Mastermind Pseudocode

Game Menu

De speler krijgt een menu te zien waarin hij kan kiezen tussen de keuze om speler 1 of speler 2 te zijn.

Speler is speler 1

De speler genereert een code van 4 getallen.

De speler kan alleen kiezen tussen de getallen 1,2,3,4,5,6 en de code moet 4

Cijfers lang zijn.

De computer voert het algoritme uit tot deze de code heeft gevonden.

Speler is speler 2

De computer genereerd een random getal code die de getallen 1,2,3,4,5,6 op willekeurige volgorde zet en die ook maar 4 cijfers lang is.

De speler gokt deze 4 getallen tot hij het goed heeft.

De speler mag maximaal 8 keer gokken.

De computer geeft feedback op elke gok van de speler.

Import Random

def game\_menu():

“Geeft een klein menu waarin de speler aan kan geven of hij speler 1 of speler 2 wilt zijn.”

#Speler is speler 1:

def kies\_code():

“Via een input kan de speler een code invoeren. Deze moet 4 karakters lang zijn en mag alleen bestaan uit 1,2,3,4,5,6. Als dit niet zo is zal de computer foutmelding geven en moet je een andere geheime code kiezen.

Return code”

def computer\_algoritme():

“Volgens het algorithme dat word gebruikt in het artikel uit de universiteit van Groningen, voert de computer een gok uit op basis van de feedback\_speler.”

def feedback\_speler():

“De speler geeft feedback aan de computer. ik ga hiervoor de techniek gebruiken die ook in het artikel van de universiteit van Groningen werd gebruikt.

Wanneer een getal wel goed is maar op de verkeerde plek returned hij een [0,1], wanneer er een getal goed staat en de juiste is returned hij een [1,0]

Deze zijn cumulatief, dus wanneer er bijvoorbeeld 2 goed zijn op de goeie plek en een goed maar op de verkeerde plek returned hij [2,1].

Return feedback”

#Speler is speler 2:

def maak\_code():

“Maak een code met 4 cijfers kiezend uit 1,2,3,4,5,6 door middel van random

code = ”

“return code”

def input():

“Vraag input en geef deze een naam, bijv “gok”. Als deze iets anders is dan een 4 karakter lange combinatie van 1,2,3,4,5,6 geef een foutmelding en vraag een nieuwe input.”

def check():

“In deze functie vergelijkt de computer de gok met de de gegenereerde code. Als deze overeenkomen, feliciteert de computer de speler.

Break

Als deze niet overeenkomt met de code word feedback\_computer aangeroepen.”

def feedback\_computer():

“De computer geeft hier feedback op de gokken van de speler, ik ga hiervoor de techniek gebruiken die ook in het artikel van de universiteit van Groningen werd gebruikt.

Wanneer een getal wel goed is maar op de verkeerde plek returned hij een [0,1], wanneer er een getal goed staat en de juiste is returned hij een [1,0]

Deze zijn cumulatief, dus wanneer er bijvoorbeeld 2 goed zijn op de goeie plek en een goed maar op de verkeerde plek returned hij [2,1].

Return feedback.

Roep input opnieuw aan.”