Voorwoord:

Wij zullen een stemsysteem simuleren waarbij er een 2d axis is die een schaal moet voorstellen van niet mee eens naar totaal mee eens. Dit gaan wij simuleren en de resultaten ervan plotten. Voor een algemeen idee zie deze video: <https://www.youtube.com/watch?v=yhO6jfHPFQU>

Onderzoeksvraag:

Hoe beïnvloeden mensen die tactisch stemmen de uitkomst van de instant-runoff stemsysteem?

Modules:

1. Agent spawning:  
Het aanmaken en in spawnen van de hoeveelheid agents in de simulatie. Hoe moeilijk is het om agents aan te maken. Wanneer kan je deze aan de simulatie toevoegen.

2. Maken van nieuwe partijen:  
Het aanmaken van nieuwe partijen waar agents op kunnen stemmen en hoe moeilijk het is om dat te doen. Wanneer kun je deze aan je simulatie toevoegen.

3. Veranderen van de stem strategie van de agent:  
Hoe geef je een agent een strategie. Kun je de strategie van de agent gemakkelijk verwisselen voor verschillende simulaties.

4. Visualiseren van winnende partij:  
Op welke manier kunnen we de uitkomst visualiseren en is deze manier handig als je het eindresultaat wilt analyseren

5. Visualiseren op welke partij een agent gekozen heeft:  
Is het mogelijk om te visualiseren op welke partij een agent gestemd heeft door bijvoorbeeld de kleur van de agent te veranderen.

6. Data opslaan:  
De data moet opgeslagen worden zodat we het kunnen visualiseren. In wat voor file moet deze data staan en waar kan de tool mee omgaan.

SFA UNITY:

# 1. Geschiktheid:

*1.1.* *De tool is geschikt om de modules hierboven te ontwikkelen. Beoordeel dit voor elk van je modules.*

**Agent spawning:**  
In Unity kan je makkelijk agents spawnen in de vorm van gameobject met scripts erop.  
(8/10)

**Maken van nieuwe partijen:**Ditzelfde geldt voor de partijen.  
(8/10)

**Veranderen van de stem strategie van de agent:**Dit is technisch makkelijk te doen omdat je c# gebruikt en wel bekende patterns kan gebruiken zoals de strategy pattern maar het is geen agent based language.   
(7/10)

**Visualiseren van winnende partij:**Het visualiseren van de winnaars en agents kan heel simpel d.m.v materials te veranderen en of tekst. Maar daar in tegen zijn grafieken heel moeilijk te maken in Unity en dit is juist hetgeen wat het beste meerdere resultaten kan weergeven.  
(6/10)

**Visualiseren op welke partij een agent gekozen heeft:**Het visualiseren van een agent gaat op dezelfde wijze als een partij.(6/10)

**Data opslaan:**C# biedt veel manieren om data op te slaan, het cijfer is equivalent aan de compatibiliteit cijfer omdat het dezelfde dataformaten support bij het opslaan als bij het lezen. Soms kan json serializing wel moeilijk zijn waardoor ik het niet de maximale punten geef.  
(8/10)

*1.2.* *Efficiëntie, snelheid: hoe lang duurt het om de simulatie te draaien, is de tool snel genoeg?*

De tool is efficiënt genoeg om de simulatie op grote schaal te draaien, indien het nodig is om op een extreem grote schaal te draaien kan er gebruik worden gemaakt van Unity DOTS.  
(7/10)

*1.3.* *Compatibiliteit: als je externe data gebruikt, kan de tool omgaan met het dataformat?*

Met Unity kan je gebruik maken van C# en deze biedt brede support aan voor een wijde selectie van dataformats, van .csv bestanden tot json formats.  
(8/10)

# 2. Haalbaarheid:

*2.1.* *Kunnen je teamleden omgaan met de tool?*

Mijn teamleden hebben geen ervaring met Unity en kunnen dus niet met de UI omgaan en hebben lichte tot geen ervaring met C#.  
(4/10)

*2.2.* *Is het technisch haalbaar een minimaal product te maken in 2 weken?*

Technisch is het goed haalbaar om een MVP te maken in 2 weken. Dit zeg ik omdat ik binnen 2-3 dagen het vorige project in Unity heb weten te maken, maar de werkdruk wordt hoger aangezien ik de enige met ervaring in C# ben.  
(6/10)

SFA MESA:

# 1. Geschiktheid:

1.1. De tool is geschikt om de modules hierboven te ontwikkelen. Beoordeel dit voor elk van je modules.

Ik denk dat module een tot en met vier geen probleem moet zijn voor mesa. Het is namelijk mogelijk om aan het begin van je simulatie te kiezen met hoeveel agents je hem wilt runnen maar tijdens de simulatie kun je niet individuele agents toevoegen. De stem strategie kun je per agent meegeven dus dat moet ook geen probleem zijn en natuurlijk is het ook geen probleem om extra partijen toe te voegen waarop ze kunnen stemmen. Het visualiseren van de resultaten zal voornamelijk bestaan uit grafieken en diagrammen. Ik weet niet of het mogelijk is om een live visualisatie te doen dus dat kan gezien worden als een minpunt. Ik denk dat het wel mogelijk is om te zien welke agent gestemd heeft op welke partij maar ook dit is waarschijnlijk geen live kleurverandering.

