**To Do**:

* Speelveld aanmaken en daarin dingen toevoegen → Numpy
* Prijs per extra vierkante meter berekenen (welk huis het gunstigst?)
  + Formules maken voor ieder huistype. Vervolgens kunnen deze in het algoritme worden toegepast, om te kijken of een verplaatsing van een huis, een netto hogere waarde oplevert.
* Aantal meters vs. Prijs
* **Algoritme idee:** begin met een random configuratie op het speelveld, die wel aan de eisen voldoet. Pas een formule toe om te berekenen wat de totale waarde van het veld is. Ga vervolgens de woningen op het veld af om te kijken of een verandering op het veld kan leiden tot een netto waardevermeerdering. Begin bij huis 1, stel een kleine verandering voor (bijvoorbeeld een eengezinswoning één plaats opzij schuiven) en kijk of deze legaal is. Als deze inderdaad legaal is, bereken dan met de gegeven formule of de stap ook tot een waardevermeerdering van de wijk leidt. Als dit zo is, zet de stap, als dat niet zo is, zet de stap dan niet. Ga vervolgens door met huis 2, huis 3, etc. Als alle huizen zijn behandeld, begin je weer bij huis 1. Op een gegeven moment zal de situatie ontstaan dat geen enkele verandering meer zal leiden tot een waardevermeerdering: je hebt een mogelijke oplossing gevonden. → onthoud deze opstelling en zijn waarden.
* Je kan dit voor verschillende random situaties doen, die waarschijnlijk allemaal tot een andere eindsituatie zullen leiden.
* Je kan ook met bepaalde patronen beginnen: als je hebt ontdekt of verwacht dat een bepaald patroon van huizen tot een betere oplossing zal leiden, kan je dit ‘hardcoden’ of als uitgangssituatie nemen.
  + Hill climber → lokaal maximum
  + Uitdaging: hoe vind je het globaal maximum?
* **Eerste model** maken, zodat we in ieder geval iets hebben dat (enigszins) werkt. Dit model zal inhouden, dat de kavels in de gegeven procentuele verhoudingen op het beschikbare land worden geplaatst. Hierdoor ontstaat dus een configuratie die in ieder geval aan de gestelde eisen voldoet (let ook op de minimale 4:1 verhouding van het oppervlaktewater), maar nog niet optimaal is; zie het als een minimale oplossing. Schrijf een code om dit probleem op te lossen.
  + Maak een speelveld van blokken 0,5 x 0,5 meter en zet de huizen gewoon op de plattegrond.
  + Geeft een base-line om mee te vergelijken: hoeveel beter is ons goede algoritme?

**Variabelen**

0 = Leeg

1 = Water

2 = Eengezinswoning

3 = Bungalow

4 = Maison

5 = Vrijstand

**Eisen:**

* Speelveld is 24.000 m2 (96.000 vakken van 0.5 x 0.5 m)
* 20% oppervlaktewater 4800 m2 (19.200 vakken van 0.5 x 0.5 m)
* 20 variant: 12 eengezinswoningen, 5 bungalows, 3 maisons
* 40 variant: 24 eengezinswoningen, 10 bungalows, 6 maisons
* 60 variant: 36 eengezinswoningen, 15 bungalows, 9 maisons

**Ideeen:**

* vrijstandsklasse(?)

**Formules:**

* **Prijs** = (evt. per soort) sommatie per huis = beginprijs + (beginprijs \* aantal extra vrijstand \* bijbehorend percentage)
* **Vrijstand** = sommatie per huis = verplichte vrijstand + aantal extra vrijstand