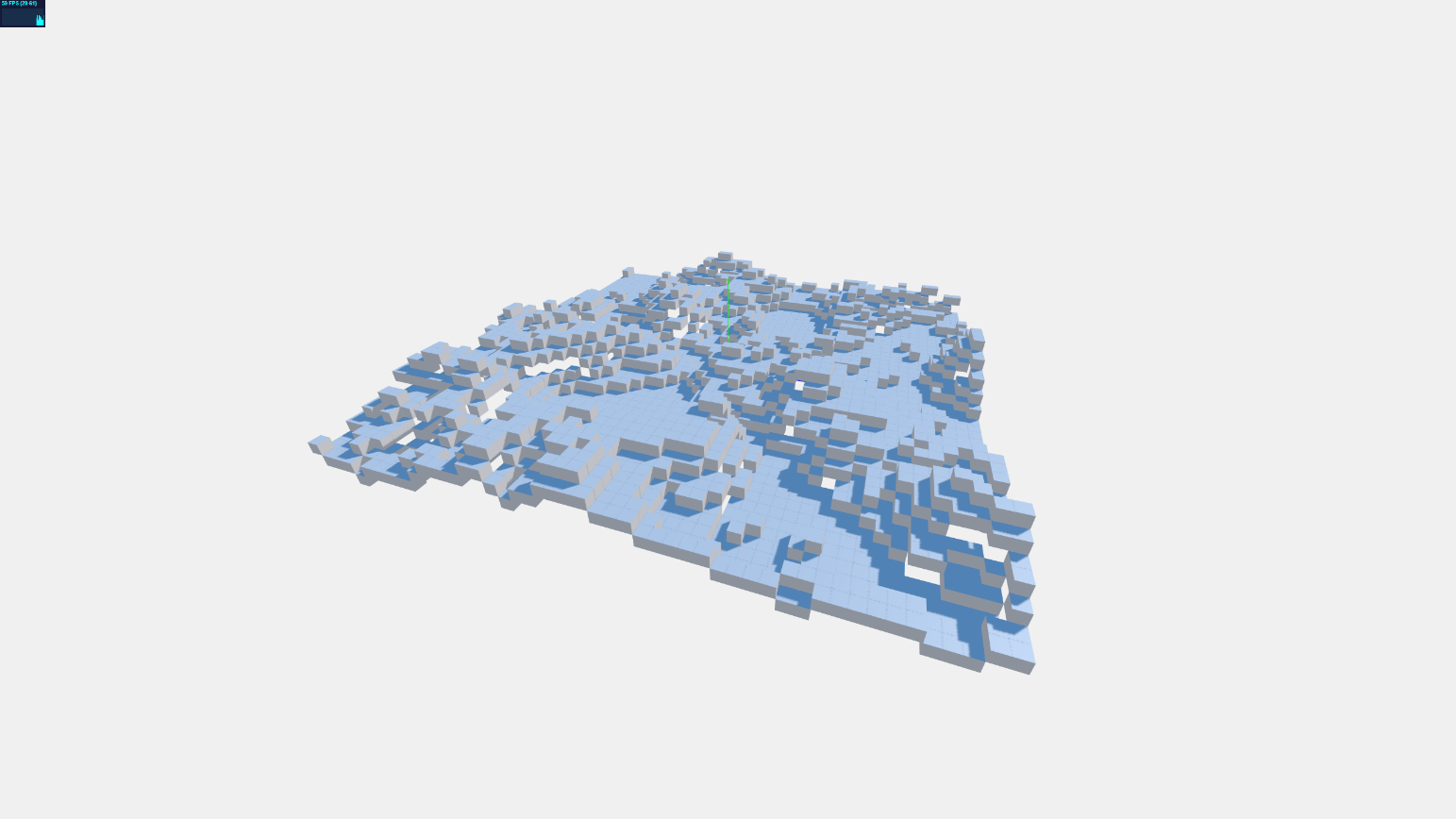
**DAY 1 – 28/10/2019**

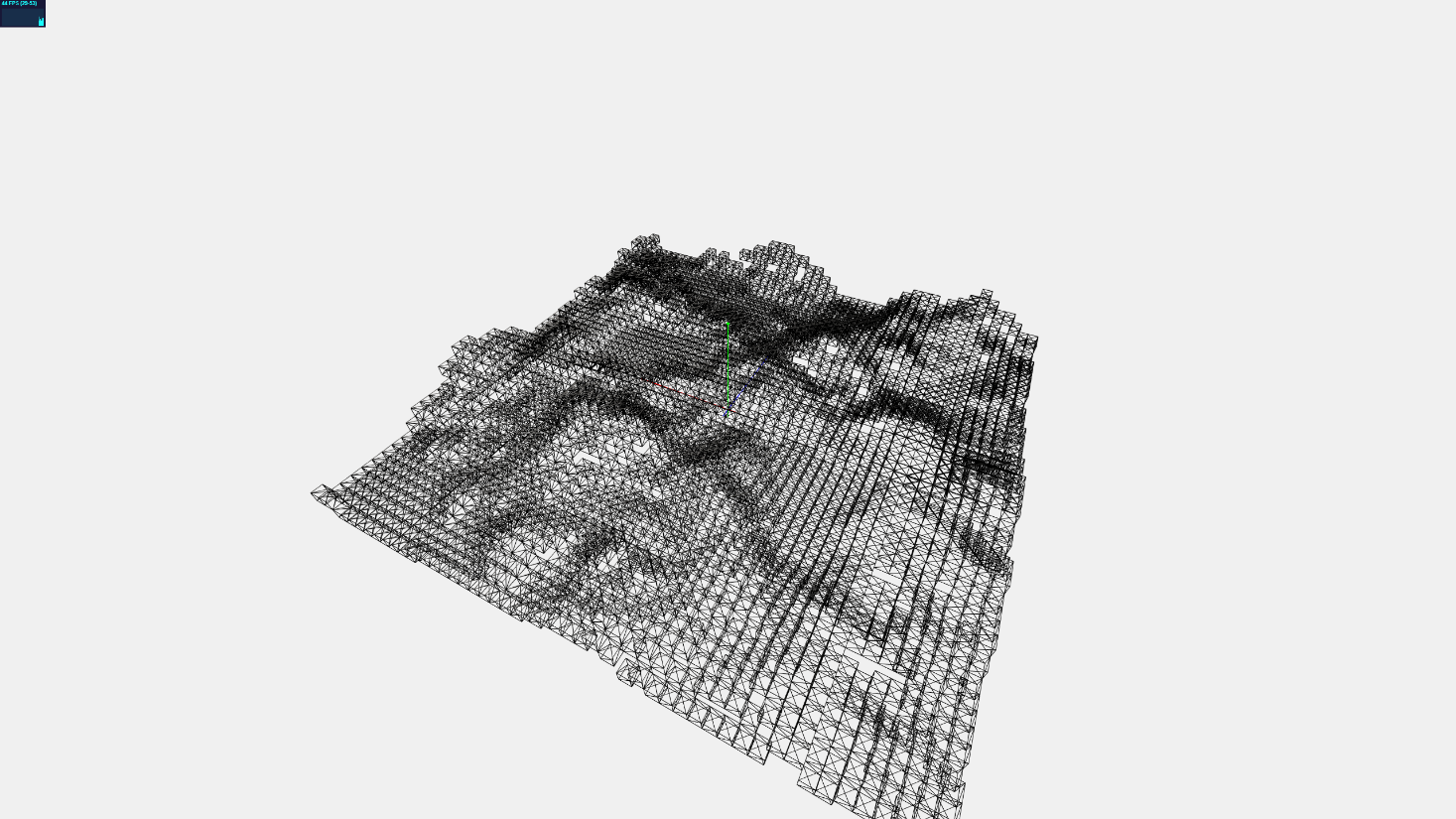
**Mansi:**

* Visualizzazione della consegna;
* Visualizzazione del codice e piccoli test;
