**Game Uno**

**Deck UnoDeck**

**Player Alice: AliceHand, AliceScore**

**Player Bob: BobHand, BobScore**

**Player Cecil: CecilHand, CecilScore**

**Player Board: BoardDeck**

**Player Draw: Drawpile**

**Action PlaceCard**

move currentPlayer.pickCard to BoardDeck

**Action getNextPlayer**

if currentplayer == Alice

currentPlayer = Bob

if currentPlayer = Bob

currentPlayer = Cecil

else currentPlayer = Alice

**Action DrawCard**

move Drawpile.lastCard to currentPlayer.getHand

**Action Reshuffle**

while BoardDeck isNotEmpty

move BoardDeck.lastCard to Drawpile

<amíg az aktuális játékosnak üres a keze, húz 7-et

ennek minden játékos esetében egyszer kell megtörténnie>

**Phase Setup**

Drawpile = UnoDeck

currentPlayer = Alice

while currentPlayer.Hand isEmpty

DrawCard 7

currentPlayer = getNextPlayer

<amíg valakinek el nem fogynak a lapjai, addig

az aktuális játékos rak egyet a kezéből az asztalra

és a következő játékos kerül sorra>

**Phase MainGame**

while currentPlayer.Hand isNotEmpty

if BoardDeck.lastCard == DrawFour

DrawCard 4

currentPlayer = getNextPlayer

if BoardDeck.lastCard.value == SkipTurn

currentPlayer = getNextPlayer

move currentPlayer.pickCard to BoardDeck or DrawCard 1

currentPlayer = getNextPlayer

<átmeneti „kijátszható kártyák” pakli létrehozása

ami megfelel a feltélteleknek, tehát az utolsó lerakott kártyával

azonos színű vagy értékű, vagy fekete (+4 vagy színváltós)

ezekből választhat a játékos>

**pickCard**

Deck availableCards

Card cardToCheck = currentPlayer.firstCard

while cardToCheck != currentPlayer.lastCard

if cardToCheck.colour == BoardDeck.lastCard.colour or

cardToCheck.value == BoardDeck.lastCard.value or

cardToCheck.colour == black

copy cardToCheck to availableCards

cardToCheck = currentPlayer.nextCard

pickedCard = availableCards.getCard

**Phase End**

do

currentPlayer.checkScore

currentPlayer = getNextPlayer

while currentPlayer.getScore == null

**checkScore**

int score

cardToAdd = currentPlayer.firstCard

if cardToAdd == DrawFour or cardToAdd.value == SkipTurn or cardToAdd.value == ColourChange or cardToAdd.value == reverse

score -= 10

else score -= cardToAdd.value

currentPlayer.setScore(score)