

# ***PMOBO: Ikasgaiaren Proiektua.*** ***Proiektuaren Helburu-dokumentua***

Taldearen izena: WhiteMesa Data: 2015/05/05



## ***Sarrera eta jokoaren deskripzioa***

Blackjack, hogeitabat ere deitua, kasinoetan jokatzen den karta jokia da. Jokoaren helburua 21era heltzea da, baina pasatu gabe. Karten balioa gehitu egiten da, J, Q eta K kartak 10 balio dute, eta A-k 11 edo 1 balio du, hamaikak eta beste kartak 21 baino gehiago lortuko balu. Bi kartekin 21 lortzen bada, Blackjack dela esaten da eta automatikoki irabazten da.



## ***Proiektuaren Helburuak***

Proiektuaren helburua BlackJack jokia 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 jokia konplexuagoa egiteko (kasinoaren IA egiteko eta kartetan erabiltzeko) eta hainbat “Easter Egg”, adibidez kotsolan pantailaratzen diren emotikono guztiak bistaratzea.

Gainera, laborategietan eta magistraletan ikasi ez ditugun ezagutzak erabiliko ditugu (shuffle metodoa).



## ***Alderdi teknikoak eta behin behineko diseinua***

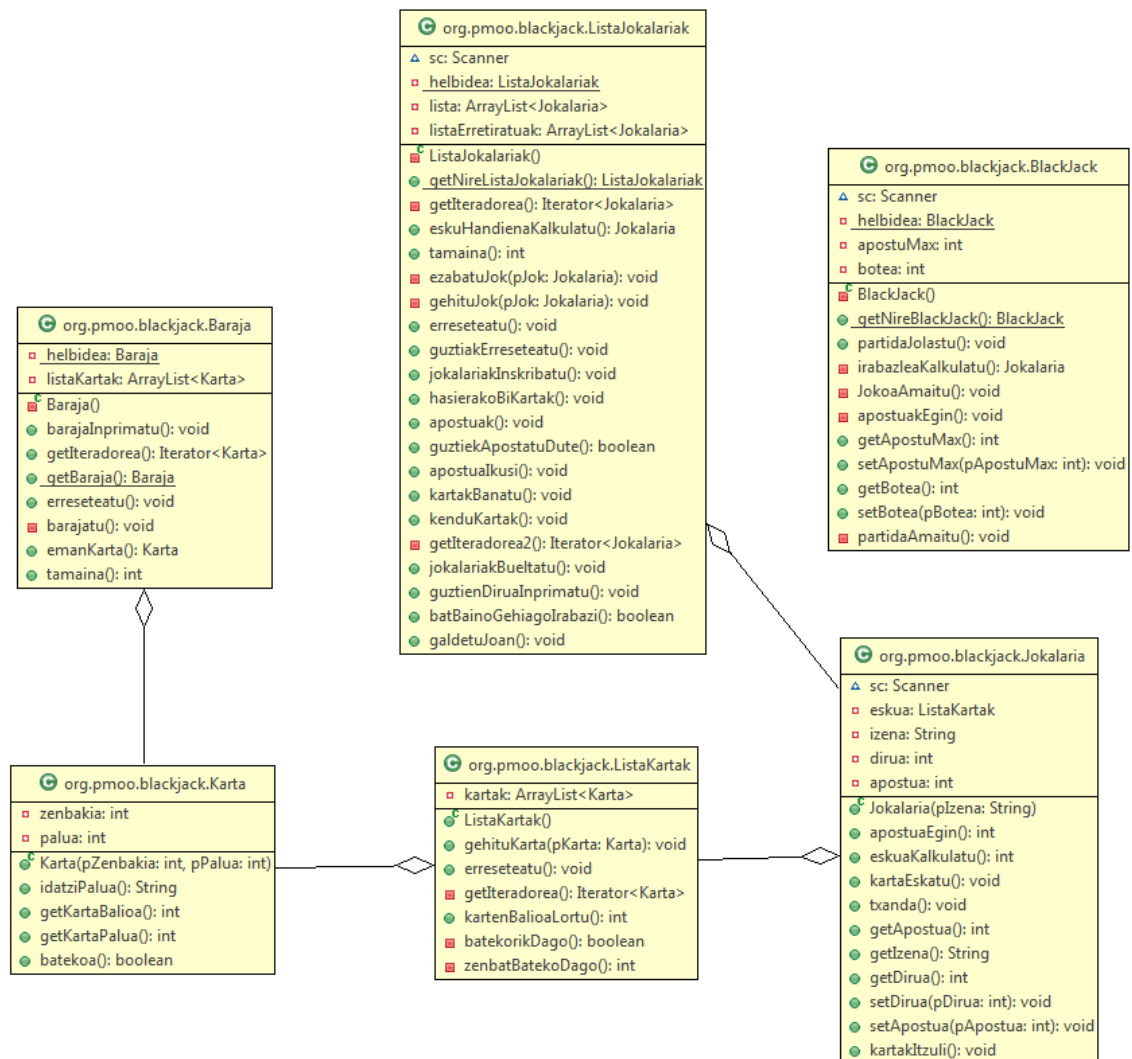
Partaideen agenda oso beteta dagoenez (laborategiak), bilerak Skype, Team Speak, Team Viewer edo beste software-en bitartez egingo ditugu. Proiektua guztien artean sinkronizatuta edukitzeko gitHub plataforma eta SourceTree programa erabiliko ditugu.

