

Stage 2022
-
Gelles Julien

Compte-Rendu n°7
-
Semaine du 30/05 au 03/06



Travaux réalisés :

1. Premier utilisation d'un .dat

Pour commencer l'utilisation d'un premier fichier .dat, j'ai décidé d'en recréer un simplifier et d'écrire un petit script afin de le transformer en matrice pour pouvoir ensuite l'exploiter.

Voilà un aperçu de ce que cela pouvait donner :

```
ViewerWeb > dat > Test2 > JS xml_parser.js > ...
1  const fs = require('fs');
2
3  const data = fs.readFileSync('simu2.dat', 'utf8');
4
5  let lines = data.split('\r\n')
6
7  for (let i = 0; i < lines.length; i++) {
8      lines[i] = lines[i].split('\t');
9      for (let j = 0; j < lines[i].length; j++) {
10         lines[i][j] = parseFloat(lines[i][j]);
11     }
12 }
13
14 console.log(lines);
15
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Copyright (C) Microsoft Corporation. Tous droits réservés.

Testez le nouveau système multiplateforme PowerShell <https://aka.ms/pscore6>

PS C:\Users\Julien\Desktop\Stage> cd .\ViewerWeb\dat\Test2\
PS C:\Users\Julien\Desktop\Stage\ViewerWeb\dat\Test2> node .\xml_parser.js

```
[
  [ -50, -50, 15 ], [ -50, -25, 10 ],
  [ -50, 0, 5 ], [ -50, 25, 25 ],
  [ -50, 50, 5 ], [ -25, -50, 50 ],
  [ -25, -25, 15 ], [ -25, 0, 5 ],
  [ -25, 25, 0 ], [ -25, 50, 40 ],
  [ 0, -50, 45 ], [ 0, -25, 5 ],
  [ 0, 0, 40 ], [ 0, 25, 15 ],
  [ 0, 50, 5 ], [ 25, -50, 50 ],
  [ 25, -25, 35 ], [ 25, 0, 5 ],
  [ 25, 25, 5 ], [ 25, 25, 10 ],
  [ 50, -50, 15 ], [ 50, -25, 10 ],
  [ 50, 0, 5 ], [ 50, 25, 30 ],
  [ 50, 50, 25 ]
]
```

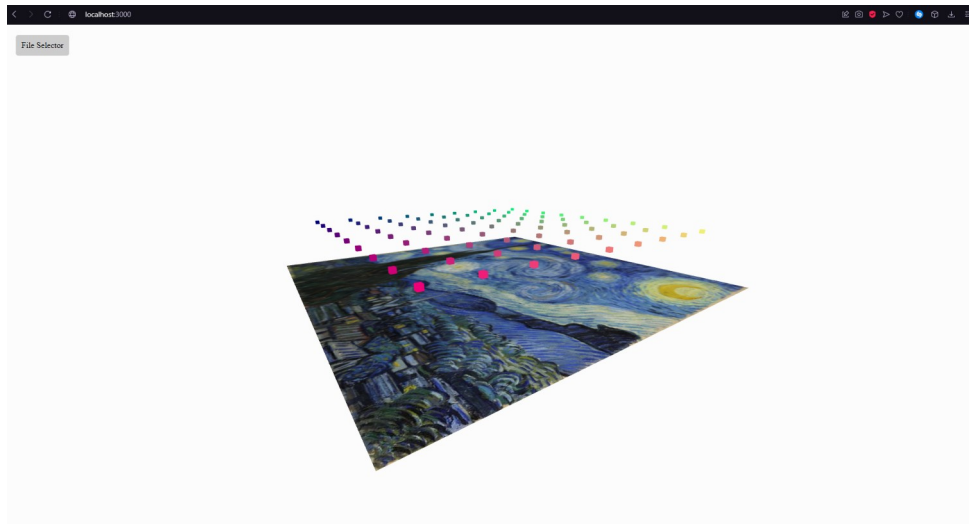
PS C:\Users\Julien\Desktop\Stage\ViewerWeb\dat\Test2> |

Transformation d'un .dat en matrice

2. Premier essai de multi-meshes

Dans l'optique de créer une heightmap plus tard, j'ai avant tout souhaité m'essayer au fait de placer beaucoup plus de meshes sur le rendu qu'un simple plan.

J'ai donc créé un champ de cube, avec lesquelles j'ai pu jouer sur la couleur, la taille et les positions.



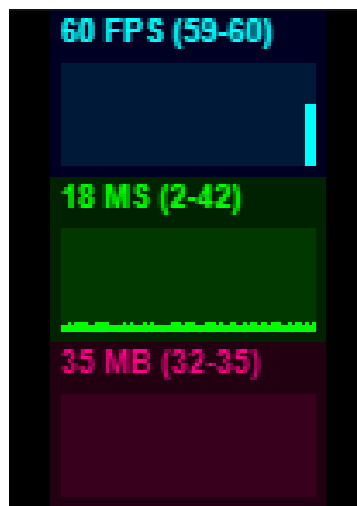
Premier essai multi-meshes

3. Premier essai de multi-meshes

Enfin, j'ai ajouté une petite fonctionnalité à ce moment là pour d'ores et déjà penser aux soucis de performances que l'on pourrait avoir dans le futur.

Cela est une fonctionnalité que l'on retrouve avec *three.js* qui nous permet d'afficher des statistiques de la page en temps réel concernant le rendu.

J'ai donc décidé d'y afficher les *fps*, les *ms* ainsi que la *consommation mémoire* comme ci dessous :



Affichage stats