

Programozás alapjai 3

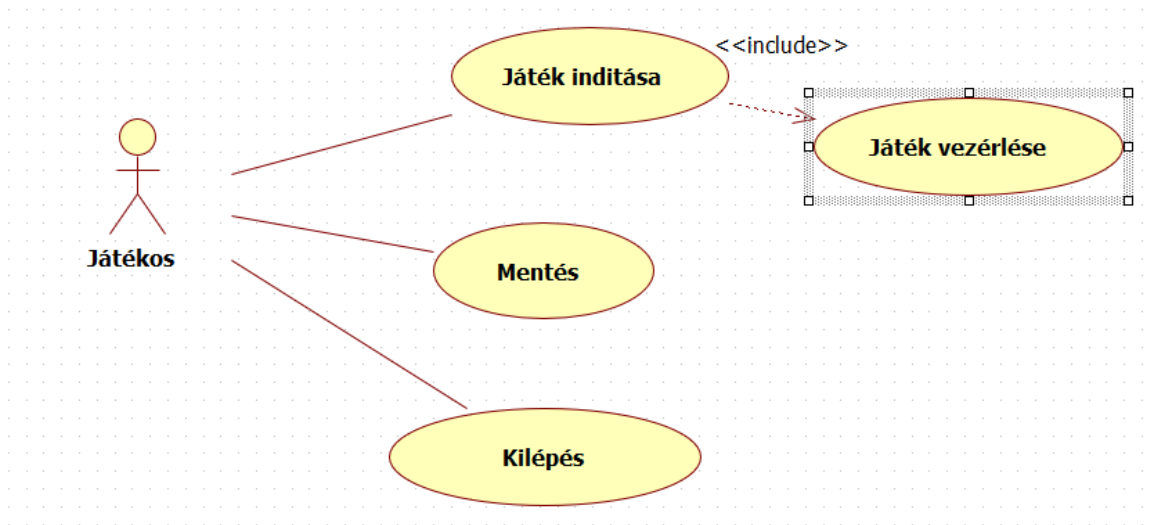
Specifikáció

Black Jack játék

Feladat leírás:

A Black Jack egy kártyajáték, amelyet francia kártyával kell játszani. A játék lényege hogy a kártyák értéke, amit osztanak, 21 értékű legyen vagy nagyobb legyen az összeg az osztónál lévő kártyák összegétől. A kártyáknak különböző értékük van. A számokkal jelölt kártyák értéke megfelel a rajta lévő számmal. A magas lapok, mint a Király, Bubi, Dáma értéke 10 és az Ász értéke 11 vagy 1 attól függően, hogy a lapok összege meghaladja-e a 21 akkor az értéke 1-re változik minden más esetben 11 ér. A játék úgy kezdődik, hogy a játékos megteszi a tétjét és ezt követően az osztó oszt két kártyát a játékosnak és két lapot magának amiből az egyik le van fordítva, hogy annak értékét ne lehessen látni. A második körben az osztó megkérdezi a játékos, hogy szeretne-e egy új kártyát húzni a többihez ezzel növelve a kártyák összegét. Ha a játékos kér egy új lapot és nem lett több az összeg 21-nél, akkor az osztó megismétli a kérdését. Abban az esetben, ha a játékos megáll, azaz nem kér több lapot akkor az osztó köre következik. Az osztó felfordítja az a lapot, amit az első körben kapott és addig húz lapokat, míg a játékosnál nem nagyobb az ő összege vagy „besokkal” (a kártyák összege több mint 21) viszont a osztónak van egy olyan feltétele is, hogy 16 alatt nem állhat meg, szóval ha a játékosnak 15 az összeg és az osztónak is 15 akkor az osztónak húznia kell még egy lapot. Ha a játékos vesztett, akkor a tét, amit a játék elején feltett az osztó nyerte el, ha viszont a játékos nyer, akkor a tét kétszeresét nyeri vissza. Például ha a játékos 10\$ tett fel és megnyeri a játékot akkor 20\$ nyert a játékos. Van egy speciális eset a nyerésre, amikor a játékos az első körben 21-et kap, azaz egy magas lapot vagy tízes lapot és mellé egy Ász kap ilyenkor a tét 1,5 szeresét nyeri meg. Itt is egy példával élve, ha a tét 10\$ és az első körben „Black Jacked” van (21 a kártyák összege) akkor a nyereményed, amit visszakapsz 25\$ lesz. Ez a játékot addig lehet folytatni míg, van tét, amit feltudsz, rakni vagy meg nem unod ☺.