* Sperimentazione del codice “heightmap”; implementazione di un primo prototipo di algoritmo di generazione del terreno.



HeightMap usata

*Screenshot primo prototipo*



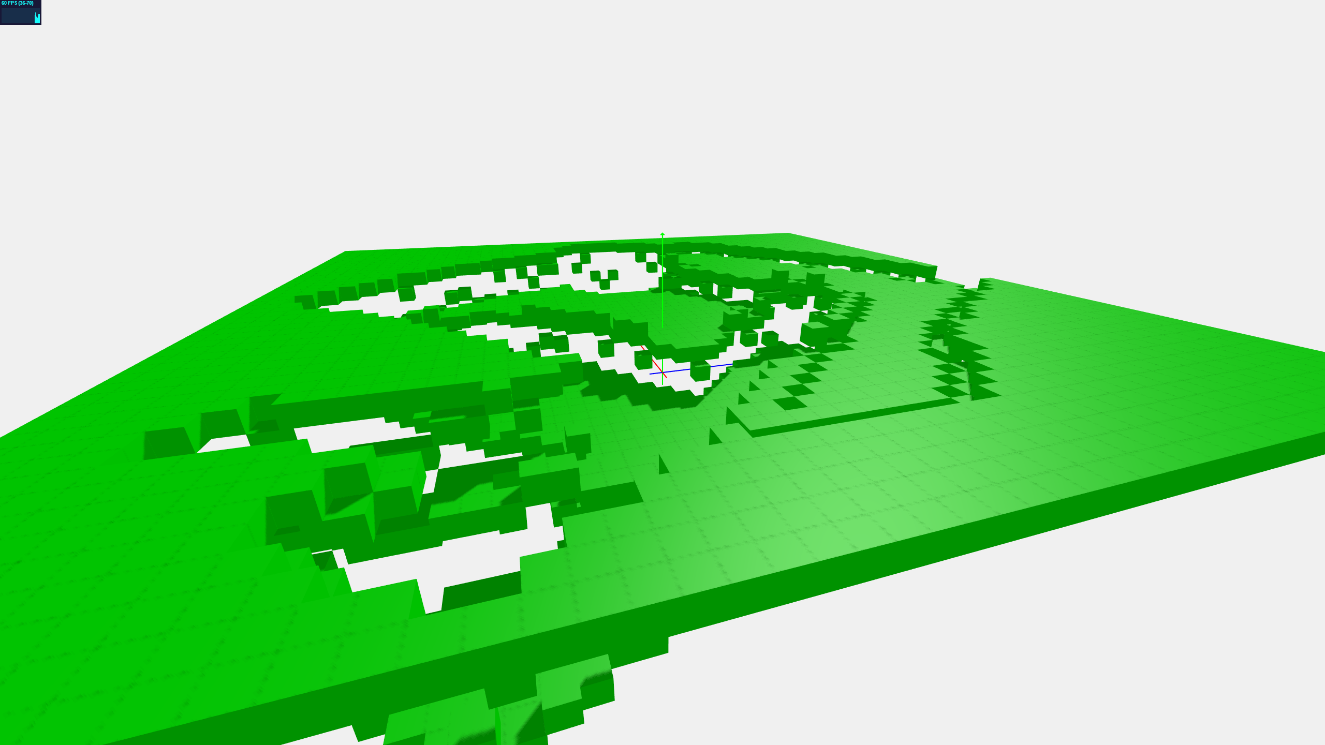
*Screenshot primo prototipo  
vista wireframe*

