# Challenge: Life Game

De beschrijving van deze challenge is kort/basic en is vooral bedoeld als startpunt om zelf tot een mooie oplossing te komen van het gevraagde. Laat dus vooral je eigen fantasie erop los om de challenge beter, mooier en leuker voor jezelf te maken.

(disclaimer: de info die je krijgt kan dus beperkt over komen, en dat kan. Die komt namelijk uit een brainstorm sessie.

Bij deze challenge hoort een klassediagram (zie content in de LifeGame directory), dat deels al gegeven is. Dit klassediagram is eens ontstaan na een brainstorm sessie met studenten over wat ze in het spel willen hebben. Het klassediagram is niet perse compleet. In het klassediagram staan ook comment met betrekking van gedrag van de applicatie. Lees die zeker wel even.

Tevens is er voorbeeldcode gegeven voor het tekenen/animeren van graphics in een windows forms applicatie. Zie het voorbeeldproject ‘BasicWindowsFormsGraphicsExample’ in de LifeGame directory. Het staat je natuurlijk helemaal vrij om de UI met een andere tool te maken (Unity3D?)

## Korte beschrijving challenge

Maak een spel/simulatie waarbij op er op een (game) wereld dieren rondlopen die hun eigen gang gaan/eigen leven leiden. Minimaal moeten de dieren zich op een eigen manier verplaatsen, zoeken naar eten en eten. Elk dier heeft health om zijn gezondheid bij te houden (zie verder het klassediagram). Er moet dus ook eten aanwezig zijn.

In het voorbeeld uit het klassediagram wordt de simulatie alleen nog maar bezet door Wolven en Schapen (en gras). De wolven lopen wat rond (dit lopen heeft specifiek gedrag voor de wolf) op zoeken naar eten. Wolven lusten graag schapen. Als ze een schaap kunnen waarnemen, dan achtervolgen ze die en eten hem op (sorry…). Natuurlijk kan een wolf niet oneindig ver kijken.

Schapen op hun beurt lopen rond op zoek naar lekker gras. En eten dat op als ze het vinden. Het opeten van gras duurt een tijdje.

De simulatie is turn-based. Elk dier krijgt om de beurt een spelbeurt waarin hij bepaald welke actie er uitgevoerd moet worden (en die wordt dan uitgevoerd), bijvoorbeeld:

* Zomaar een stukje lopen in een bepaalde richting of op een bepaalde manier, als er niets interessants in de buurt te zien is. Dieren lopen op hun eigen manier rond op de spelwereld.
* Rondkijken en als er iets interessants in de nabije omgeving aanwezig is, er dan een stukje naar toe lopen (bijvoorbeeld indien eten gedetecteerd wordt)
* Is er eten zo dicht bij dat er gegeten kan worden, dan eten (of juist alleen als er honger voor is)
* Vluchten als het spannend wordt?
* …??? etc

Een dier mag maar een aantal stappen verzetten per beurt (lees: pixels verplaatsen, zie gegeven filmpje). De verplaatsing in afstand is per soort dier verschillend. Een dier kan dus niet ineens naar een graspol toelopen aan de andere kant van de wereld in zijn beurt.

Ieder soort dier

* Heeft een aantal health punten
* Kan maar een bepaalde afstand (pixels) om zich heen kijken, zie MaxFeedSenseDistance (zie klassediagram). Betere naam misschien: ViewRange?
* Heeft een positie op de wereld
* Een eigen manieren snelheid van lopen voor het ‘zomaar een stukje lopen’
* Een eigen manier en snelheid van lopen om naar voedsel te lopen
* Een eigen loopje voor zomaar wat lopen
* Heeft honger of niet
* Vermenigvuldigd zich?
* Vormt groepjes?
* …??? etc

Zie ook het filmpje voor een functionele representatie van een implementatie (die niet erg mooi is 😊).

Ok, tot zover de inspiratie… Pas zelf aan waar je wilt… Overleg wel met de docent over wat je wilt aanpassen en waarom. De gamewereld is bijvoorbeeld nogal basis….

Natuurlijk moet je bij het implementeren steeds een klassediagram kunnen tonen dat up-to-date- is met je oplevering. En moet je code getest zijn. En je kun use cases maken bij de applicatie.