

Ugeopgave 10g

PoP

Benjamin, Mads og Thawatchai
Datalogisk Institut, Københavns Universitet

20. december 2017

SimpleJack

Regular expression: Vi har tilføjet en del regular expressions i starten, da det er vigtigt at programmet ikke fejler, når brugeren kommer til at taste nogle bogstaver forkert. Siden vi har en del `console.ReadLine` hvorpå brugerens input bliver brugt, har vi lavet 7 regular expressions. Alle regular expressions, på nær 2, er hvor brugeren kan skrive hele sætninger eller lign. hvorpå keywordet skal være tilstede, for at godkendes. Der blev lavet 2 regular expressions hvor brugeren har mulighed for at skrive med stort eller småt i forhold til valg af strategi i AI.

String of Cards: Vi har defineret hele vores dæk af kort som strings, som er opdelt i 4 forskellige kategorier - clubs, diamonds, hearts and spades - hvorpå alle kort er beskrevet som en string. Dermed er det nemmere for os at ændre de enkelte strenge, hvorpå den kan have global effekt, hvis det ønskes senere.

How many players: Når programmet starter bliver `amountofPlayers` sat til at være lig 0. Når programmet har mindre end 1 spiller, vil der aktiveres en `console.WriteLine` som spørger hvor mange spillere der skal spille spillet. En `console.ReadLine` aktiveres og matches med brugerens input. Denne match har en regular expression tilknyttet, så brugeren kan skrive et tal mellem 1-5, hvorved denne string bliver konverteret til en integer. Hvis brugeren skriver noget over eller under 1-5, vil det prompte en string "That amount is not supported". Bliver der ikke skrevet noget tal eller lign. vil det prompte en error med string "Quitting".

Player drawing cards: Denne sektion tjekker om `player ≤ amountofPlayers`, hvis dette er tilfældet så vil sektionen køre, hvorpå spilleren får sine kort. Der bliver kaldt til `DrawHand` klassen hvorpå vi også har en funktion til både `hand.FirstCard` og `hand.SecondCard`. `playerVal` værdien (mutable) bliver sammensat af `hand.FirstCard` og `hand.SecondCard`, men hvor de kaldt sammen med `RealValueConverter` som er en funktion der tager regular expression i forhold til vores `String of Cards` så den konverterer strengene og giver dem en værdi. De værdier bliver outputtet sammen med `player`. En `if` statement tjekker om der er 2 aces der bliver trukket, ved hjælp af regular expression, hvis dette sker, vil det ene kun have en værdi af 1.

Hit, Stand or AI I vores **Hit, Stand and AI** bliver de også matchet med regular expression, så programmet ikke crasher af forkert bruger input. **Hit** fungerer således at brugeren får et kort, bliver kaldt via **DrawCard** klassen. Der bliver outputtet **playerVal** efter hvert trukket kort. Kommer **playerVal** over 21, bliver player bust.

Stand virker sådan at spilleren står med deres nuværende værdi.

AI giver spilleren 2 mulige strategier **A** eller **B**, som brugeren kan vælge. Strategi **A** følger samme regel som **Dealer** hvor AI står ved 17, ellers alt under 17, trækker AI et kort. Strategi **B** fungerer sådan at AI tilfældigt trækker et kort, eller står ved nuværende værdi.

Player Results En for-løkke kører for at outputte hvad de enkelte spillere fik i deres slutværdi. Denne løkke kører til værdien **amountofPlayers**.

Dealer Dealeren i dette spil har samme strategi som **AI option A**. Funktionerne der blev brugt her er de samme, dog med en **dealerVal** i stedet for **playerVal**.

Result Screen Under **Result** kører en for-løkke, samme funktionalitet som hos **Player Results**, hvorpå den viser hvad resultatet af de enkelte spillere i forhold til **Dealer**.

Classes

Random number generator and cacheCheck Vi laver en random number generator tilknyttet funktionen **rnd**. Denne bliver brugt til at genere vores kort. Vi laver desuden en funktion, **cards**, som laver en liste [1..52]. Der bliver oprettet en **cacheCheck** hvorpå den tjekker det generede kort allerede er blevet generet eller ikke. Hvis dette er tilfældet, springes denne værdi over og næste kort i rækken trækkes.

DrawHand Denne klasse generer vores 2 første kort. Dette gøres ved hjælp af tidligere funktion **rnd**. **DrawHand** har **card1** og **card2** som bliver tilknyttet til member **FirstHand** og **SecondHand**, som er vores 2 første kort i spillet.

DrawCard Denne klasse generer vores næste kort, når spilleren, AI eller Dealer vælger **Hit**. Den bruger også funktionen **rnd** til at genere værdi.

RealValueConverter Denne funktion kører en vigtig regular expression som tager alle de strenge som er blevet defineret, og giver dem en værdi.