

## Welke stelsysteem is het meest succesvol?

**Gekozen onderwerp:** Stelsystemen

**Eerder gebruikte onderzoeksvraag:** Wat is het effect van plurality voting op het aantal mensen dat niet op zijn eerste voorkeur kiest?

**Uitgebreide onderzoeksvraag:** Welke stelsysteem is het meest succesvol?

## Onderzoeksvraag

### Eerste inzicht

Om de uitgebreide onderzoeksvraag te beantwoorden, moet eerst de eerder gebruikte onderzoeksvraag beantwoord worden. Dit, in tegenstelling tot de uitgebreide, wordt er alleen gekeken naar plurality stelsysteem. Deze vraag kan beantwoord worden door de betrouwbaarheid te berekenen. De simulatie wat het verschil is tussen het strategisch kiezen (andere voorkeur, maar ten minste beter dan de slechtste) of favoriete partij (eerste voorkeur, maar grote kans dat de slechtste kan winnen). Om dit te simuleren moet de simulatie het volgende doen:

### Setup

- Cleared eerder gemaakte simulatie en maakt een nieuw wit vlak aan
- Maakt bepaald hoeveelheid kandidaten aan.
- 2d oppervlakte
- 4 Issues (gemaakte partij/kandidaat keuzes)
- Zet kandidaten bij willekeurige issues op de 2d oppervlakte
- Zet meer dan 2 (herkenbare)partijleiders op de oppervlakte en geef ze een (herkenbare) kleur

### Go (voor nu favoriet plurality)

- Cleared alle kleuren van de kandidaten (behalve de partijleiders)
- Kandidaten zoeken naar de dichtbijzijnde partijleider
- Kandidaat veranderd naar de dichtbijzijnde partijleider's kleur
- Systeem telt welke kleur er het meeste van is
- De winnende partij(leider) krijgt een herkenbaar teken waarin hij gewonnen heeft

### Go (voor nu tweede keuze plurality)

- Cleared alle kleuren van de kandidaten (behalve de partijleiders)
- Kandidaten zoeken naar de tweede dichtbijzijnde partijleider
- Kandidaat veranderd naar de tweede dichtbijzijnde partijleider's kleur
- Systeem telt welke kleur er het meeste van is
- De winnende partij(leider) krijgt een herkenbaar teken dat hij gewonnen heeft

### GUI

- Een model, wat de 2d oppervlakte weergeeft met daarin alle kandidaten en partijleider
- Een favoriet plurality histogram plot, wat de hoeveelheid getelde kleuren zijn op het model
- Een tweede keuze plurality histogram plot, wat de hoeveelheid getelde kleuren zijn op het model

## Welke stelsysteem is het meest succesvol?

- Een go knop, om de go functie (voor nu plurality) op aan te roepen
- Een setup knop, om de setup functie op aan te roepen
- Aantal kandidaten slider
- Aantal partijleiders slider

## Tool gebruik

De 3 tools waaruit we moeten kiezen zijn Unity, Netlogo en Python Mesa. Unity vinden we heel erg lastig en onhandig om mee te werken en laten deze voor nu achterwegen. De andere twee gaan we vergelijken met elkaar

**Suitability:** De simulatie moet voor het onderzoek best simpel en eenvoudig eruitzien. Daarbij is de GUI van groot belang. Netlogo heeft al een ingebouwde GUI in tegenstelling tot Python Mesa die dit nog moet aanmaken. Daarnaast zijn verkiezingen het beste te simuleren in willekeurigheid en de uitkomst van de kandidaten op te tellen, dit moet zo simpel mogelijk kunnen, waarbij Netlogo een enorm groot voordeel mee heeft. Netlogo kan erg simpel al een oppervlakte met kandidaten en ze kleuren geven. Python Mesa moet daarvoor vele code schrijven, waardoor de verhouding van code tussen de twee tools erg van elkaar verschillen. Netlogo is daarom de beste in dit onderdeel.

