Kártya és kártyajáték leírás Heartstone/Yugioh szerű párbaj játékhoz. Két játékos a saját összeállított paklijával megpróbálja az ellenfelének életerejét nullára csökkenteni.

KÁRTYA

*//Kártya koncepció a párbaj játékhoz//*

**Card Creature**

id : int *//ID egyedi azonosításhoz*

Name : string *//név játékon belüli azonosításhoz*

Cost : int *//költség, ennyi nyersanyagot veszít a játékos a kártya kijátszásakor*

Attack : int *//támadás érték, ennyivel csökkenti az ellenfél életerejét*

Health : int *//életerő, ha nullára csökken, a kártya kikerül a játékból*

available : boolean *//elérhetőség, ha a kártya a körben nem támadott, akkor igaz értékű*

ALAPELEMEK

*//A játék elemei*

**Game BattleGame** *//a játék*

**Deck JimDeck**  *//Jim saját összeállított paklija*

**Deck BobDeck** *//Bob saját összeállított paklija*

*//Hand: a játékos kezében levő kártyák halmaza*

*//Board: a játékos táblája, ide kerülnek az általa kijátszott kártyák*

*//Drawpile: a játékos húzópaklija*

*//Discardpile: a játékos játékból kikerült lapjai*

*//Health: a játékos életerő tokenje*

*//Mana: a játékos nyersanyag tokenje*

*//ActivePlayer: az aktív játékost jelző token, 1 ha aktív, 0 ha nem*

**Player Jim: Hand, Board, Drawpile, Discardpile, Health, Mana, ActivePlayer**

**Player Bob: Hand, Board, Drawpile, Discardpile, Health, Mana, ActivePlayer**

FÁZISOK:

*//A játék előkészítő fázisa, a változók inicializálása*

**Phase Setup**

JimDrawpile = JimDeck *//Jim húzópaklija az általa összeállított pakli*

BobDrawpile = BobDeck *//Bob húzópaklija az általa összeállított pakli*

*//minden játékos kap adott életet és manát, valamint húz kártyát a húzópaklijából a kezébe*

foreach Player

Health = 20

Mana = 3

Draw 5

*//sorsolással eldől, hogy ki az aktív játékos, azaz ki kezdi a játékot*

*//neki az ActivePlayer tokenje beállítódik, illetve az activeplayer változó eltárolja a játékost*

activeplayer = random Player

activeplayer.ActivePlayer = 1

*//A játék harc fázisa, azaz a játékosok közötti interakció*

**Phase Battle**

*//addig megy a játék, amíg valamelyik játékos életereje nullára nem csökken*

while foreach Health > 0

*//a kör fázisokra van osztva, melyeken sorrendben végigmegy az aktív játékos*

*//ez a játékmenet loopolva van, az aktív játékos kiléte egy ciklus végén változik*

loop activeplayer TurnStart

activeplayer Preparation

activeplayer Offense

activeplayer End

*//kör kezdete fázis*

*//a játékos húz egy lapot és kap két manát*

**Phase TurnStart**

Draw 1

Mana += 2

*//előkészítő fázis, a játékos kijátszhat lapokat a kezéből amennyiben manájából ki tudja fizetni az adott lap költségét*

**Phase Preparation**

optional Summon

*//támadás fázis, a játékos a tábláján lévő, már kijátszott lapokkal támadhat, amennyiben az adott kártya még nem támadott a körben – az available értéke 1*

**Phase Offense**

optional AttackCard

optional AttackPlayer

*//a kör vége fázis*

*//a játékos kártyái ismét elérhetőek lesznek a következő körre, valamint az aktív játékos kiléte változik*

**Phase End**

SetAvailable

SwitchActivePlayer

AKCIÓK

*//a játékos húz egy lapot a húzópaklijából a kezébe*

**Action Draw**

move Card from Drawpile to Hand

*//a játékos kiválaszt egy kártyát amit kijátszik a kezéből*

*//amennyiben a lap költségét ki tudja fizetni a manakészletéből, a lap kijátszásra kerül*

**Action Summon**

pickedcard = <user input>

if Mana > pickedcard.cost

move pickedcard from Hand to Board

Mana -= pickedcard.cost

*//kártya támadása kártyával*

**Action AttackCard**

attacker = <user input>

target = <user input>

if attacker.available == 1 *//elérhetőség ellenőrzése*

target.Health -= attacker.Attack *//célpont életerejének csökkentése*

attacker.available = false *//támadó kártya elérhetőségének átállítása*

*//kártya életereje 1 alá csökken - meghal*

if target.Health <1

move target from Board to Discardpile

//ellenfél játékos támadása

**Action AttackPlayer**

attacker = <user input>

if attacker.available == 1 *//elérhetőség ellenőrzése*

Health -= attacker.Attack *//ellenfél életerejének csökkentése*

attacker.available = false *//támadó kártya elérhetőségének átállítása*

*//a játékosok ActivePlayer tokenjének értéke az ellenkezőjére fordul*

**Action SwitchActivePlayer**

foreach Player

if ActivePlayer == 1

ActivePlayer = 0

else ActivePlayer = 1

*//A játékos összes, a már kijátszott, azaz a Board pakliban levő kártyájának elérhetőség változója igaz értéket vesz fel*

**Action SetAvailable**

foreach Card in Board

available = true