**Agent spawning:**Je kan in mesa de simulatie runnen en van te voren instellen hoeveel agents je wilt.   
(7/10)

**Maken van nieuwe partijen:**Dit is goed te doen en die kun je dan ook weer invoeren wanneer je de volgende batch runt.   
(7/10)

**Veranderen van de stem strategie van de agent:**De stem strategie kun je ook van te voren aanpassen en dat moet dus geen probleem zijn.   
(6/10)

**Visualiseren van winnende partij:**  
Het is goed te doen om met mesa visualisaties te maken maar dan krijg je wel visualisaties zoals grafieken.   
(7/10)

**Visualiseren op welke partij een agent gekozen heeft**:  
Ik denk dat het wel mogelijk moet zijn om de individuele data van een agent bij te houden maar misschien met een kleine omweg.  
(6/10)

**Data opslaan:**  
Het is mogelijk om de data op te slaan zowel intern als extern, ook is deze tool goed te combineren met bijvoorbeeld pandas en andere tools waarmee je data verwerkt in python.  
(8/10)

1.2. Efficiëntie, snelheid: hoe lang duurt het om de simulatie te draaien, is de tool snel genoeg?

Ik merkte tijdens het maken van de tutorial wel dat het af en toe lang duurde maar het is de eerste keer dat ik met simulaties werkte dus ik weet niet of mesa langzamer is dan andere tools.   
(5/10)

1.3. Compatibiliteit: als je externe data gebruikt, kan de tool omgaan met het dataformat?

Ik weet niet of te combineren is met elke module, op de site van mesa stond wel dat het goed samenwerkt met de panda’s module.   
(10/10)

# 2. Haalbaarheid:

2.1. Kunnen je teamleden omgaan met de tool?

Het zal niet zo zijn dat je na het doorlopen van de tutorial meteen de module kunt gebruiken tot zijn volle potentie maar er mee omgaan zal wel moeten lukken.  
(9/10)

2.2. Is het technisch haalbaar een minimaal product te maken in 2 weken?

Ja, dat moet wel mogelijk zijn, er zijn namelijk genoeg tutorials te vinden die ons hierbij kunnen helpen.  
(8/10)

SFA NETLOGO:

# 1. Geschiktheid:

1.1. De tool is geschikt om de modules hierboven te ontwikkelen. Beoordeel dit voor elk van je modules.

**Agent spawning:**

NetLogo kan in principe al deze modules aan. Het spawnen van agents kan gemakkelijk, maar moet echter wel gedaan worden voor het starten van de simulatie, en kan niet tussentijds.  
(7/10)

**Maken van nieuwe partijen:**

Het maken van partijen kan ook gemakkelijk, maar moet net als het spawnen van agent voor het starten van de simulatie gedaan worden.  
(7/10)

**Veranderen van de stem strategie van de agent:**

Het veranderen van stem strategie moet in principe ook kunnen, aangezien NetLogo selectie-menu’s ondersteund waarin je keuzes tussen strategieën kan selecteren.  
(8/10)

**Visualiseren van winnende partij:**

Het visualiseren van winnende partij kan ook makkelijk, en eventueel ook in verschillende soorten grafieken.  
(6/10)

**Visualiseren op welke partij een agent gekozen heeft:**

Het visualiseren van voor welke partij wordt gekozen door een agent kan in de grafiek gedaan worden door kleuren en is daardoor ook heel makkelijk te doen.  
(6/10)

**Data opslaan:**

Het opslaan van data kan, zolang het maar in een .txt file wordt gezet. Maar als het in een ander formaat opgeslagen moet worden kan dit niet.  
(4/10)

1.2. Efficiëntie, snelheid: hoe lang duurt het om de simulatie te draaien, is de tool snel genoeg?

NetLogo werkt heel snel. Tijdens de tutorial was elke stap vrijwel direct berekend, en in NetLogo kan je zelfs eventueel de snelheid van de simulatie verhogen als je dat zou willen tijdens het runnen.  
(8/10)

1.3. Compatibiliteit: als je externe data gebruikt, kan de tool omgaan met het dataformat?

NetLogo kan dit, zo lang het een .txt file is. Zodra het een ander dataformat is kan het niet worden geïmporteerd worden en dus ook niet gebruikt worden. (3/10)

# 2. Haalbaarheid:

2.1. Kunnen je teamleden omgaan met de tool?

De codetaal van NetLogo is niet heel complex dus zelf al weten ze niet hoe ze het moeten gebruiken, het is heel gemakkelijk om te leren en heel gebruiksvriendelijk, en kan in een uur of twee begrepen worden.  
(8/10)

2.2. Is het technisch haalbaar een minimaal product te maken in 2 weken?

Ja, aangezien ik al zei dat het heel makkelijk is en heel gebruikersvriendelijk, waardoor het heel snel af zou kunnen zijn.  
(8/10)

Toolkeuze:  
Onze tool keuze is uiteindelijk op Mesa gevallen.   
  