UseCase:



1. Játékos: A felhasználó, aki játszani fog a játékkal.
2. Játék indítása: Ez egy menüpont lesz a programban. Erre kattintva a játékos elkezdheti a játékot
- 2.1 Játék vezérlése: Itt történik meg a tényleges játék a programon belül
3. Mentés: A játékos elmentheti a játékban megszerzett pénz összegét, ami a következő indításkor majd betöltődik.
4. Kilépés: A program bezárására alkalmas menüpont.

Megoldási vázlat:

A programhoz a mentési folyamatokhoz „Serializable” fogok használni ez egy állományba, történí mentés és majd ebből az állományból foggom az adatokat betölteni. A Players fogja megvalósítani mivel itt, van tárolva az játékos pénz összege, amit a játék alatt fog szerezni. A grafikai megvalósításhoz JavaFX-et foggom használni

Az osztályok leírása:

1. class Cards

A Cards osztály valósítja meg a kártya objektumokat azok véletlenszerű húzását és az kártya értékét.

Attribútumok:

Private int[] cardsnum	A kártyákat számmal azonosítja pl.: Király = 13
Private String[] cardscolor	A kártyák színeit azonosítja a képek megjelenítéséhez
Private int[] card	Ebben tárolódik a kártya száma és színének száma melyet az előző két tömb azonosít

Metódusok:

public Cards()	A konstruktor két random számot generál és azokat eltárolja a card tömbben
public String getCard()	Visszaadja a kártya nevét
public int getCardValue()	A kártya értékét adja vissza (de nem a számolás hoz megfelelő értékét)

2. class Game

A játék irányításáért felelős osztály

Attribútumok:

private boolean startGame	A játék indításáért felelős kontroller (csak konzolos változatnál használtam)
private boolean botround	Ha a játékos Stay-t nyom akkor az értéke true-ra változik és a AI fog tovább játszani
private int rounds	Meg adja, hogy hányadik körnél jár a játékos

Metódusok:

public void Start(String com)	Konzolos változat elindítása
public boolean getStartGame()	Visszatér a startGame értékével
public void Hit()	A játékos húz egy kártyát
public void Busted(Players player)	Megvizsgálja, hogy az adott játékos „Busted” azaz, hogy a kártyáinak az értéke több mint 21
public int HowWin()	Vissza adja, hogy kinyerte a játékot ha -1 akkor a bank nyert, ha 0 akkor döntetlen a játék, ha pedig 1 akkor a játékos nyert

public int getRounds()	Visszaadja a körök számát
public void resetRounds()	Visszaállítja a körök számát az elsőre

3. class Players implements Serializable

A játékban a játékos valósítja meg itt tárolódig a játékos pénze, kártyái és hogy mennyi pénz tet fel az adott körben. Ez az az osztály, amely mentésre kerül, hogy a megszerzett pénze ne vesszen el.

Attribútumok:

private String name;	A játékos azonosítására alkalmaz
private int cash	Ebben tárolódig a játékos pénze, amit a játék alatt megszerzett
public ArrayList<Cards> yourcards	A kártyák, amelyek felhúzott
private int bet;	Mennyi tétet rakott fel
protected boolean busted	Ez jelzi ha egy játékos besokalt

Metódusok:

public Players()	Lefoglalja a ArrayList területét
public int SumCards()	Összeszámolja, hogy a játékosnál vélő kártyák mennyit érnek
public void DropCards()	Eldobja a játékosnál lévő összes kártyát
public void addCard(Cards c)	A felhúzott kártyát a többihez rakja az ArrayList-ben
public void setBet(int ybet)	Beállítja a játékos tétjét és az le is vonja az összegéből
public void setDoubleBet()	A duplázáshoz állítja be a tétet és ugyan úgy levonja a játékos összegéből
public void getYourcards()	Konzolos változatnál írja ki a játékos kártyáit
public boolean isBusted()	Visszaadja a játékos busted értékét
public void setBusted(boolean busted)	Beállítja a játékos busted értékét
public void resetCash()	A kezdő értékre állítja a játékos összegét (500\$)
public int getCash()	A játékos összegét adja vissza
public void setCash(int c)	A nyereményt adja hozzá az összeghez
public int getBet()	Visszaadja a játékos tétjét

4. class PlayerAI extends Players

A mesterséges intelligenciát valósítja meg. Ez az az osztály, aki a játékos ellen fog majd játszani.

