

# World Conquest

## De werelden

Voor het maken van een ‘wereld’ zijn er 2 algoritmes beschikbaar. Het eerste is een kolomgeïntereerd, het tweede is gebaseerd op startpunten en een gewenste dekkingsgraad. Meer gedetailleerde info over de algoritmes s terug te vinden in de code.

The output van het eerste algoritme kan er dan als volgt uit zien :

[illegible]

Een sterretje wil zeggen dat het vakje deel uit maakt van de wereld, een spatie stelt dan een niet bewoonbaar deel van de wereld voor.

Een voorbeeld van het tweede algoritme :

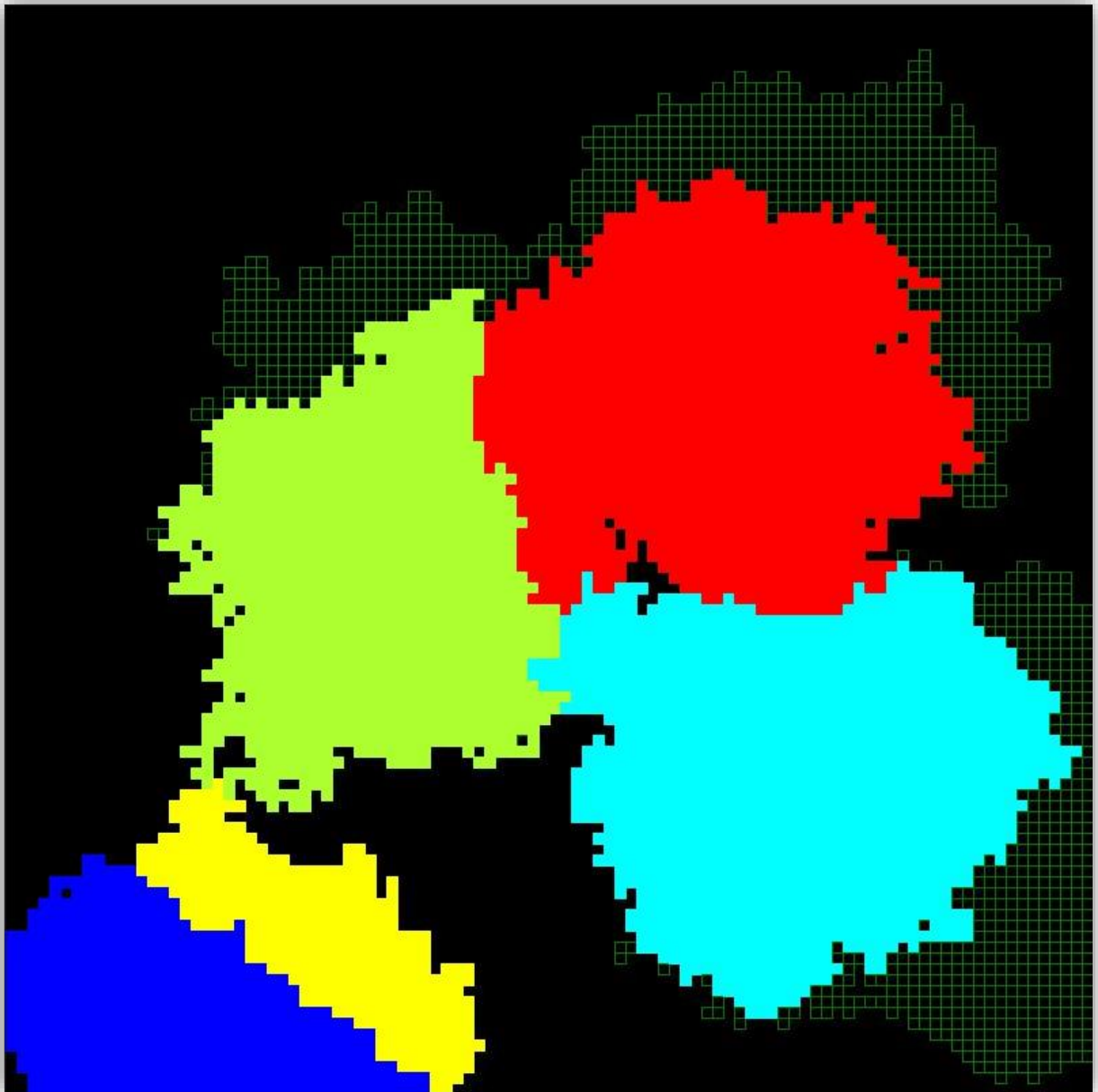


## De verovering van de wereld

Ook hier zijn er verschillende algoritmes uitgetoetst. We tonen het resultaat voor elk algoritme