

1.

Netlogo is beter dan Mesa en Unity om verschillende redenen. Netlogo is grotendeels beter dan Unity omdat verschillende versies van Unity geld kosten, en bij Netlogo zijn ze allemaal gratis. Daarnaast is Netlogo makkelijker te gebruiken en coderen dan Unity en Mesa, die iets complexer zijn. Aangezien we nog aan het leren zijn over coderen is Netlogo ook beter, omdat deze meer voorbeelden en tutorials heeft dan Unity of Mesa. Ten slot is het visualiseren van data in Netlogo een simpele acties van objecten op het interface scherm toevoegen in plaats van ze erin te coderen.

Netlogo is Agent based, omdat het grootste deel van de interfaces ervoor zijn gemaakt om meerdere entities en de interactie tussen deze entities weer te geven.

In de losse aanpassing van het programma heb ik het gemaakt dat agents met 0 wealth geen bedelen bij mensen met meer dan 100 wealth, en dat agents met meer dan 250 wealth 2 wealth moeten weggeven in plaats van 1 wealth. Wat je hieruit kan zien is dat het simple economy model stabiel wordt, en de top 10% niet rijker wordt dan de bottom 50%.

2.

De Simple Economy simulatie heeft geld en agents. De agents hebben een interne staat hoeveel geld ze bezitten. Ze kunnen alles in de omgeving zien. De initiele interne staat van de agent is dus de hoeveelheid geld die ze bezitten. De functie See neemt een staat van de omgeving en gooit een percept eruit. Ze kunnen transacties uitvoeren met elk van de agents, waarbij ze 1 geld weggeven. Ze kunnen echter alleen transacties uitvoeren als hun eigen hoeveelheid geld groter dan 0 is. Oftewel, de Act functie kijkt naar de interne staat van de agent. Als die beschrijft dat deze meer dan 0 geld heeft, haalt hij 1 geld van de huidige agent weg, en geeft hij deze aan een random andere agent. De Update functie verandert voor de agent met meer dan 0 geld dat het geld  $- 1$  word, en dat het voor een andere random agent geld  $+ 1$  word. In deze simulatie zijn de See, Act en Update functie samengevoegt in de functies Go en Transact, waarbij Go de juiste agents geld laat uitdelen, Transact de agents geld laat weggeven, en Go het daarna update.

3.

In de Simple Economy omgeving kunnen alle agents de huidige waardes van random andere agents ophalen, waardoor de omgeving heel accessible is. In deze omgeving staan de gevolgen niet altijd vast; de hoeveelheid geld blijft hetzelfde, of er wordt 1 geld weggegeven, maar het wordt aan een random agent gegeven. Hierdoor kunnen we de omgeving non-deterministic noemen. De omgeving kijkt bij acties alleen naar huidige waardes, waardoor het ook episodisch is. Er zitten geen andere processen in de omgeving behalve de acties, wat de omgeving static maakt. Tot slot kan de omgeving oneindig lang doorwerken, waardoor het ook continuous is.

4.

Bij het Simple Economy model is het feit dat het continuous is heel belangrijk, maar bij een model van een winkel niet, omdat je hier een werkweek of alleen een werkdag kan simuleren. Het feit dat het static is is hier ook niet belangrijk, omdat de prijzen van de producten kunnen veranderen. Ten slot is het feit dat het accessible is ook niet belangrijk, omdat je van tevoren niet kan weten of een klant genoeg geld heeft om te betalen.