

Battlesimulator

Project 2

Lars Posthumus & Musaab Azawi | Groep 16 | 27-03-2020

# VERANDERLINGEN EN IMPLEMENTATIES

## Big o

Tijdens het project is de big o analyse toegepast door alle for loops in Game.ccp na te lopen. Hierdoor is een beeld geschetst welke Algoritmes verbeterd moeten worden. Een van de Algoritmes is insertion\_sort\_tank\_health.

## Mergesort

Doormiddel van een Big O heeft Insertionsort een complexiteit van O (n²), deze methode is niet efficiënt. Om de algoritme complexiteit te verlagen en de Battlesimulator te optimaliseren zal Mergesort geïmplementeerd worden. Deze algoritme heeft een big O(n log n).

## Grid

Er is geconstateerd dat in de update methode een stuk code aanwezig was die de tanks per frame controleert. Deze update methode itereerde door elke element(tank) door tank vector. Om het trage collation detectie algoritme in de software te versnellen, is er een Grid geïmplementeerd. Deze algoritmische verbetering doormiddel van een drie dimensionele vector wordt door een fragment van alle tanks geïtereerd.

## particlebeam

De particlebeam is toegepast op de Grid. Deze verandering zorgt ervoor dat deze niet door de hele tank vector maar per fragment, net als tanks op de Grid.

## Methods toegevoegd

* updateGrid
* sortTanks
* erase\_from\_vector
* mergeSort

## Work & span

De updaterocket methode maakt gebruikt van nestend for loop. Deze methode heeft complexiteit O(n²). Deze methode is omgezet in threads. Door het verdelen in threads is er nog maar een for loop en heeft complexiteit O(n). Dit betekend dat deze sneller is dan de originele methode.