

# OVERZICHT CELLULAIRE AUTOMATA

input:

- dimensies
- radius (vorm?)
- wat zijn de buren
- regels
  - (meerdere toestanden)
  - normale
  - randvoorwaarden
- begintoestand
  - standaard alles dood

Klassen:

Generic\_CA( aantal\_dimensies, startgrid,Burenlist):

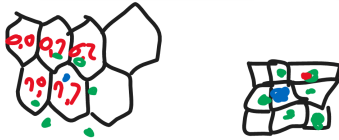
drie functies:

\_\_init\_\_()

evoluur()

regels()

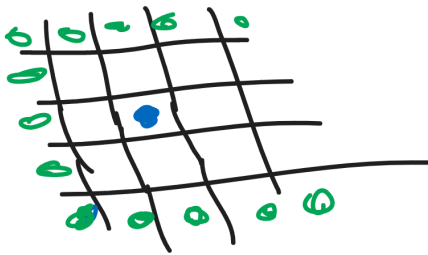
Proces



Hexagon naar matrix geeft problemen in het rechtsboven vakje dus werkt nite

idee om de waarde van cellen op te slaan in de  $N+1$  rij van de vector die die positie aangeeft.

Gesprek over regels. Regels hangen van het aantal burens.

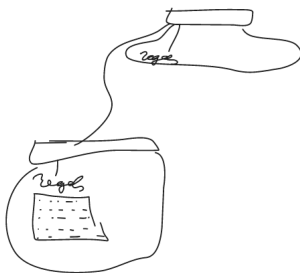


wat als je z'n buurpatroon hebt

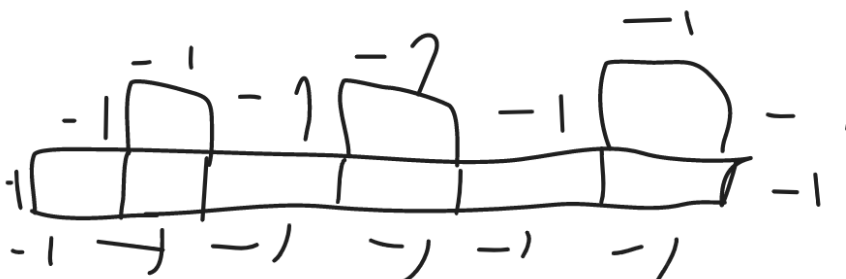


of wat als je grid z'n vorm heeft

De regels zouden met een parent-child relatie voor elk voorbeeld doorgevoerd kunnen worden



de randvoorwaarden zouden in de regels opgenomen kunnen worden door alle randvakjes de toestand -1 te geven.



We weten nog niet hoe we dingen die niet in een  $n$  dimensionaal grid passen willen doen.

Optimalisatie check niet vakjes met -1 in evolueer() want die doen niet mee

hoe voorkom je dat je buiten de matrix checkt?

Gebruik numpy om snel met hogere dimensies matrices te rekenen

de grote  $O(N)$  van evolueer method moet echt klein zijn en we willen meer optimalisaties