Plan van Aanpak

Groep 6

Namen:
Luuk Bouwman
Thijme de Bruijn
Jasper van Loon
Mahmoud Chebil

Inhoud

Probleemomschrijving en onderzoeksvraag

In dit project gaan we aan de slag met een agent based simulatie. We moeten onderzoeken wat het effect is van plurality voting. Plurality voting is een electie systeem waar ieder persoon maar op 1 kandidaats mag stemmen. We moeten dus agents simuleren die kunnen stemmen en dat doen met hun beste interesses in gedachten. Me moeten het probleem oplossen van dat we agenten simuleren die de meest logische keuzes maken voor hunzelf. Met een simulatie waarin agents de meest logische keuzes maken voor een electie is dan de vraag: Wat is de invloed van strategische stemmers?

Taken

Om de simulatie te realiseren zijn de volgende taken benodigd te vervullen.

- Het kiezen van een tool om de simulatie in te maken via het SFA model
- Het maken van een plan van aanpak
- Er is een environment waarin agents gesimuleerd kunnen worden
- Een environment moet de mogelijkheid geven dat agents kunnen stemmen op een partij
- Er moeten agents gecodeerd worden die ze de mogelijkheid geven om te kunnen stemmen
- Een variabele in partijen en agenten wat hun theoretische ‘ideologie’ is
- De simulatie moet electies houden waarin agenten kiezen op een partij met als doel de meest geliefde partij als winnaar te krijgen (of dat nou de partij is waarop ze hebben gestemd of niet)
- Voorgaande electies en hun resultaten kunnen worden opgeslagen.
- Partijen kunnen stoppen met het meedoen aan volgende electies.
- De simulatie moet een logica bevatten waarin agenten stemmen gebaseerd op factoren zoals beste partij voor hun, partij met kans om te winnen, motivatie om hun minst liefste partij niet te laten winnen. Elk van deze logica punten kunnen gezien worden als zijn eigen taak om erin te coderen.
- Het programma en code moet zo gemaakt worden dat aanpassing zoals het veranderen naar een ander stem systeem makkelijk toe te voegen is.
- Er is een GUI waarin belangrijke variabelen makkelijk aangepast kunnen worden. Waaronder het kunnen resetten van de simulatie met nieuwe variabelen.
- De code moet schoon en ordelijk zijn.

Benodigde modules

In onze simulatie zijn de volgde modules van belang

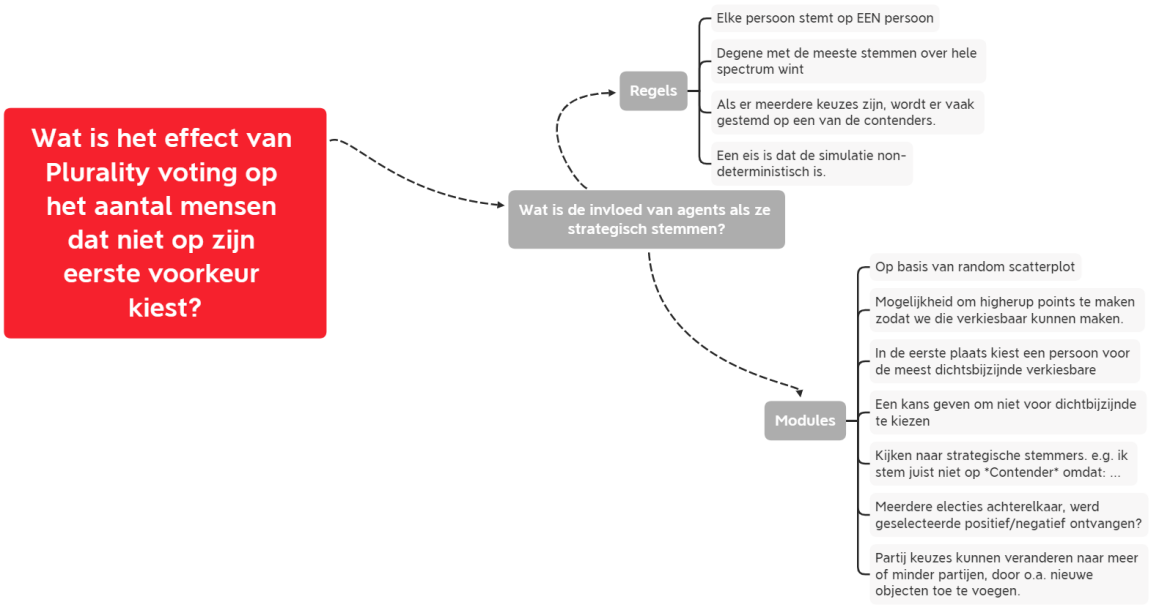
- De simulatie moet (via een GUI) kunnen worden herstart, mogelijk met nieuwe instellingen
- Agenten en hun keuze worden per electie geupdate, na deze update maken agents een nieuwe keuze voor de volgende electie.
- Een GUI om de voorbenoemde feature van herstarten met nieuwe instellingen mogelijk te maken en de simulatie te visualiseren.
- De resulaten en keuzes van agents kunnen worden opgeslagen en gebruikt worden in de toekomst.
- Opgeslagen resultaten zorgen voor een verandering in keuze bij agents en dus de uitkomst van een electie (h: s0 –a0-> s1 --a1-> s2 –a2-> s3 --a3-> su...)
- Partijen en agents hebben een theoretisch 'ideology' waarop keuzes worden gebaseerd