Proiektuaren lan zatien asignazioak gitHub plataformaren bidez egingo ditugu, issues erreminta erabiliz.

Gure proiektua hemen alokatua egongo da:

<https://github.com/Thadah/BlackJack>

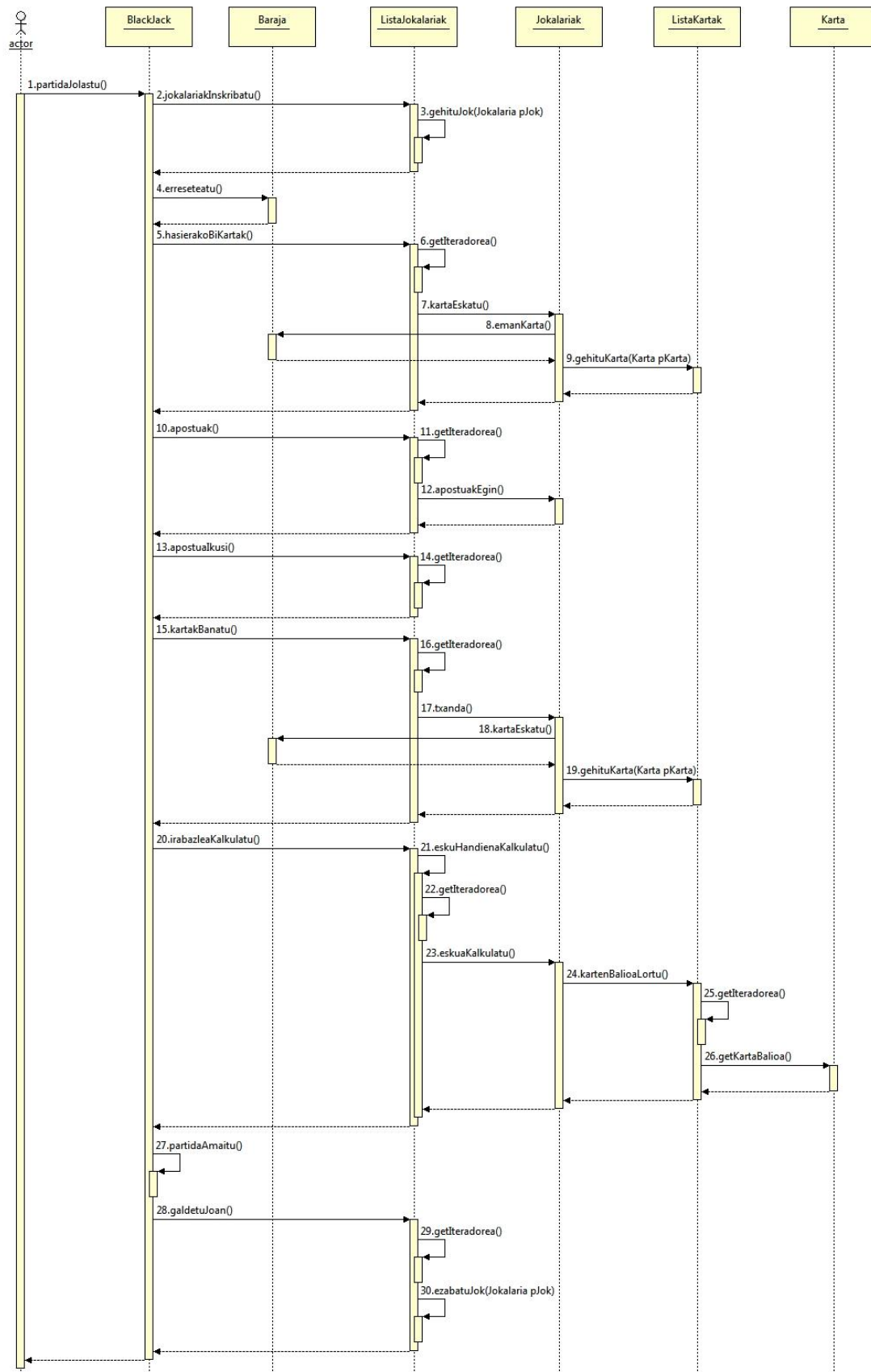
Klase-diagrama:



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.
- 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.

## Sekuentzia-diagrama:



Junit-etan hainbat kasu larri probatu beharko dira:

<ul style="list-style-type: none"><li>Jokalari batek duen diru kopurua baino handiago bat apostatu nahi badu, kasu larri horretaz informatzeko mezua pantailaratuko da, eta beste kopuru bat apostatzeko aukera emango zaio.</li></ul>
<ul style="list-style-type: none"><li>Jokalari batek hasierako bi kartak jaso ostean, beste karta bat eskatzen badu eta haren eskuren balioa 21 baino handiagoa bada, pertsona hori jokoz kanpo geratuko da eta egoera horretaz informatzen duen mezua pantailaratuko da.</li></ul>
<ul style="list-style-type: none"><li>Bateko bat barajatik jasotzean, hasierako balioa 11 izango da. Hala ere, batekoa duen jokalaria eskua 21 baino handiagoa bada, automatikoki batekoaren balioa 11-tik 1-era pasatuko da.</li></ul>
<ul style="list-style-type: none"><li>Partida bat hasteko, 2 jokalarik egon behar dira gutxienez, bestela “Beste jokalarik bati itxaroten...” mezua agertuko da. Era berean, jokalarikopuru maximoa 7 izango da, eta beste jokalarik batek partidaren parte hartu nahiko balu, ez duela adierazten dion mezua pantailaratuko da (<math>2 \leq \text{Jokalarikop.} \leq 7</math>)</li></ul>



## **Planifikazioa**

GitHub proiektu maneilatzailea erabiliko dugunez, gure planifikazioa asko sinplifikatuko da, bakoitzak egindako lanak besteek kontrolatu ahal izango dutelako edozein momentuan, baita akatsak zuzendu ere. Horretarako, TeamSpeak, TeamViewer eta SourceTree programak erabiliko ditugu lana guztion artean egiteko. Dena den, Aste Santuan inplementazioa amaitzea espero dugu, nahiz eta hainbat errore sortu.

Pentsatu dugu planifikazio metodorik hoberena, lan zati bakoitzean denon arteko kolaborazioaren bidez gure helburuak modu eragingarriago batean egitea lortzea da.