

# Touring Machines and Electric elections

Hussin Almoustafa

Martin Rog

Fares Hamad

# Inhoudsopgave

Inleiding

Onderzoeksvraag

Agents

    Lijsttrekker

    Stemmers

Scenario's

Toolkeuze

    Conclusie toolkeuze

Risico's

Userstories

    Tasks

Planning

## Inleiding

Veel landen hebben een tweepartijstelsel, hoewel de regels dat meestal niet zeggen, en de meeste mensen zijn het erover eens dat een tweepartijstelsel niet zo prettig is.

Toch is dit gebruikelijk omdat het een resultaat is van stemmen. We willen een paar verschillende stelsystemen en het stemgedrag simuleren met de strategieën die daarbij horen.

## Onderzoeksvraag

Wat is het effect van plurality voting ten opzichte van approval op het aantal mensen dat niet op zijn eerste voorkeur kiest?

## Agents

Voor dit simulatie maken we 2 soorten agents, een lijsttrekker en een stemmer.

## Lijsttrekker

Een Lijsttrekker heeft geen acties wat hij kan doen. Het is puur ter onderscheiding van de kiezers. de leider heeft de volgende eigenschappen;

- Positie :

De plek op de grid om de visie van de partijleider te verduidelijken

- Naam :

De naam van de leider

## Stemmers

Een kiezer is een persoon die zich ergens op het vlak van de simulatie bevindt. Deze agent heeft een positie op de grid en kan de positie van de lijsttrekker zien en zijn stemgedrag daarop aanpassen.

- Positie

De plek op de grid om politieke voorkeur te tonen  
de kiezers maken

- Strategie

De staatsstrategie die de kiezer handhaaft om een partijleider te stemmen.

## Gui

We hebben een beschrijving van de gui in het netlogo programma verwerkt onder de info tab. Hier is terug te vinden hoe de gui werkt, hoe je het gebruikt en een beschrijving van de items in de interface.

## Scenario's

## Toolkeuze

### Us1 – Stemmer agent maken

SFA	Mesa	Netlogo	Unity
Geschiktheid	7	9	8
Haalbaarheid	7	9	8
Totaal	14	18	16

## Us2 – Lijsttrekker Agent maken

SFA	Mesa	Netlogo	Unity
Geschiktheid	7	9	8
Haalbaarheid	7	9	8
Totaal	14	18	16

## Us3 – Stemmer Strategie maken

SFA	Mesa	Netlogo	Unity
Geschiktheid	6.5	9	8
Haalbaarheid	8	9	7
Totaal	14.5	18	15

## Us4 – Gui maken

SFA	Mesa	Netlogo	Unity
Geschiktheid	7	9	8
Haalbaarheid	9	9	6
Totaal	16	18	14

## Us5 – Simulatie veld bepalen & maken

SFA	Mesa	Netlogo	Unity
Geschiktheid	6	9	8
Haalbaarheid	9	10	8
Totaal	15	19	16

## Totaal SFA

SFA	Mesa	Netlogo	Unity
	45.5	55	45

## Conclusie toolkeuze

We hebben ervoor gekozen om Netlogo te gebruiken voor deze opdracht. De reden hiervoor is dat de simulatie van een agent op een 2D grid en interactie tussen verschillende agents eenvoudig te realiseren is. Het enige kleine knelpunt dat we tegenkomen is dat de programmeertaal in principe basic is.

Mesa is ons back-up plan omdat de programmeeromgeving ons goed bekend is, maar het is voor ons doel moeilijker om een goede simulatie te maken met agents.

We kunnen Untiy gebruiken, maar het is een compleet nieuwe omgeving en Unity is niet echt voor deze simulaties.

## Risico's

Het grootste risico voor dit project is tijd.

## Userstories

Ons project is opgedeeld in vijf user stories. Hieronder worden ze kort benoemt met een "definition of done" erbij (DoD).



### **User story 1: Stemmer agent maken**

DoD: Deze agent moet random in het simulatie veld worden geplaatst en het aantal hiervan moet door de gebruiker bepaalt kunnen worden.

### **User story 2: Lijsttrekker Agent maken**

DoD: Deze agent moet random in het simulatie veld worden geplaatst en het aantal hiervan moet door de gebruiker bepaalt kunnen worden.

### **User story 3: Stemmer Strategie maken**

DoD: De gebruiker moet een keuze kunnen maken tussen de drie van te voren bepaalde stem systemen.

### **User story 4: Gui maken**

DoD: De GUI moet makkelijk te begrijpen zijn voor een buitenstaander en moet een goede beschrijving hebben.

### **User story 5: Simulatie veld bepalen & maken**

DoD: De agenten moeten random worden geplaatst elke keer dat de Setup knop gebruikt wordt.

## Planning

Voor de planning van dit project zijn we agile te werk gegaan. Dit hebben wij gedaan met behulp van het github planboard. Hier staan alle user stories in verwerkt. Op deze manier konden we goed bijhouden welke taken al voltooid waren en welke we nog moesten doen. We konden zo ook goed zien voor welke onderdelen we nog extra tijd nodig hadden en hoe we dit eventueel konden realiseren.