**IDEA punten H@cherman**

**Link to github:** https://github.com/Stof95/Hackerman.git

Infrastructuur: 5

We hebben een file main.py die de user vraagt om hoeveel huizen hij wilt plaatsen, met welk algoritme hij dat wilt doen en eventueel hoeveel iteraties hij wilt uitvoeren. Afhankelijk van de input, roept main.py een algoritme aan uit de map met algoritmes. De algoritmes roepen op hun beurt weer functies op uit helpers.py die door meerdere algoritmes gebruikt worden. Hierin staat bijvoorbeeld een functie die de score van een lijst met alle buildings kan berekenen. Bovendien staan de objecten zoals de huizen en water opgeslagen in classes.py. Als laatst hebben we een file die zorgt voor de visualisatie van de map.

Datastructuur: 5

Onze data (de huizen) staan in een file classes.py, waarbij een class voor eengezinswoning, een class voor de bungalow, een class voor de maison, een class voor water en een map-class. Door deze structuur kunnen we makkelijk manipulaties uitvoeren op onze data. Bovendien kunnen we met een scorefunctie die in helpers.py staat, de score per huizensoort in classes.py aanroepen en zo de score van de hele map uitrekenen. Dit doen we door voor elk huis in een lijst met alle huizen, de afstanden tot het dichtste huis te bepalen met een functie calculate\_distance.

Experimentatie: 3

We hebben een heuristiek toegevoegd aan het algoritme Hillclimbers, dat we ‘expanding universe’ noemen. Hierbij zetten we de kleinste huisjes in het midden van de map, de bungalows en maisons via een parametervoorstelling hieromheen. Daarna gaan we met een aangepaste Hillclimber, stukje bij beetje de huisjes als een uitdijend heelal richting de randen laten lopen, waarbij we steeds kijken of de score nog toeneemt. Dit doen we omdat we denken dat vrije ruimte die de maisons delen meer opleveren dan vrije ruimte die kleine huisjes delen.

Algorime: 4

Het random algoritme werkt. Hillclimber met expanding universe en greedy.