* Refactoring del codice (creazioni funzioni per inizializzare camera, scena, renderer, lightsystem etc. etc.)
* Commit del codice su GIT

**DAY 2 – 29/10/2019**

**Mansi:**

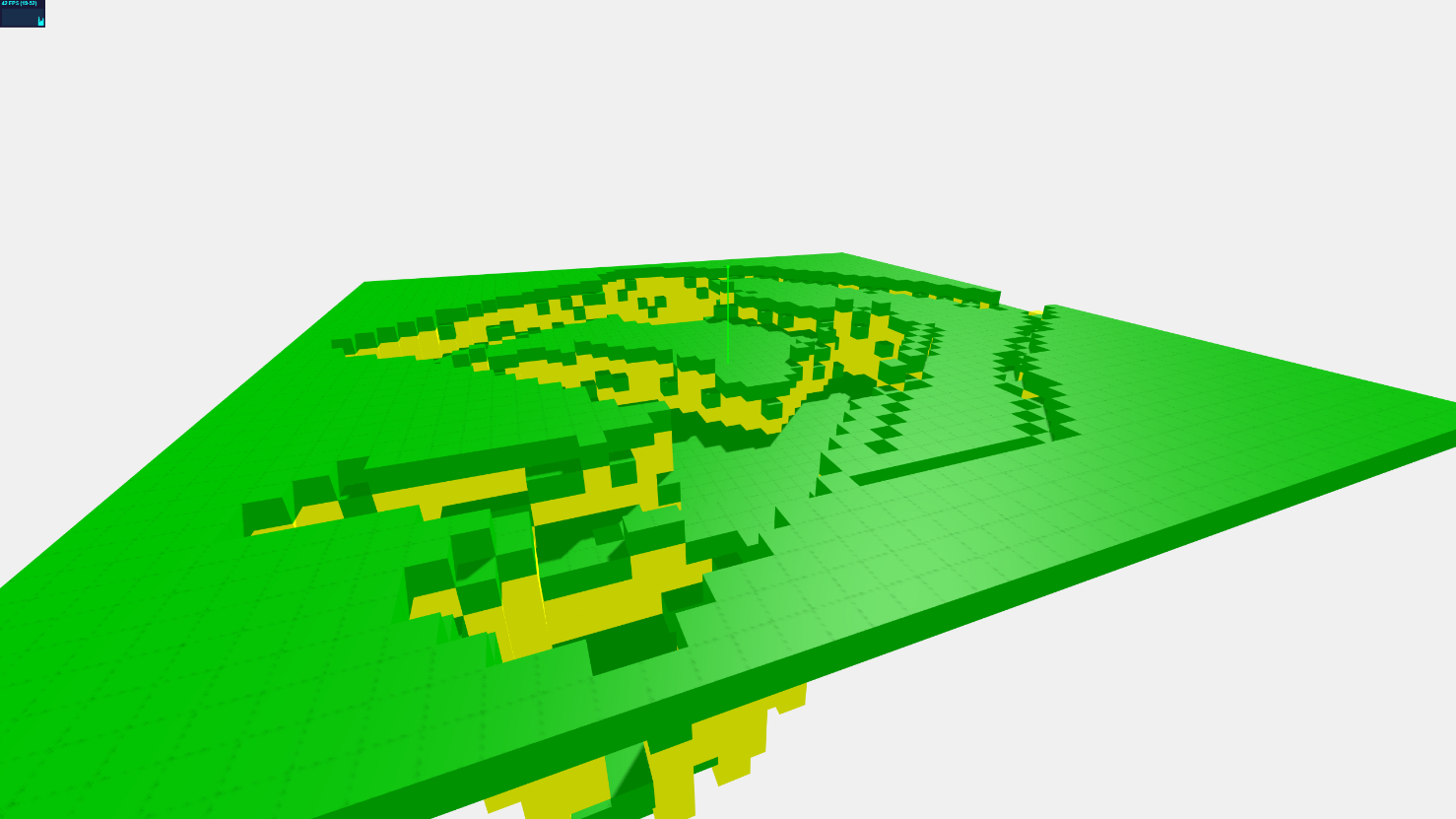
* Ulteriori sperimentazioni con l’algoritmo di generazione del terreno, sperimentazione e analisi del primo problema riscontrato: Il terreno non risulta uniforme perchè ci sono dei “fori” nei cambi di altezza troppo bruschi, da risolvere.