op een wereld van 100 op 100 hokjes met een dekkingsgraad van 60 %, waarbij er telkens 25000 beurten zijn uitgevoerd.

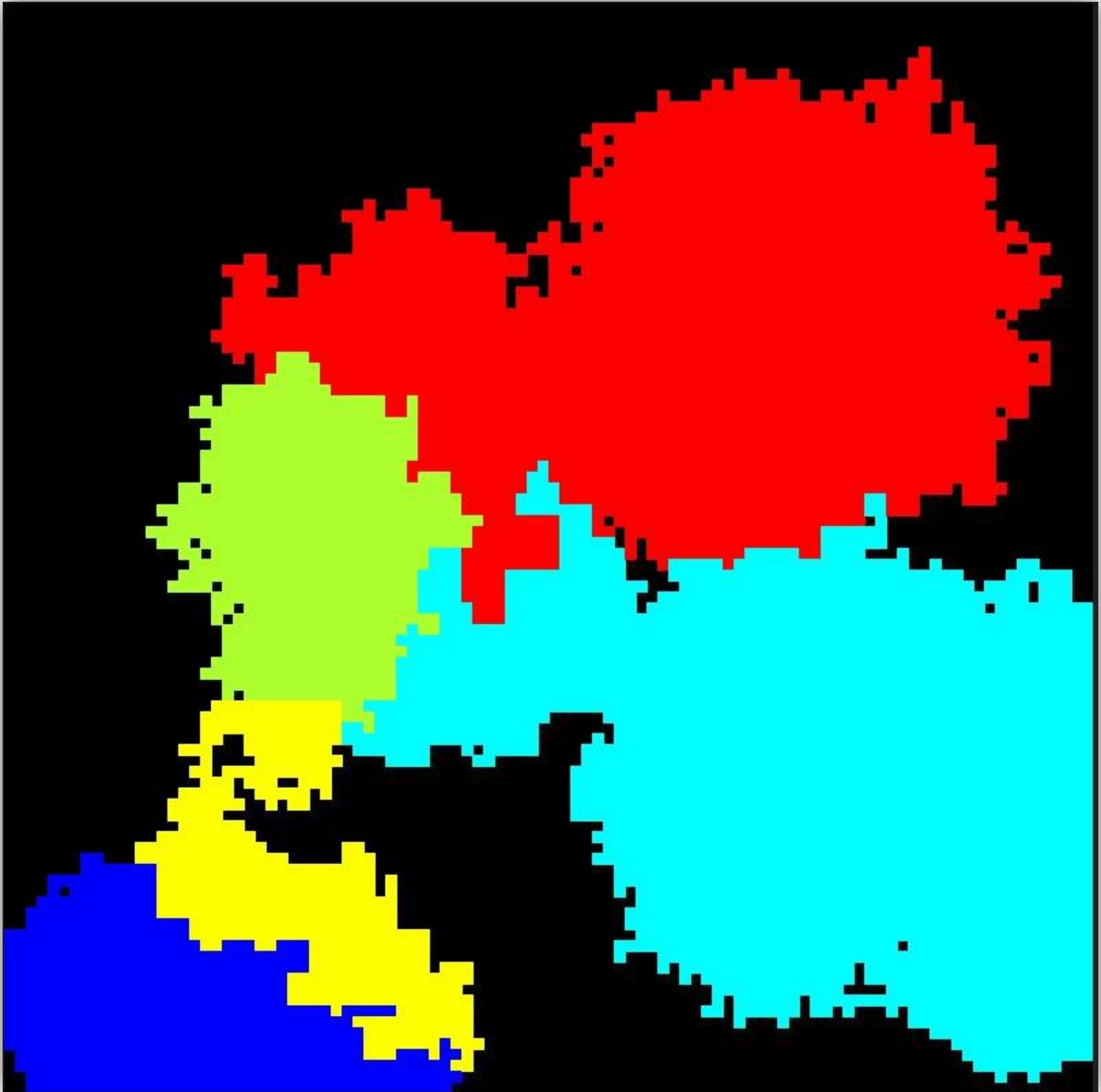
## Conquer1

```
/*  
 * nEmpires indicates the number of empires who will try to conquer the world  
 * it is a turned based algorithm, every turn each empire will try to expand  
 * on each turn,  
 *     for each empire a random location owned by the empire is selected  
 *     a new location is selected by picking a random adjacent location  
 *     if that location is within the boundary of the map and not occupied by an empire, the location is assigned to  
 *     the empire  
 */  
public int[,] Conquer1(int nEmpires, int turns)...
```