Attribútumok:

private boolean finish	Ez jelzi, hogy a AI befejezte a játékot
------------------------	---

Metódusok:

public void AIturn()	Itt a gép addig húz kártyákat míg nem lesz több mint a játékosnak vagy be nem sokkal
public boolean isFinish()	Ha befejezte a játékot a gép akkor true-val tér vissza
public int FirstCardValue()	Az első kártyának az értékét adja meg ami még a játékos láthat
public String FirstCard()	Vissza adja a első kártyának a nevét

5. class Main extends Application

A main osztály ami elindítja az alkalmazást.

Attribútumok:

static public Game game	A játékvezérlő
static public Players player1	Az egyik játékos
static public PlayerAI botbank	A gép aki a játékos ellen játszik

Metódusok:

public static void main(String[] args)	A main függvény
public void start(Stage primaryStage) throws Exception	Megjeleníti az ablakot

6. class Message

Az értesítéseket megjelenítő osztály.

Attribútumok:

-

Metódusok:

public void Mess(String t)	A „t” üzenetet ír ki az ablakba
----------------------------	---------------------------------

6. class Controller

Az ablak irányításáért felelős osztály.

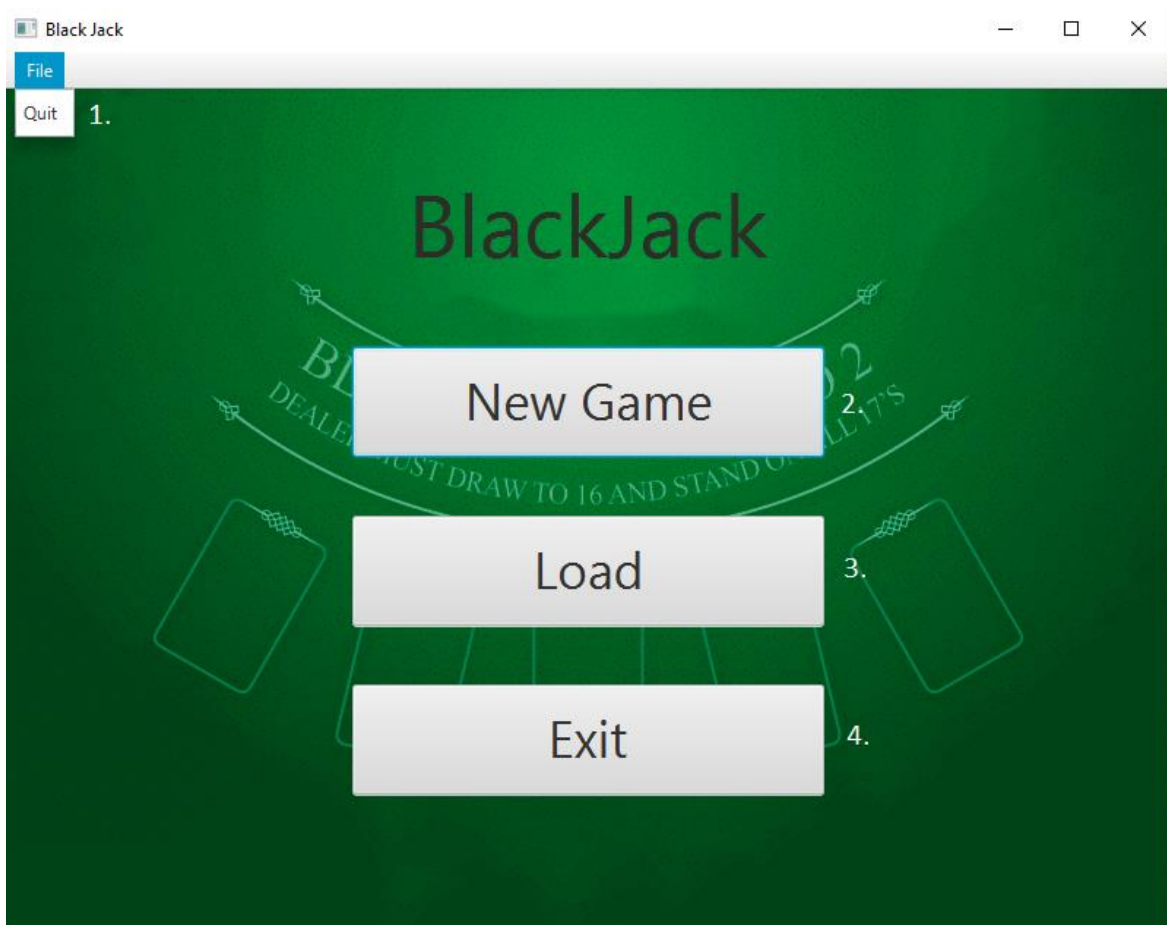
Attribútumok:

private Message m	Az értesítésekhez használt objektum
private boolean isScene2	Jelzi, hogy melyik scene akti jelenleg ha fales akkor a scene1 ha true akkor a scene2

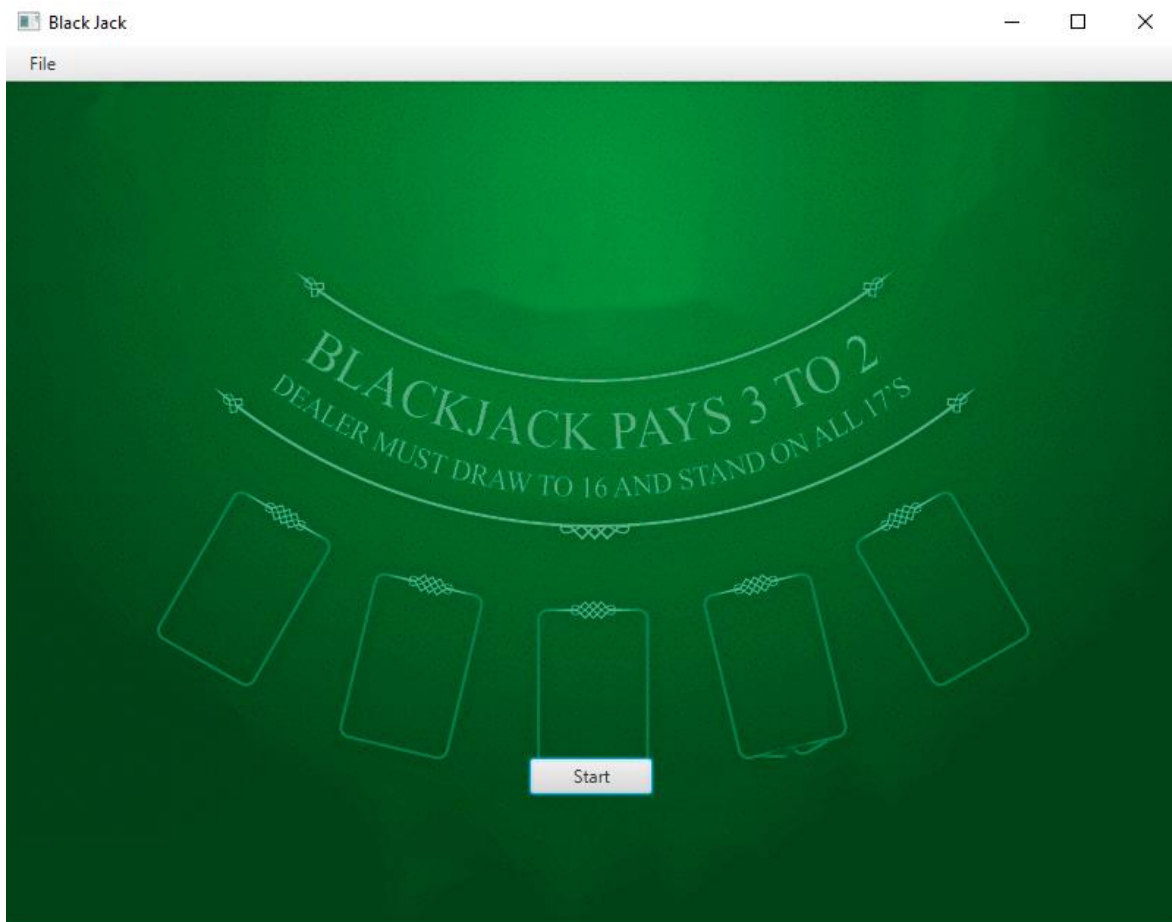
Metódusok:

public void TakeBet()	Az ablakon a bet gomb vezérlője
public void Hit()	A hit gomb vezérlője
public void DoubleHit()	A double gomb vezérlője
public void Stay()	A stay gomb vezérlője
public void BlackJack()	Ha a játékosnak az első körben 21 van, akkor a játékos automatikusan nyer
public void NewGameB()	A new game gomb vezérlője
public void LoadB()	A load gomb vezérlője. Betölti a játékos mentését
public void SaveB()	A save gomb vezérlője. El menti a játékos összegét a savedata.ser fájlba
public void Exit()	Kilép a programból
public void NewGame() throws Exception	Elindítja az új játékot és betölti a játék felületet
public void Load() throws Exception	Betölti a játékos mentését és a játék felületet
public void GetStart()	Elindítja a játékot

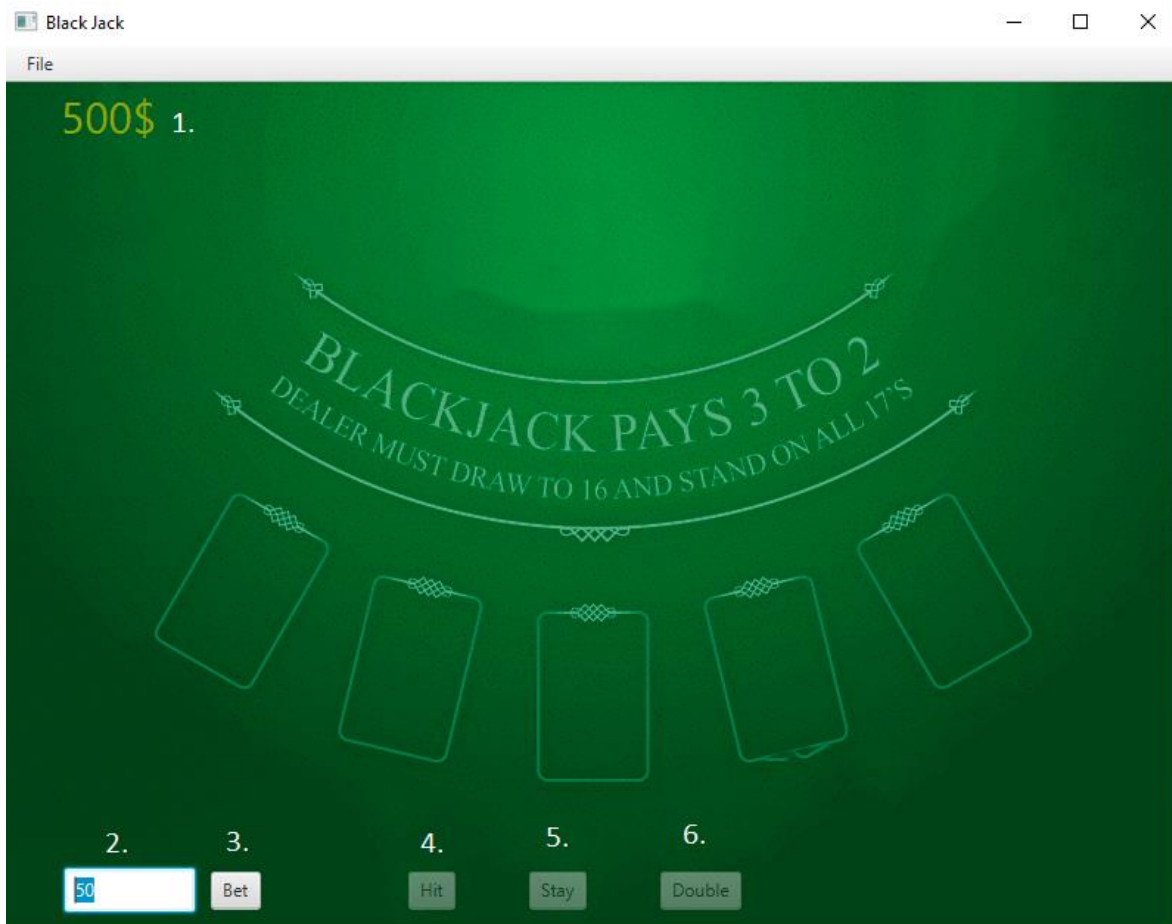
Felhasználói kézikönyv:



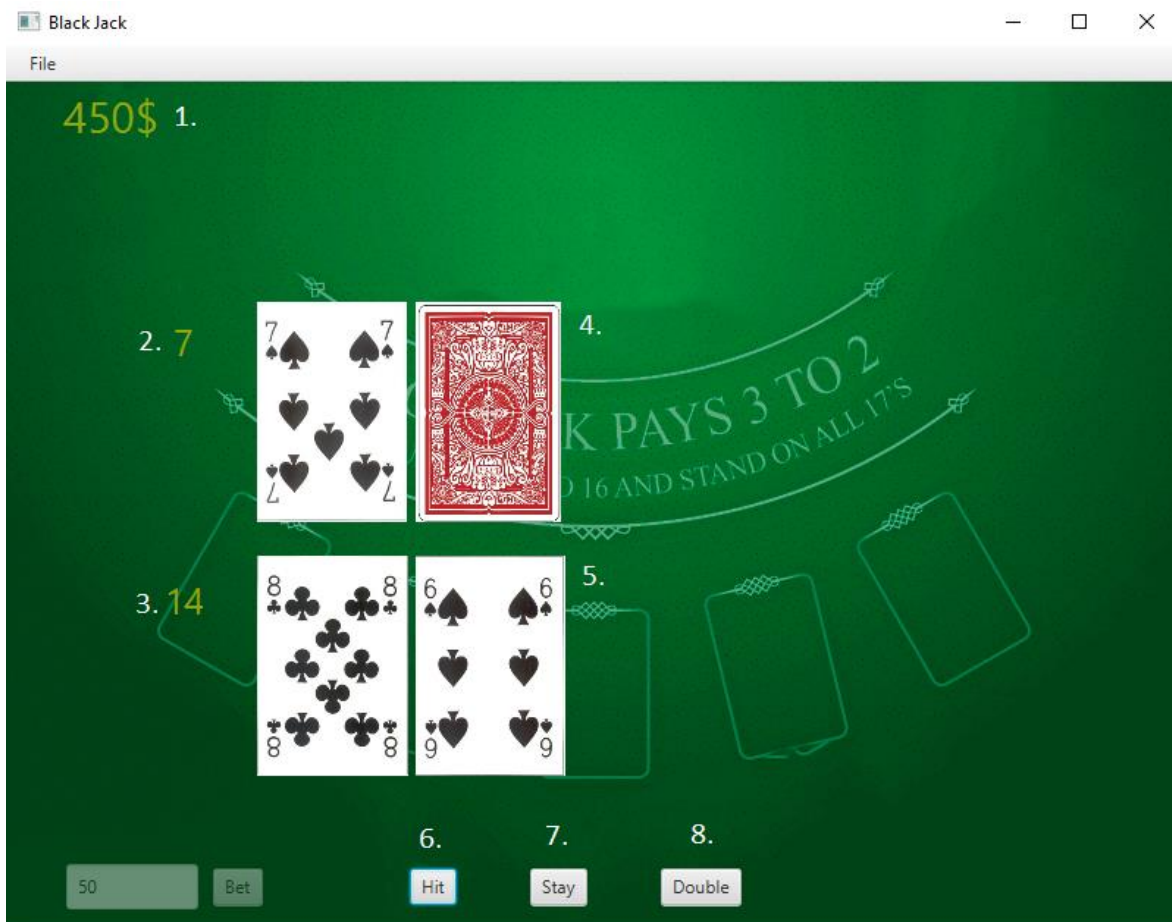
A játék ezzel az ablakkal fog elindulni. A 1. jelzésű gombbal be lehet zárni az játékot, de előtte egy felugró ablakon meg kell erősíteni, hogy tényleg be akarod zárni a játékot. A 2. gombra nyomva egy új játékot lehet kezdeni ilyenkor a játék, betölt és a játékos 500\$-ra kezdi el a játékot. A 3. gombbal a játékos visszatöltheti a régebben elmentet összegét és onnan tudja folytatni a játékot. A 4. gomb ugyan azt csinálja, mint az első gomb.