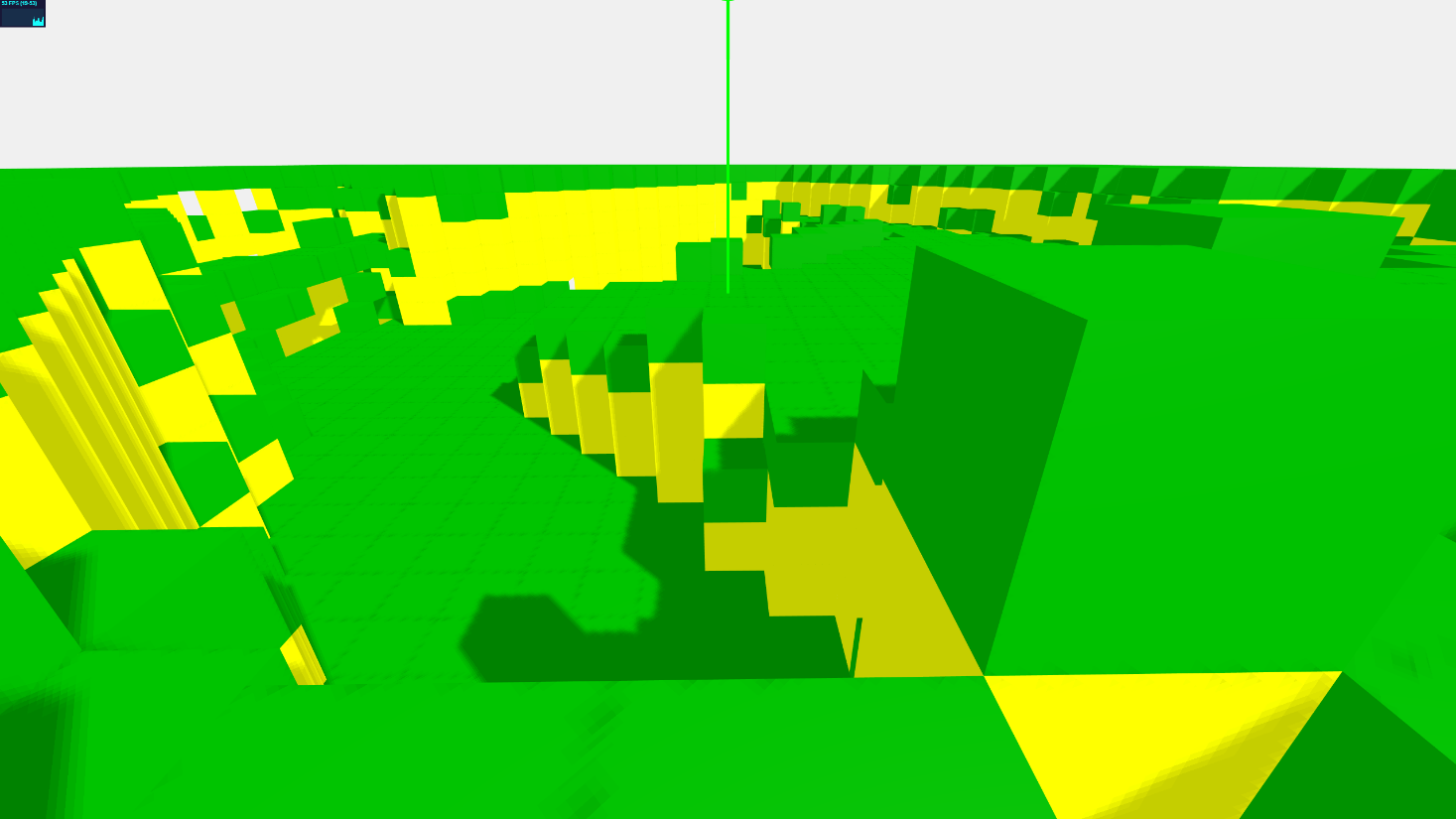


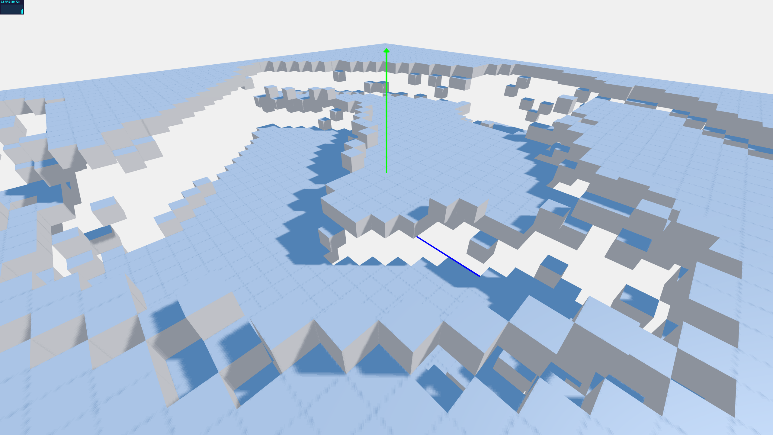
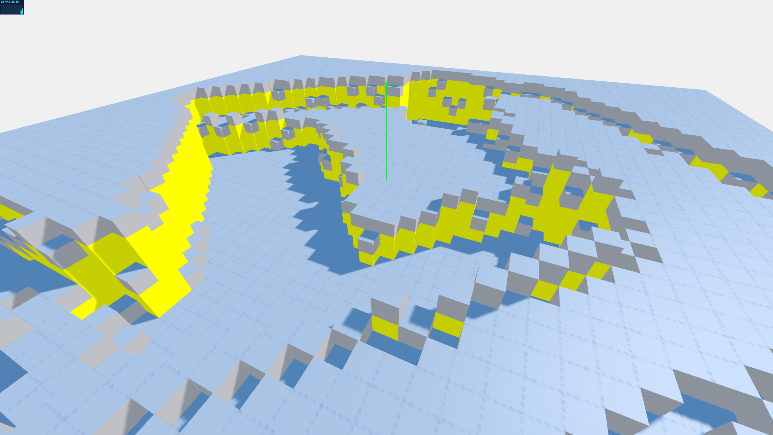
