

Presentatie Probleemstelling

Wat is het probleem?

Ons doel bestaat erin een didactisch programma te maken dat op een visuele manier combinatorische problemen kan oplossen. Combinatieleer omvat de tak van de wiskunde die de mogelijke verdelingen telt van een groep objecten. Bijvoorbeeld: op hoeveel manieren kan een pak speelkaarten geschut worden? Waarop het antwoord $52!$ is. Het is een belangrijk deel van het middelbaar onderwijs en wordt vaak gezien als een van de kenmerken van menselijke intelligentie. Mensen kunnen dergelijke problemen namelijk gemakkelijk oplossen aangezien ze de achterliggende structuren herkennen, waarvoor eenduidige wiskundige formules bestaan. Voor artificiële intelligentie is dit echter niet vanzelfsprekend.

Slechts recent is een techniek binnen de AI, genaamd “Lifted Reasoning” toegepast op de combinatoriek, met een positief resultaat. De zogenaamde Combinatorics Solver, of CoSo, maakt gebruik van een declaratieve taal genaamd CoLa en is in staat om sneller telproblemen op te lossen dan ooit tevoren. Dit doet hij door de problemen op te delen in subproblemen, vergelijkbaar met hoe mensen redeneren. Juist hierom is het een logische aanvulling om voor de solver een visualisatiesoftware te schrijven die mensen in staat stelt om op een intuïtieve manier telproblemen te begrijpen.

Waarom is het belangrijk?

Zoals eerder aangegeven, willen wij een didactisch programma maken. Het zal studenten met moeilijkheden op het vlak van combinatieleer vooruit helpen. Door gebruik te maken van de *CoSo solver*, kan het programma op een efficiënte manier een antwoord bieden op een ingevoerd telprobleem. Sterker nog, het geeft ook de strategie weer die het gebruikt heeft om tot deze oplossing te komen. Een student kan zo deze stappen volgen en leren hoe hij het probleem zelf oplost.

In feite bestaat er al een zeer gelijkaardig programma aan hetgeen dat wij willen bereiken, genaamd *SymboLab*. Dit is een veelgebruikt programma dat voor verschillende wiskunde problemen, uitgebreide stap-voor-stap oplossingen biedt. *SymboLab* is echter veel minder krachtig op het domein van combinatieleer dan wat wij ontwikkelen en kan daarom niet gezien worden als een vervanging. Wel bewijst het dat er vraag is naar dergelijke programma's en dat verantwoordt ons doel.

Hoe aanpakken?

Voor de implementatie van de visualisatiesoftware hebben wij gekozen voor de Game Engine *Godot*. *Godot* is een programma waar wij beide al jaren ervaring mee hebben, dus waar we gemakkelijk mee aan de slag kunnen. Het gebruikt een eigen programmeertaal genaamd *GDScript*, die verwant is aan Python, de taal waarin de *CoSo solver* geschreven is. Bovendien kunnen we met *Godot* gemakkelijk externe bestanden zoals de *CoSo solver* en andere *Python libraries* gebruiken.

Eerst zijn we begonnen aan de voorstelling van de domeinverzamelingen. Wij hebben ervoor gezorgd dat we de input van de *solver* kunnen doorgeven aan onze implementatie. Concreet kunnen we al de domeinverzameling voorstellen, op basis van een declaratie in CoLa. Voor deze voorstelling hebben we gebruikt gemaakt van *Area-proportional Venn Diagrams*. Zoals de naam aangeeft, zijn dit Venndiagrammen waarbij de verzamelingen en de doorsneden in verhouding staan met de domeingroottes. De wiskunde achter dit probleem behoorde echter niet tot ons hoofddoel en hebben we dan ook overgelaten aan een externe *Python library* genaamd *matplotlib-venn*. Onze volgende stap bestaat uit de voorstelling van simpele telproblemen en daarna kunnen overgaan naar telproblemen met *constraints*. Om de verschillende stappen in het oplossingsproces te verduidelijken, zal onze implementatie gebruik maken van visuele animatie, met bijvoorbeeld pijltjes tussen transities.

Wanneer de *CoSo solver* uitgebreid wordt om een groter arsenaal aan combinatieproblemen te ondersteunen, kan ons programma mee uitgebreid worden.