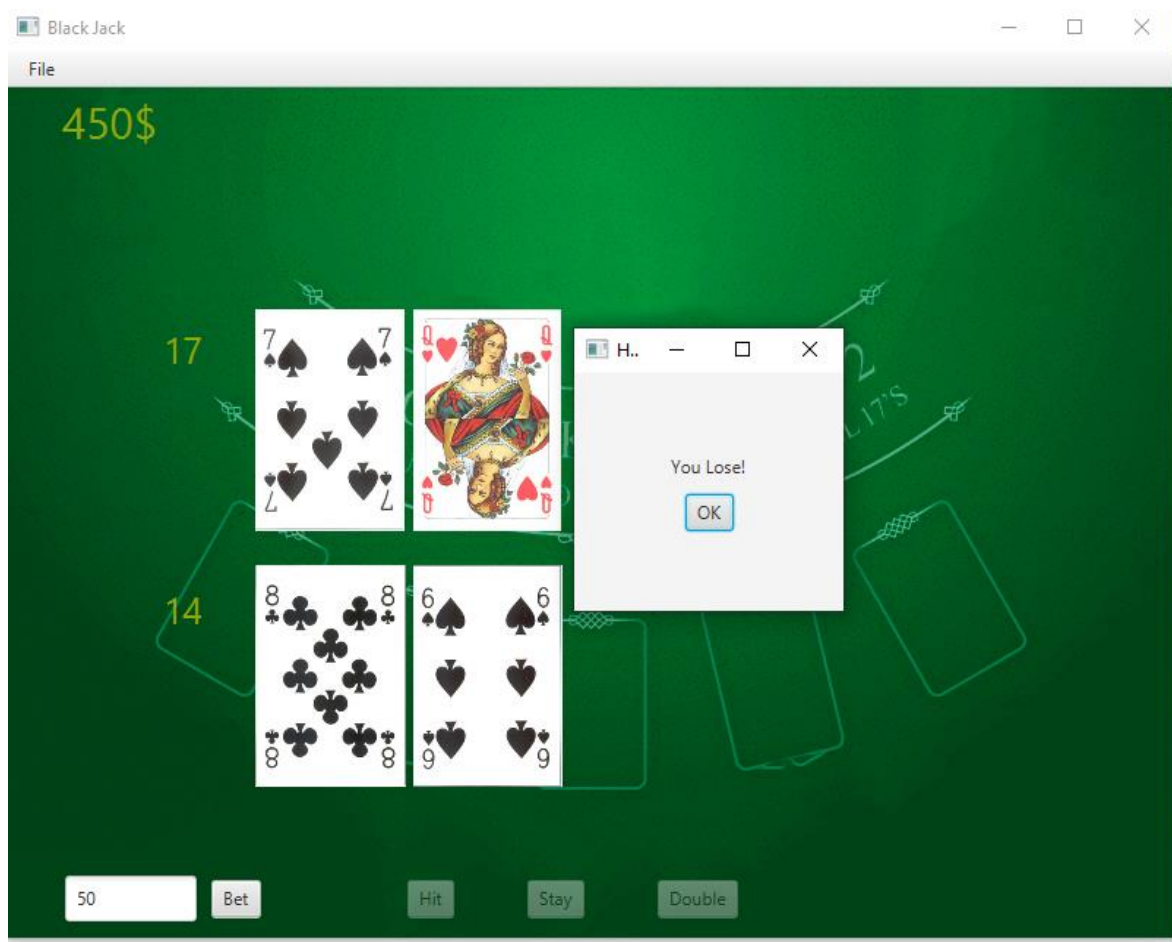
Ha a NewGame vagy a Load gombra nyomtunk egy ilyen ablak fog minket várni itt a start gombra nyomva a játék elkezdődhet.