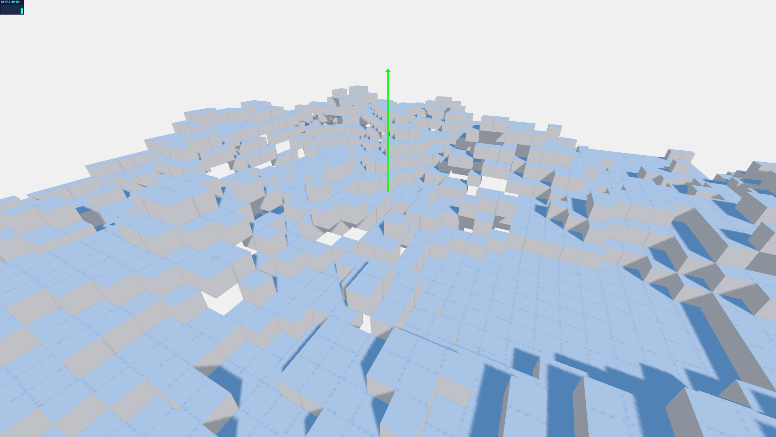
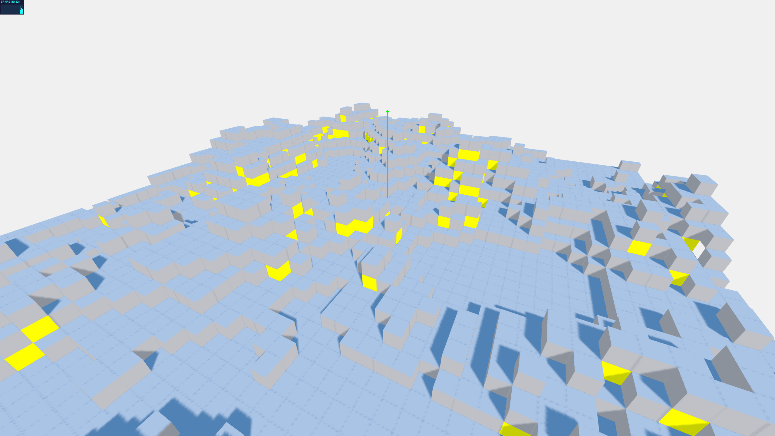
*Heightmap utilizzata, creata al fine di accentuare il problema riscontrato*