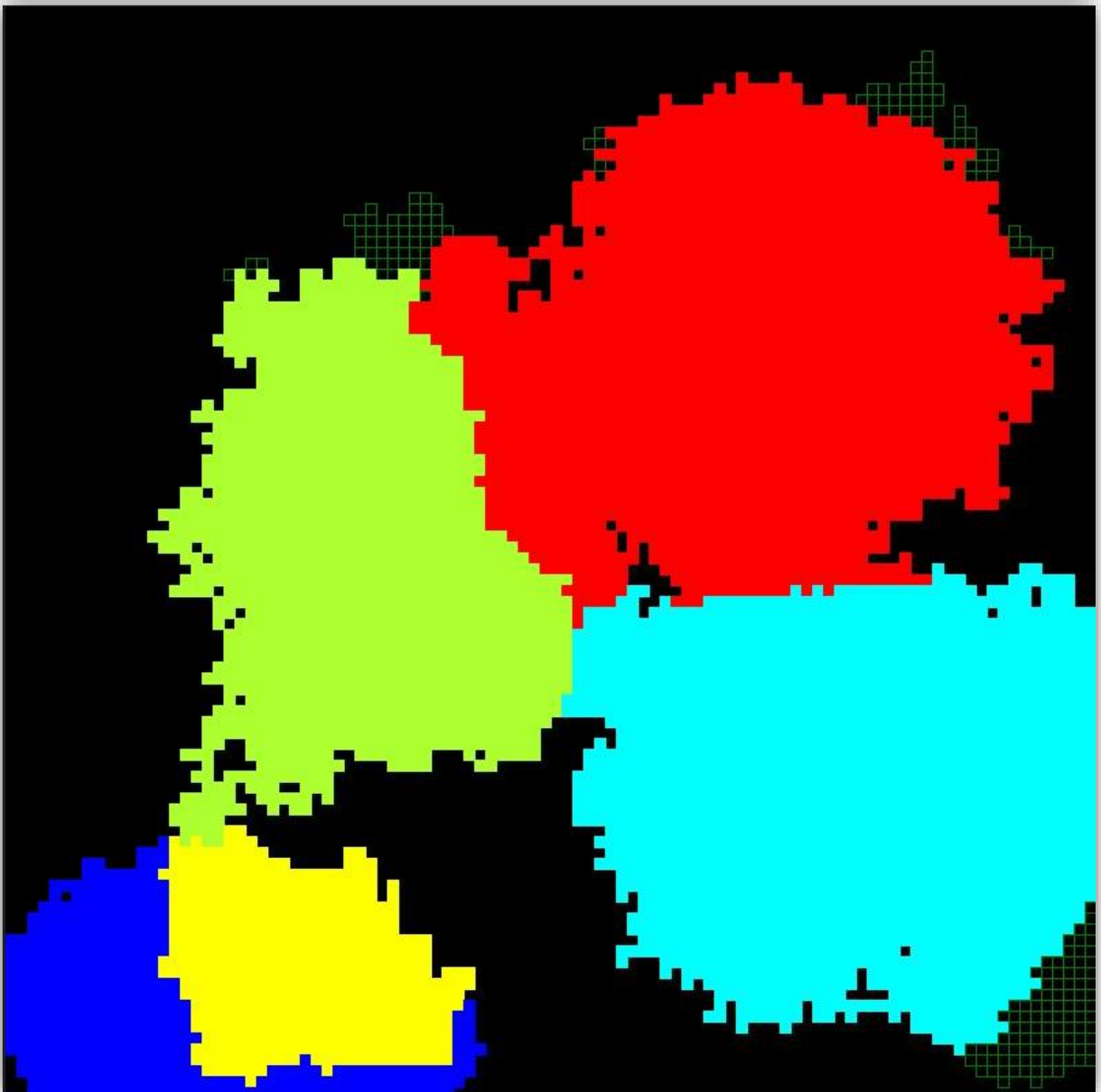
## Conquer 2

```
/*
 * nEmpires indicates the number of empires who will try to conquer the world
 * it is a turned based algorithm, every turn each empire will try to expand
 * on each turn,
 *     for each empire the locations with the most free adjacent locations are selected
 *     out of this list a random location of the empire is selected
 *     a new location is selected by picking a random adjacent location
 *     if that location is within the boundary of the map and not occupied by an empire, the location is assigned to the empire
 */
public int[,] Conquer2(int nEmpires,int turns){...}
```