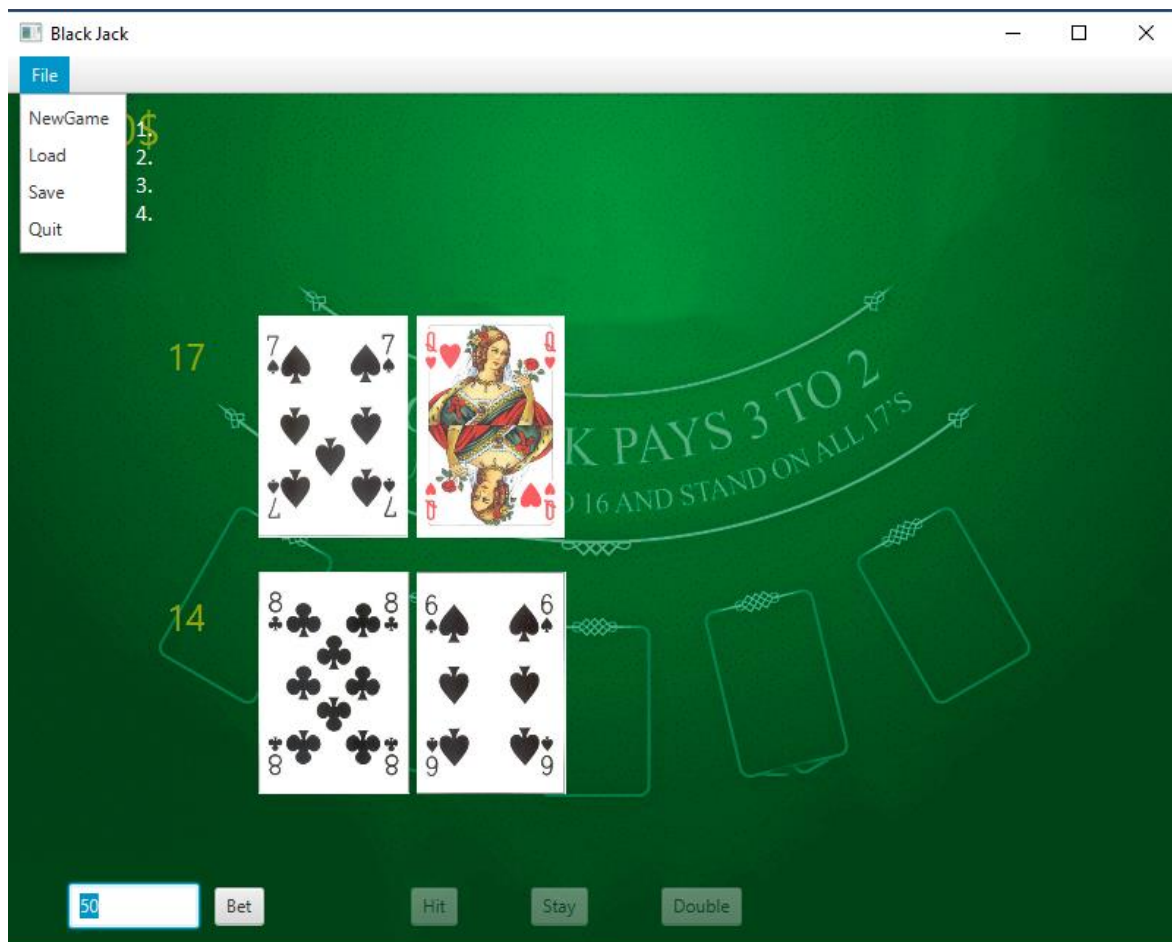
Az 1. jelölt szám az az összeg, amivel elkezdted a játékot. A 2. jelölt szövegdoz az ahol a tétet be tudod állítani. A tét nem lehet negatív szám és nem lehet több mint amennyi az összeged, ha még is nagyobb összeget adsz meg akkor az összes pénzed felrakod. A 3. gombbal lehet a tétet megerősíteni. A 4. , 5. , 6. gombok csak az után lesznek elérhetőek miután megtetted a tétet.



Miután megerősítette a tétet az összeg levonódik a pénzedből ez a 1. jelzésű számnál lehet látni. A 2. a gép által felhúzott kártya értéke, de csak az első kártyáját láthatod 4. a lapok, amiket húzott. A 3. a te lapjaidnak az összege és az 5. a te lapjaid. A 6. gombbal tudsz új lapot kérni, ami növeli a húzott kártyák összegét, ha több mint 21 akkor besokkaltál és vesztetél. A 7. gombra nyomva megáll és át adod a kört a gépnek. A 8. gombbal pedig húzol egyet és megduplázod a tétet de azt követően a gép jön.



Miután a gép is befejezte a játékot a játék kiírja, hogy nyertél vagy vesztetél. Ha nyertél, akkor a téted dupláját kapod vissza. Ha az ok gombra kattintva folytathatod a játékot.



Az egyes gombbal új játékot lehet kezdeni miután az megerősítetted. A 2. gombbal betölthetted egy régebi mentésedet. A 3. gombbal a pénzüsszegedet mentheted le. A 4. gombbal ki léphetsz a játékból miután az megerősítetted.