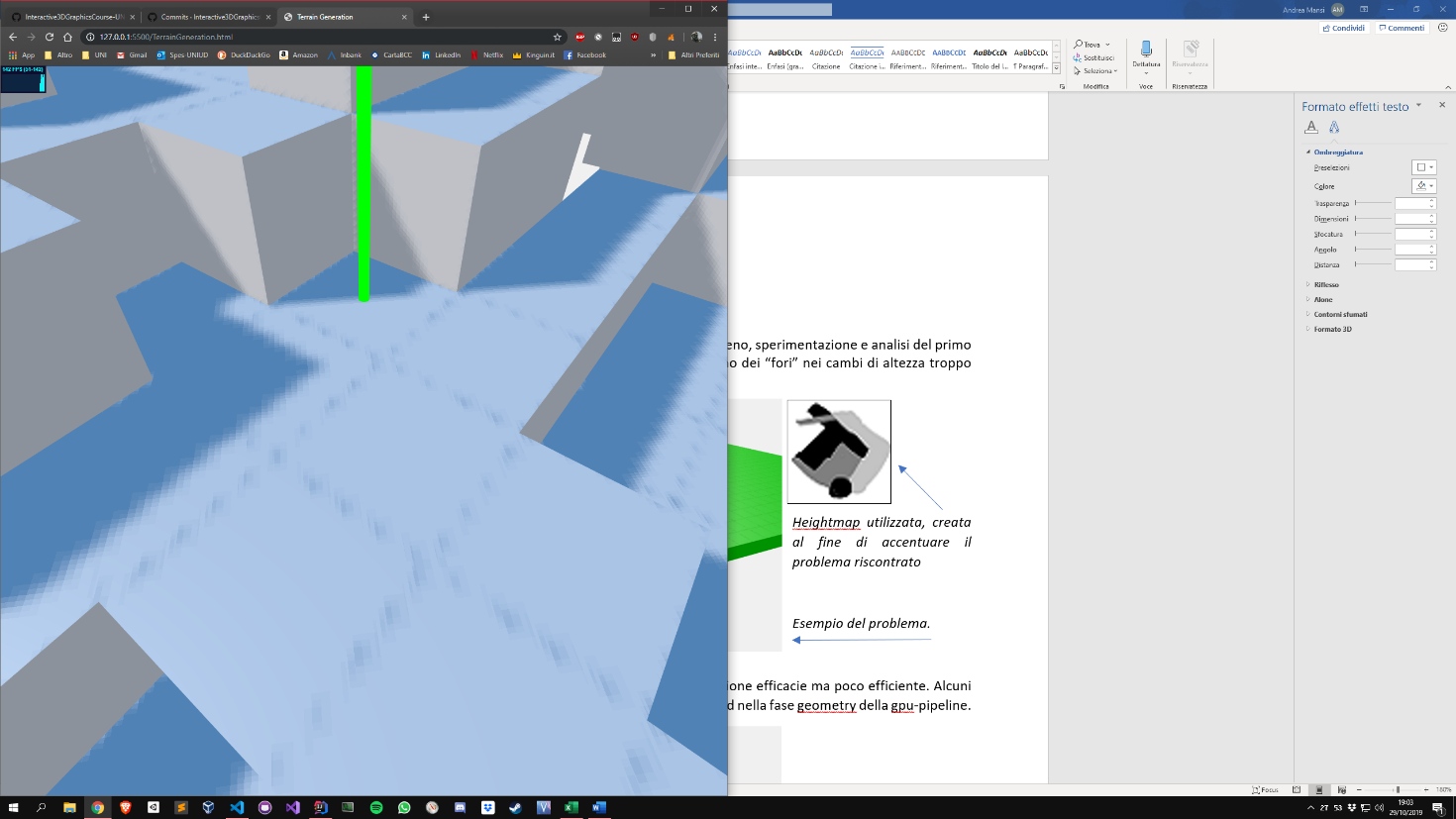
*Esempio del problema.*

* Prima sperimentazione che tenta di risolvere il problema, soluzione efficacie ma poco efficiente. Alcuni cubi mancanti vengono inseriti più volte portando ad un overhead nella fase geometry della gpu-pipeline.



Esempio del terreno “fixato”  
In giallo i cubi aggiunti successivamente alla prima fase di generazione del terreno.

* Seconda Implementazione che risolve senza overhead la problematica dei dislivelli elevati:
* Pulizia del codice, Commit su GIT
* Altri problemi riscontrati: Pesante impatto sulle performance da parte del sistema di ombre e illuminazione, artefatti grafici sui “bordi dei cubi”. 40-50 fps su una RTX 2070 – i7 9700K.