**Feasibility:** De opties die beide tools hebben, verschillen erg van elkaar. De meeste Netlogo's opgeloste problemen staan in de handleiding van Netlogo. Daarnaast is Netlogo ook beperkt tot deze handleiding gemaakt door de makers, wat Netlogo niet uitbreidbaar maakt. Dit is ook de reden waarom Netlogo niet grote project kan maken zoals een grote 3d game simulatie, omdat het simpelweg niet de kennis heeft. Python Mesa daarentegen is een library van Python, wat betekent dat je alle libraries van python tot beschikking hebt. Daarnaast heeft python Mesa ook vele opgeloste problemen beantwoord op meerdere sites zoals stackoverflow of youtube. Dit is Netlogo zeer beperkt dus daarom is Python Mesa de winnaar van dit onderdeel

**Acceptability:** Python Mesa geeft volgens de andere onderdelen aan dat het erg goed is met uitbreiden en grote projecten, maar tegen vele problemen kan komen, waardoor voor kleine projecten het de moeite niet waard is om het te simuleren. Netlogo kan met behulp van zijn heel erg overzichtelijke code en software, erg gemakkelijk kleine problemen tegengaan, waardoor een klein project erg geschikt is voor Netlogo. De verkiezingssimulatie verstaan we onder een klein project, waardoor dit onderdeel gewonnen is door Netlogo

**Conclusie:** Netlogo ons advies om te gebruiken voor dit project. Wat Netlogo namelijk veel beter kan dan Mesa, is dat het zo erg eenvoudig project kan maken en er zo veel eerder gemaakte voorbeelden/modellen zijn die je kan helpen met het maken van de back-end en het front-end.

Welke stelsysteem is het  
meest succesvol?

## Plan van aanpak

### User stories

Titel	Beschrijving	Duur
Setup	Maakt aantal turtles aan, turtle eigenschappen, global variabele, plaats ze op oppervlakte en kiest verschillende turtles als partyleader met kenmerkende eigenschappen.	2 uur
Gofirst	Cleared eerst alle eerder verandere variabele en turtles, roept de findfirst party functie aan en dan de functie countelections	3 uur
Gosecond	Cleared eerst alle eerder verandere variabele en turtles, roept de findsecondparty functie aan en dan de functie countelections	1 uur
Findfirstparty	Elke turtle gaat checken in zijn radius of er een partyleader is anders verbreed het zijn radius totdat het een partyleader vindt en dan zijn kleureigenschap veranderd naar de partyleaders en dan stopt de loop	5 uur
Findsecondparty	Elke turtle gaat checken in zijn radius of er een partyleader is anders verbreed het zijn radius totdat het een partyleader vindt en dan zijn kleureigenschap veranderd naar de partyleaders, dit doet hij 2 keer (om de 2de partyleader te vinden)	4 uur
GUI	Knoppen Setup, GOfirst en GoSecond toegevoegd	1 uur
Clearboard (tijdelijk)	Tijdelijke functie om alle kleuren van kandidaten te reseten en daarna deze functie eigenschappen/code regels in de go functies toepassen	1 uur
CountElections	Zet de globale tellings variabelen op nul, telt per turtle hoeveel er per kleur is en zet het in een variabele. De variabelen worden uitgeprint in de output	3 uur
GUI uitbreiding	Histogram van de zichzelf tellende code van het aantal kleuren en zet de in een plot met x-as de kleur code en y-as de hoeveelheid. En de output van Countelections waarin in exacte waardes te zien is welke kleuren meer votes hebben. En nog een klein notitieblokje om aan te geven welke output bij welke kleur hoort en de kleurcode nummer (white = 9.9)	2 uur

### Planning

#### Week 2 planning

Onderdeel	Beschrijving	Weekdeel	Persoon
Vorbereiding	Korte standup van week 2	MA	Iedereen
Vorbereiding	Userstories aanmaken	MA	Iedereen
Simulatie	Plurality, Strategisch Kiezen toevoegen	MA-DI	Ruben
Vorbereiding	Run-off Stelsysteem onderzoek en checklist aanmaken	DI	Guy & Koen
Simulatie	Run-off, Basics functies	WE-DO	Koen
Simulatie	Run-off, Uitbreiding Basics Functies	DO	Ruben
Simulatie	Run-off checklist doorlopen en afmaken	DO-FR	Guy&Koen
Vorbereiding	Approval StemSysteem onderzoek en checklist aanmaken	DO-FR	Ruben

#### Week 3 planning

Onderdeel	Beschrijving	Weekdeel	Persoon
Vorbereiding	Korte standup van week 3	MA	Iedereen

Welke stelsysteem is het  
meest succesvol?