Pros: Wij hebben deze keuze gemaakt omdat wij het visualiseren dit alles een belangrijk punt is, Mesa is python based en biedt meerdere mogelijkheden om te visualiseren in bijzonder om vrij makkelijk grafieken te plotten wat onze uiteindelijke doel is en onze visie van het project. Verder hebben wij alle 3 ervaring met Python dus zullen we alle een bijdrage kunnen leveren aan de code.Ook zijn er veel extra modules mogelijk om te installeren voor python wat de compatibiliteit een hoge score geeft.

Cons: Het biedt geen mogelijkheid voor een 3d visualisatie, verder is python van nature niet agent based en zal een andere optie zoals netlogo misschien sneller zijn. We moeten wel alle wennen aan Mesa.

Naar veel onderzoek gedaan te hebben en veel gediscussieerd te hebben zijn we van Unity naar Netlogo naar uiteindelijk Mesa gegaan met onze keuze. Onze keuze werd verder bevestigd toen we zagen wat voor leuks je er mee kan doen bij de demo van Casper de student assistent.

|  |  |  |  |
| --- | --- | --- | --- |
| Modules | Unity | Mesa | Netlogo |
| Agent spawning | (8/10) | (7/10) | (7/10) |
| Partij spawning | (8/10) | (7/10) | (7/10) |
| Stem strategie | (7/10) | (6/10) | (8/10) |
| Agent visualisatie | (6/10) | (7/10) | (6/10) |
| Partij visualisatie | (6/10) | (6/10) | (6/10) |
| Data opslaan | (8/10) | (8/10) | (4/10) |
| Efficiëntie | (7/10) | (5/10) | (8/10) |
| Compatibiliteit | (8/10) | (10/10) | (3/10) |
| Team ervaring | (4/10) | (9/10) | (8/10) |
| Technisch haalbaar | (6/10) | (8/10) | (8/10) |
| Totaal | (68/100) | (73/100) | (65/100) |

We zien in deze tabel dat onze voorkeur wordt bevestigd en dat Mesa boven op uit komt.

Planning:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Week 1 | Week 2 | Week 3 |
| Maandag | - Onderzoeksvraag bepalen  - Tool onderzoeken en kiezen d.m.v SF(A) modellen | - Agents aanmaken  - Partijen aanmaken | - Onderzoek uitvoeren  - Analyse uitvoeren |
| Dinsdag | - Rapport met tool verklaring  - PVA met user stories en taken | - Plurality simulatie maken  - Instant-runoff simulatie beginnen | - Verslag maken |
| Woensdag | - Basis geval programmeren  - Alles polishen met eventuele feedback | - Instant-runoff simulatie afmaken    - Visualisatie maken van de stem strategie | - Afronden en feedback implementeren |
| Donderdag | - Alles polishen op basis van feedback | - Afronden en feedback implementeren | - Afronden en feedback implementeren |

Voor de taakverdeling zullen we rouleren om het interessant te houden voor iedereen, verder zullen we gezamenlijk elke dag een moment inplannen om er samen aan te zitten en de voortgang te bespreken. De planning is gebaseerd op de user stories en zal tussentijds niet veel aangepast worden zolang we deze behalen, in het geval dat we eerder klaar zijn zullen we extra user stories maken en deze in de planning verwerken.

User stories:

# User story 1: Simulatie maken voor plurality voting

Beschrijving:   
Als product owner wil ik dat de simulate werkt zodat het een goede beeld van de werkelijkheid schetst.

DoD:  
De stappen van de simulatie zijn doorlopen en werken allemaal naar behoren. Verder zitten er in de simulatie weinig tot geen bugs.

Prioriteit:   
9

Estimate:   
7 uur

# User story 2: Simulatie maken voor instant-runoff voting

# Beschrijving: Als product owner wil ik dat de simulate werkt zodat het een goede beeld van de werkelijkheid schetst.

DoD:  
De stappen van de simulatie zijn doorlopen en werken allemaal naar behoren. Verder zitten er in de simulatie weinig tot geen bugs.

Prioriteit:   
9

Estimate:   
7 uur

# User story 3: Agents aanmaken

Beschrijving:  
Als product owner willen wij agents aanmaken die stemmen zonder tactiek ook willen wij agents aanmaken die stemmen met tactiek om te kijken wat de invloed is op de uitkomst.

DoD:  
Twee verschillende soorten agents, een die gewoon op zijn top partij stemt en de andere die tactisch stemt.

Prioriteit:  
8  
Estimate:  
4 uur

# User story 4: Visualisatie van de resultaten.

Beschrijving:   
Als product owner wil ik dat de visualisatie duidelijk is zodat het een goede weergave van de simulatie geeft.

DoD:  
De grafieken zijn te beredeneren en logisch te verklaren.Het is snel en makkelijk duidelijk wat alles moet voorstellen.

Prioriteit:   
9

Estimate:   
6 uur