Artefatti grafici da parte del sistema di ombre/illuminazione

ADD Schemi a mano

**DAY 3 – 30/10/2019**

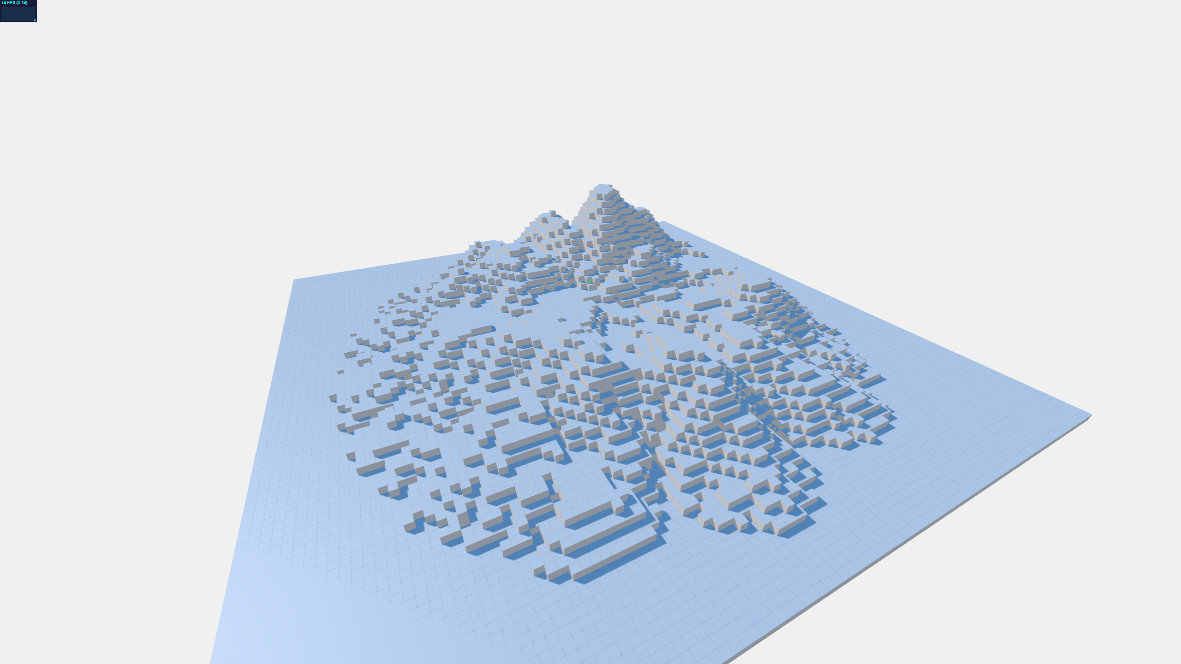
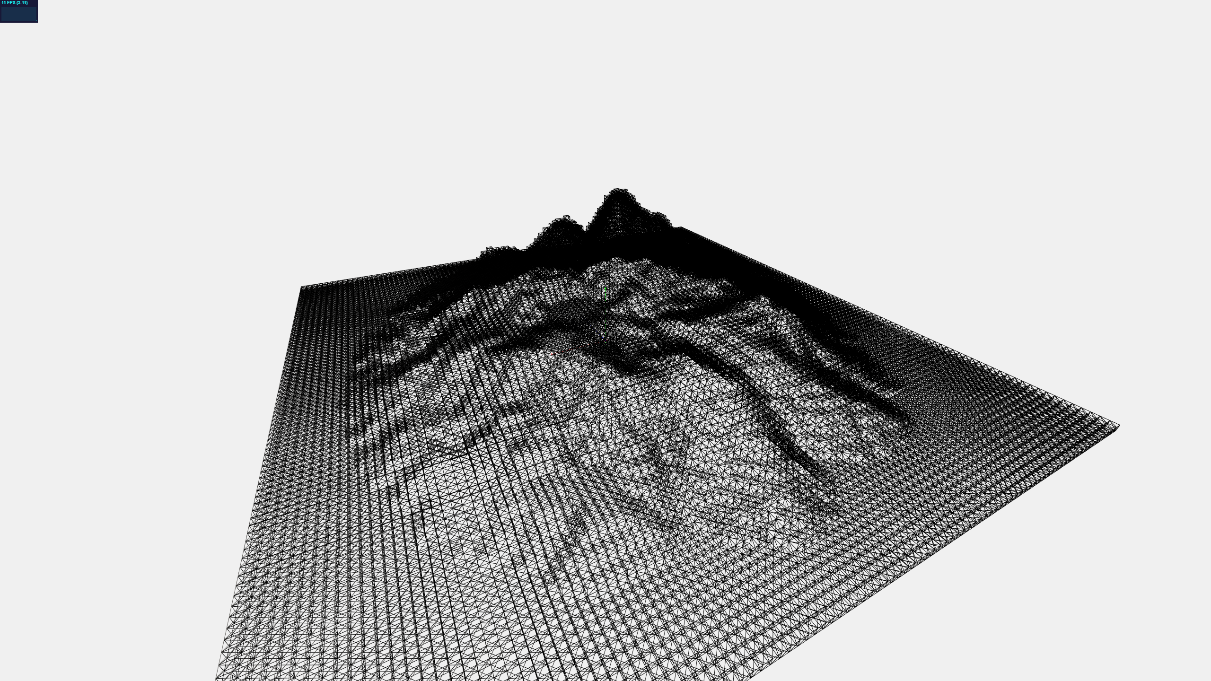
#######################################################################################**Meeting di gruppo**, progettazione della scena di base e suddivisione dei compiti.

Durante i primi giorni di lavoro i compiti sono stati suddivisi in questo modo:

* + Relazione: Daniele
  + Generazione del terreno: Mansi
  + Animazioni, modellazione: Daniele
  + LightSystem & Visuals: TODO
  + Spawning di oggetti nella scena: TODO
* Scena di base (minima da soddisfare i requisiti della consegna)
  + Generazione del terreno con:
    - Terreno sui blocchi della superficie;
    - Roccia sui blocchi sottostanti (visibili dall’esterno)
  + Erba e Cespugli e alberi posizionati randomicamente sulla superficie:
    - Con animazione
    - Interattività con la luce (ombre)
  + Edificio in centro alla scena
    - Con animazione
  + Animazione della telecamera