Voorbereiding	Userstories aanmaken	MA	Iedereen
Simulatie	Approval, Basics functies	MA-DI	Ruben & Koen
Simulatie	Uitbreiden en vorige functies verbeteren	MA-DI	Guy & Koen
Simulatie	Hele simulatie in orde maken en fouten oplossen	WE-DO	Iedereen

### SF(A) model met modules voor tools

Scores zijn van 1-5, waarbij 1 het laagst is en 5 het hoogst haalbare is.

- Mesa: 29 punten
  - **Gui: 4**  
Je hebt een best wel moderne GUI en je kan ook het veranderen van parameters implementeren.
  - **Agents: 4**  
De agents zijn makkelijk op te zetten en makkelijk functies te maken voor de agents zelf.
  - **Environment: 4**  
De environment is ook makkelijk op te zetten en best simpel.
  - **Staat van simulatie toonbaar: 4**  
Voor extra informatie moet je het wel zelf inprogrammeren, dus het weergeeft niet vanuit zichzelf de status van een agent als je erop klikt/erover heen hoveret.
  - **Batchrun: 4**  
Je zou makkelijk een batchrun kunnen uitvoeren in Mesa. Het is al geïmplementeerd in Mesa.
  - **Suitability: 4**  
Qua efficiëntie is Mesa redelijk. Je moet nog steeds wel aardig wat code schrijven om iets simpels te maken. , Mesa heeft een aardige snelheid qua het runnen en displaying van de code, omdat Mesa in python word gebruikt is de compatibility hoog
  - **Feasability: 5**  
Iedereen weet hoe ze moeten coderen in python, dus er hoeft niet echt een nieuwe taal aangeleerd te worden. Mesa is aardig goed te begrijpen. Technisch is het zeker haalbaar om in 2 week iets goed in elkaar te zetten.
- Unity: 25 punten
  - **Gui: 5**  
De GUI in Unity is heel uitgebreid en zorgt ervoor dat alle nodige info zichtbaar is, mits je weet waar je het moet vinden.
  - **Agents: 4**  
Agents zijn goed op te zetten in Unity.

Welke stelsysteem is het  
meest succesvol?

- **Environment: 4**  
Een environment opzetten in Unity is vrij makkelijk.
  - **Staat van simulatie toonbaar: 3**  
Afhankelijk van de simulatie is dit makkelijk of moeilijk. Als je een visuele indicator toevoegt wordt het al een stuk leesbaarder, maar het is moeilijker dan in de andere 2 talen
  - **Batchrun: 5**  
Batch is een zeer goed geïntegreerde command in Unity aangezien ze tijdens de development van Unity wisten dat compilen, runnen etc. lang zou kunnen duren.
  - **Suitability: 2**  
Suitability is niet erg hoog. In Unity zou onze simulatie niet efficiënt zijn, de compilatietijd zou vrij lang duren. Waarschijnlijk zou Unity beter geschikt zijn voor de snelweg-simulatie.
  - **Feasability: 2**  
Wij zijn alle 3 niet echt bekend met Unity/C#, en we zouden niet goed weten waar we zouden moeten beginnen. Technisch zou het waarschijnlijk wel haalbaar zijn, maar het zou langer duren dan nodig is en in andere talen is het vele malen makkelijker.
- Netlogo
    - **Gui: 5**  
De gui in Netlogo is al ingebouwd en staat als template al klaar om gebruikt te worden, het enige wat er gedaan moet worden is het aanmaken van turtles
    - **Agents: 4**  
Agents kunnen aangemaakt worden met 3 woorden, het is erg simpel om ze aan te maken en ze te inspecteren.
    - **Environment: 5**  
De basis van het environment wordt al aangemaakt als je een template aanmaakt en is daarom automatisch aangemaakt.
    - **Staat van simulatie toonbaar: 5**  
De agent informatie is erg makkelijk te zien door bij het environment op de agent te klikken en dan kan je het inspecteren en kan zien welke variabele met agent te maken hebben en welke waarde het heeft
    - **Batchrun: 3**  
Heeft BehaviorSpace i.p.v. batch mode. BS runt op een parallelle core op je CPU in plaats van wanneer je PC in idle mode is. BehaviorSpace werkt wel goed maar is dus in werkelijkheid geen echte batch mode.
    - **Suitability: 4**  
Efficientie?, snelheid?, compatibiliteit?
    - **Feasability: 4**  
Teamleden begaand met tool?, Technisch haalbaar?