## Conquer 3

```
/*
 * nEmpires indicates the number of empires who will try to conquer the world
 * it is a turned based algorithme, every turn each empire will try to expand
 * on each turn,
 *     for each empire a random location owned by the empire is selected
 *     a new location is selected by searching a free adjacent location
 *     if that location is within the boundary of the map and not occupied by an empire, the location is assigned to the empire
 */
public int[,] Conquer3(int nEmpires, int turns){...}
```



## Opdracht

- Creëer een tiental werelden en sla deze op in de databank. Voorzie ook een naam voor elke wereld, het type (welk algoritme is gebruikt), de afmetingen (maximale x- en y-waarden) en de dekkingsgraad.
- De code voorziet momenteel 3 conquer-methodes waarbij 3 verschillende algoritmes worden toegepast maar telkens voor alle 'empires'. Pas de code aan zodat voor elk 'empire' apart kan bepaald worden welk algoritme er moet worden gebruikt.
- Voor de analyse van de verschillende algoritmes, moet er data worden verzameld door verschillende algoritmes uit te proberen in verschillende werelden.
  - Bereken de grootte van elk 'empire' (aantal vakjes + percent van de wereld) en het algoritme dat werd gebruikt. Zorg dat je minstens voor elk algoritme één 'empire' hebt.
  - Voer deze berekening uit voor verschillende startposities van elk 'empire' in éénzelfde wereld.
  - Voer de berekeningen ook uit voor verschillende werelden.
  - Elk van de resultaten moet worden opgeslagen (gebruikte wereld, voor elk 'empire' het algoritme en de bezetting).
  - Bereken ook statistische gegevens voor elk algoritme (gemiddelde bezetting voor verschillende startposities en werelden).

Zorg voor duidelijke en flexibele code !

Zorg er steeds voor dat alles zo efficiënt mogelijk verloopt (denk async) !

De broncode is terug te vinden op : <https://github.com/tvdewiel/ConsoleAppSquareMaster>

De databank die moet worden gebruikt is MongoDB

Enkele leestips :

<https://refactoring.guru/design-patterns/factory-comparison>

<https://refactoring.guru/design-patterns/strategy>

Veel success !