Thread = Een ‘route’ die de applicatie afloopt.

# Aanbevelingen

Alhoewel de functionaliteit van de V&K simulatie is zoals gevraagd worden er nog aangeraden naar bepaalde onderdelen te kijken. Deze onderdelen veranderen niks aan het uiterlijk of de functies van de applicatie maar maken deze wel flexibeler. Het gaat hier om threads en de Simulation settings

## Threads

Een thread is een ‘route’ die de applicatie afloopt. Wanneer u in de V&K simulatie 100 stappen simuleert kan de applicatie niks anders doen, dat komt omdat hij bezig is met die 100 stappen en niks kan daar tussenkomen. Hij moet die route aflopen.

Wanneer u in het programma een tweede thread toevoegt kan het programma twee routes aflopen. Dit betekend dat er één route is die constant eventuele stappen simuleert en een andere route die op gebruikersinvoer wacht. Wanneer u nu een stap zou simuleren is er nog steeds de mogelijkheid onderdelen van de GUI te gebruiken zonder te hoeven wachten totdat het aantal stappen gesimuleerd is.

## Simulation settings

Op dit moment is elke schuifbalk die u ziet in het Simulation settings scherm handmatig ingevoerd. In het onderstaande tekstfragment staat één schuifbalk.

row = **new** JPanel(**new** GridLayout(0,2));

name = **new** JLabel("Max bear population");

row.add(name);

combine = **new** JPanel();

text = **new** JLabel();

text.setText(String.*valueOf*(Hunter.*bear\_max\_population*));

combine.add(text);

slider = **new** JSlider();

slider.setMaximum(5000);

slider.setMinimum(0);

slider.setValue(Hunter.*bear\_max\_population*);

slider.addChangeListener(listener.**new** HunterBear(text));

combine.add(slider);

row.add(combine);

sliders.add(row);

In de huidige simulatie (V3.0) staan vijftien schuifbalken. Wanneer u nu het uiterlijk van die balken wilt veranderen zal dit vijftien keer aangepast moeten worden.

Dit komt omdat er niet bekend is hoeveel variabele waardes een actor heeft. Beter zou zijn om elke variabele waarde in een actor een klasse te maken met de waardes *minimum*, *maximum*, *standaard* en *huidig* en methodes om deze waardes aan te passen. Al deze klassen kan je dan in een lijst gooien en makkelijk door de hele lijst heen itereren. Bij elke iteratie zou je dan een schuifbalk kunnen maken. Dan hoeft er maar op één plek een verandering gemaakt te worden en dan veranderen alle schuifbalken.