuak
 - Croupierraren kontra jolastea
 - Emotikonoak bistaratzea
- Inplementazioan zehar bururatu zaizkigunak
 - Jolasteko bi metodoak bateratzea
 - Musika
 - Logroak (Emotikonoen handipen bat)
 - Ranking eta ListaPartidak
 - Jokalarien balantzea indibidualki azaltzea
 - "Double" Baldintza sartzeara

<ul style="list-style-type: none"> org.pmoos.blackjack.Logroak helbidea: Logroak blackjack: boolean bankarrita: boolean erretiratu: boolean lagunkaGabe: boolean lagunaAste: boolean mahaikoErregea: boolean bikoitza: boolean helloWorld: boolean ludopate: boolean emotikonoak: boolean pistarenErregea: boolean logroenErregea: boolean
<ul style="list-style-type: none"> Logroak() <ul style="list-style-type: none"> getNireLogroak(): Logroak blackJackLogroa(): void erretiratuLogroa(): void bikoitzaLogroa(): void logroenErregeaLogroa(): void helloWorldLogroa(): void ludopateLogroa(): void bankarritatLogroa(): void mahaikoErregeaLogroa(): void lagunkaGabeLogroa(): void lagunaAsteLogroa(): void emotikonoakLogroa(): void pistarenErregeaLogroa(): void

<ul style="list-style-type: none"> org.pmoos.blackjack.ListaEmotikonoak helbidea: ListaEmotikonoak lista: Array<String> ListaEmotikonoak() <ul style="list-style-type: none"> getNireListaEmotikonoak(): ListaEmotikonoak getIteradore(): Iterator<String> getHiruEmotikonoa(pEmo: String): void emotikonoBadaGo(pEmo: String): boolean listarenTamaina(): int
--

<ul style="list-style-type: none"> org.pmoos.blackjack.Lokalaria sci: Scanner eskua: String izen: String dirua: int apostua: int erretiratu: boolean Jokalaria(pIzena: String) <ul style="list-style-type: none"> apostuaEgintu(): void bueltatu(): void apostuaBueltatu(): void apostuakusitu(): void eskuakKalkulatu(): int karatskatu(): void boteaHartu(): void boteaHartuCroupier(pBotea: int): void bueltatu(): void getIzena(): String getDirua(): int eskubetara(): void eskuakidatzu(): void blackJackDua(): boolean galdeWutan(): boolean diruakPrimatu(): void karakopu(): int
<ul style="list-style-type: none"> org.pmoos.blackjack.ListaLokalariek sci: Scanner helbidea: ListaLokalariek lista: Array<Jokalaria> ListaLokalariek() <ul style="list-style-type: none"> getNireListaLokalariek(): ListaLokalariek getIteradore(): Iterator<Jokalaria> eskuHandienakKalkulatu(): Jokalaria JokalariekBueltatu(): void tamaina(): int erretiratu(): void guztiaErretiratu(): void boteaBanatu(pBotea: int, pRanking: Ranking): void JokalariaKalkulatu(): void croupierInskribatu(): void hasierakoBikartak(): void apostuak(): void apostuGuztiakKusitu(): void karakBanatu(): void kenduKartak(): void guztienDiruakPrimatu(): void batBainoGehiagoItebatu(): boolean rankingCroupier(): Ranking rankingErCroupier(): void apostuakBueltatu(): void galdeWutan(): void erretiratu(pJok: Jokalaria): void JokalarikMintu(): int

<ul style="list-style-type: none"> org.pmoos.blackjack.Croupier Croupier(pIzena: String) <ul style="list-style-type: none"> getIteradore(): Iterator<Karta> sartu(pProb: double): boolean apostuaEgintu(): void apostuakusitu(): void galdeWutan(): boolean bueltatu(): void probakKalkulatu(pEskua: int): double diruakPrimatu(): void
--

<ul style="list-style-type: none"> org.pmoos.blackjack.ListaKartak kartak: Array<List<Karta>> ListakKartak() <ul style="list-style-type: none"> getHiruKarta(pKarta: Karta): void erretiratu(): void getIteradore(): Iterator<Karta> kartenBaloakLortu(): int batekorkDago(): boolean zenbatBatekoDago(): int tamaina(): int

<ul style="list-style-type: none"> org.pmoos.blackjack.Karta zenbakia: int palua: int Karta(pZenbakia: int, pPalua: int) <ul style="list-style-type: none"> idatzipalua(): String getKartabaloa(): int batekoa(): boolean kartakidatzu(): String figuraDa(): boolean
--