Project resultaten

Het resultaat van dit project bevat een simulatie dat met verschillende variables te werken is. Een environment met agents die visueel laten zien hoe ze stemmen op een partij. Een belangrijk resultaat is ook de beantwoording van de onderzoeksvraag wat het effect is van plurality voting en wat er gebeurd als agents strategisch stemmen. De voorgaand beschreven taken moeten gedaan worden, met de minimum van taken die van belang zijn om de onderzoeksvraag te kunnen beantwoorden en visualiseren. We hebben dus een GUI om de variabelen te kunnen aanpassen en de simulatie te visualiseren.

Aan het eind van het project kunnen we dus een documentatie inleveren van onze keuzes en verloop van het maken van de AMBS. Code dat de simulatie en zijn uitkomsten draait en een beantwoording van de onderzoeksvragen. Allemaal opgeleverd in GitHub.

Mindmap



Planning

		Begin week 1	Eind week 1	Begin week 2	Eind week 2	Begin week 3
2	PvA met mindmap					
3	code prototype					
4	Keuze tool en FSA model					
5	Onderzoeksvraag bedacht					
6						
7	Environment met agents gemaakt					
8	Agents kunnen stemmen op partij					
9	Uitslag electie worden opgeslagen					
10	Code flexibel maken voor aanpassingen					
11	agents en partijen hebben een 'ideologie'					
12	GUI maken					
13						
14	Agents hebben een motivatie om voor een bepaalde partij te stemmen met Logica erachter (keuze met interessen)					
15	Agents hebben een logica om ervoor te Zorgen dat hun minst liefste partij niet wint					
16	partijen kunnen stoppen met een volgend Electie					
17						
18	Legenda:					
19	Werken aan					
20	Moet dan af zijn					
21	Mag beginnen (optioneel)					
22	Datum gaat over 'voor donderdag'					
23						

Risico's

X

Kiezen van een tool

SFA Unity

Suitability

- Unity is erg geschikt voor het maken van dit simulatie. Het heeft een breed assortiment aan tools, libraries en bevat weinig limieten met wat je kan simuleren. Het is wel dat Unity niet gemaakt is voor dit soort simulaties waardoor er vaak meer nagedacht moet worden over de werkingen en logica achter Unity om hetzelfde eindresultaat te krijgen. Daarnaast hebben wij zelf totaal geen ervaring met Unity, we hebben in totaal maar 4 dagen voor Unity gehad, dus zullen wij niet veel van de tools afweten. **5/10**

Feasability

- Unity gebruiken is totaal niet haalbaar. Unity is lastig te gebruiken en bevat een overweldigende aantal keuzes en toepassingen waardoor het gebruik hiervan erg langzaam zal worden. Daarnaast is Unity niet heel geweldig voor je RAM en is het bedoeld voor 3D. **1/10** wouldnt rate again

Totaal: 3/10

SFA Mesa

Suitability

- Mesa is erg simpel te begrijpen en heeft de standaard opbouw van python waardoor iedereen wel zou begrijpen wat er staat. Ook is het geschikt voor het maken van deze simulatie. Het enige min punt is dat de weergave niet altijd even mooi is. Verder zijn er geen opties voor sliders. En is er geen mogelijk om de simulatie niet tussentijds aanpassen. **8/10**

Feasability

- Mesa is best haalbaar, dit komt omdat Mesa eenvoudig te gebruiken is. Daarnaast is het geschreven in Python, waar bij iedereen een grote kennis al beschikbaar is. We kunnen er een Jupyter Notebook van maken om de code nog beter te laten verlopen. **10/10**

Totaal: 9/10

SFA NetLogo

Suitability

- NetLogo is redelijk geschikt voor het maken van deze simulatie. De taal heeft niet veel libraries. Maar het heeft wel genoeg tools om complexe simulaties mee te maken. NetLogo is gemaakt voor simulaties maar het ligt eraan hoe ingewikkeld wij de simulatie willen maken of NetLogo de goede keus is. **9/10**

Feasability

- NetLogo is echt een "baby's first programming language". Het is supermakkelijk te begrijpen en het werkt ook nog snel. Natuurlijk moet je wel wat basis dingetjes leren. Maar je kan al snel aan de slag met het echte werk. **9/10**

Totaal: 9/10