<ul style="list-style-type: none"> org.pmoos.blackjack.Blackjack sci: Scanner helbidea: Blackjack apostuMax: int botea: int croupierarekin: boolean main(ARGS: String[]): void BlackJack() <ul style="list-style-type: none"> getNireBlackJack(): Blackjack partidaLolastu(): void croupierarekin(): boolean irabazicCroupierGabe(): void irabazicCroupierarekin(): void irabazteakKalkulatu(): void JokoaAmaitu(): void getApostuMax(): int setApostuMax(pApostuMax: int): void getBotea(): int setBotea(pBotea: int): void partidaAmaitu(): void kontsolagarbitu(): void entzertaratu(): void rankingakKusiNahi(): boolean rankingakInprimatu(): void batfidoztu(): boolean

<ul style="list-style-type: none"> org.pmoos.blackjack.Baraja helbidea: Baraja listaKartak: Array<List<Karta>> Baraja() <ul style="list-style-type: none"> getBarajak(): Baraja getIteradore(): Iterator<Karta> erretiratu(): void barajatu(): void emankartatu(): Karta tamaina(): int barajakPrimatu(): void
--

<ul style="list-style-type: none"> org.pmoos.blackjack.ListaPartidak helbidea: ListaPartidak listaPartidak: Array<List<Ranking>> halOffName: Array<List<Jokalaria>> Listapartidak() <ul style="list-style-type: none"> getNireListaPartidak(): ListaPartidak getIteradore(): Iterator<Ranking> getIteradoreaz(): Iterator<Jokalaria> partidaOorde(pRanking: Ranking): void partidakidatzu(): void badagoHalOffName(pJok: Jokalaria): boolean halOffNameBete(): boolean

<ul style="list-style-type: none"> org.pmoos.blackjack.Ranking ranking: Array<List<Jokalaria>> ESKUA: Comparator<Jokalaria> Ranking() <ul style="list-style-type: none"> getIteradore(): Iterator<Jokalaria> rankingeanSartu(pJokalaria: Jokalaria): void irabazteKopu(): int rankingaOndentatu(): void posizioan(pJok: int): Jokalaria partidakidatzu(): void irabazteakInprimatu(): void norDatuhenengo(): Jokalaria rankingeanDago(pJok: Jokalaria): boolean



Klaseen azalpena

- Karta: Karta bakoitzaren informazioa gordeko duen klasea (zenbakia eta palua)
- ListaKartak: Jokalari bakoitzaren eskua gordeko duen klasea. Haren metodoen artean kartenBalioaKalkulatu() dago, zeinek esku osoaren balioa kalkulatu duen, batekoen kasu larriak kontuan hartuta ere.
- Jokalaria: Jokatuko duen pertsona bakoitzaren dirua, izena, eskua... gordeko duen klasea. apostuaEgin(), kartaEskatu() (barajari) eta txanda() metodoak ditu, adibidez. Txanda metodoaren barruan, ea karta gehiago nahi dituen ala ez galdetuko zaio jokalaria, eta haren eskuaren balioa 21 baino handiagoa bada, partidatik kanpo geratuko da hurrengo partida hasi arte.
- Croupier: Jokalaritik heredatutako klasea, ordenagailuak kontrolatzen duen jokalaria da eta banka moduan jokatzen du.
- Baraja: Kartak nondik hartuko diren gordeko duen klasea da. Aipagarria da barajatu() metodoa, baraja desordenatzen delako *Shuffle* metodoa erabiliz.
- ListaJokalariak: Partidan jokatzen ari diren pertsonak gordetzen duen klasea da. Jokoa abiarazteko behar diren ia metodo guztiak ditu barnean.
- BlackJack: Klase nagusia, bertatik joko osoa kudeatuko da.
- Ranking: Partida bateko emaitzak gordeko ditu. Rankingaren barruan jokalaria guztiak egongo dira eskuaren tamainaren arabera ordenatutak.
- ListaPartidak: Ranking array bat da. Hemen jokatu diren partida guztiak gordetzen dira.
- Logroak: Partidan zehar desblokeatu daitezkeen logroak eta beraien egoera gordetzeko erabiltzen den klasea.
- ListaEmotikonoak: Emotikonoak kudeatzeko klasea. Emotikono guztiak desblokeatuta daudenean logro bat lortuko da.

Sekuentzia diagrama handiegia denez .jpg artxibo moduan igo dugu

[Sekuentzia diagrama \(GitHub\)](#)