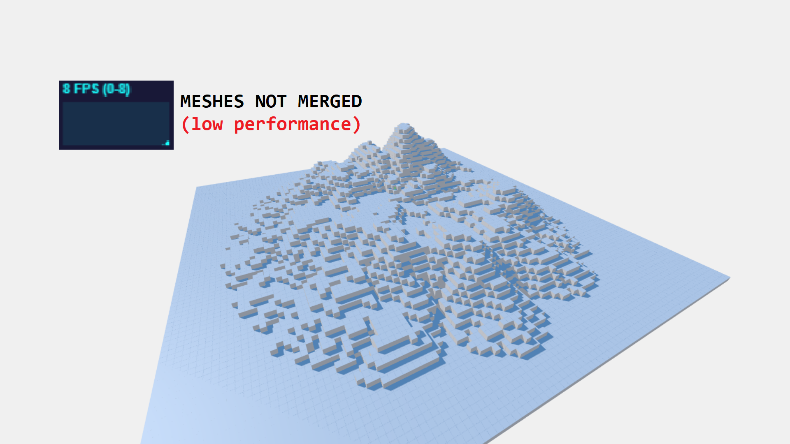
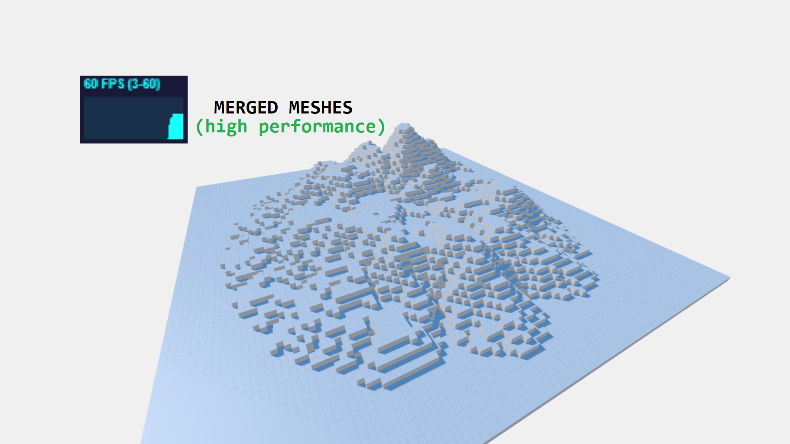
#######################################################################################

**Mansi:**

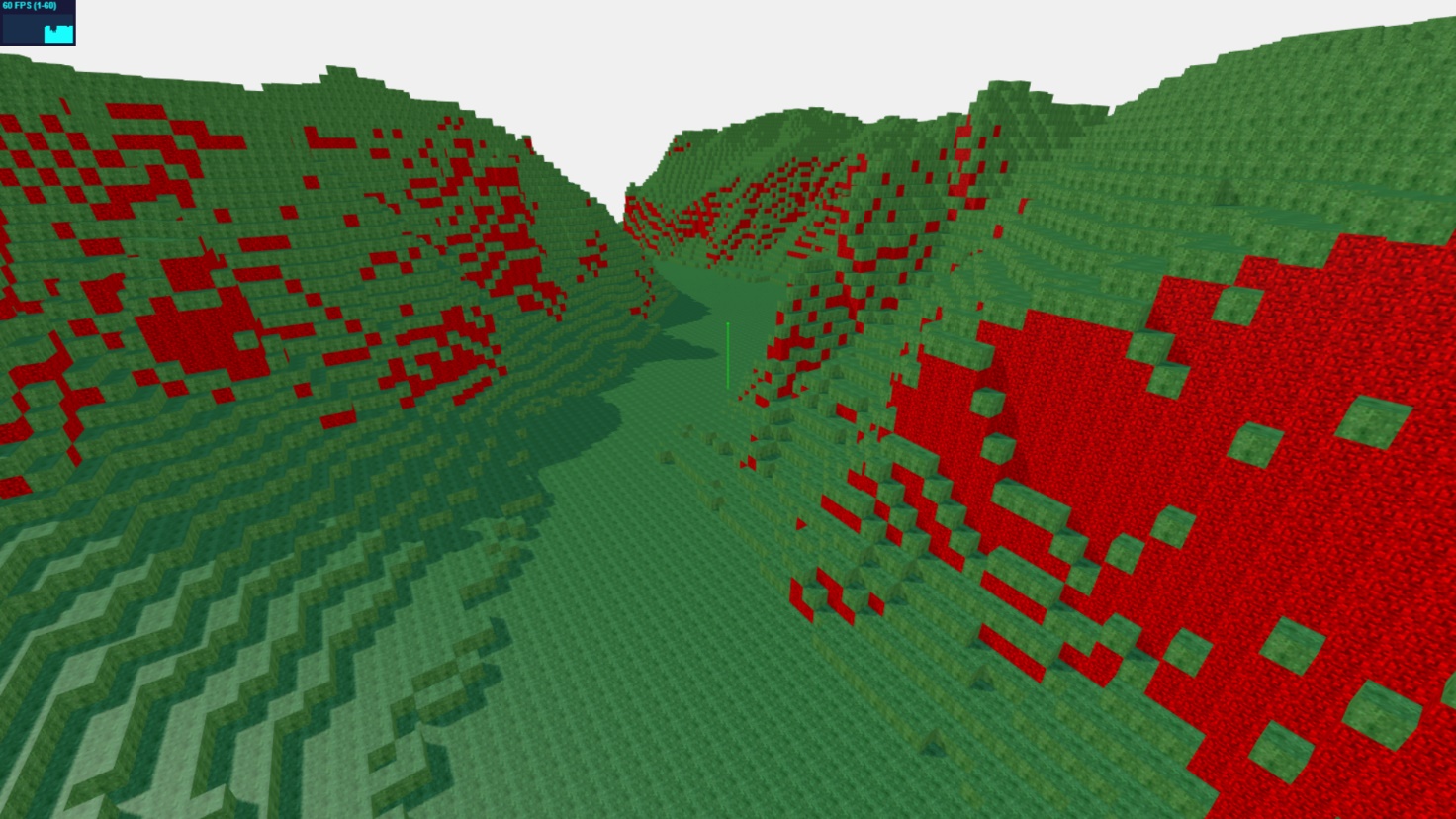
* **Test con heightmap più complessa ->** 100x100=10K cubi = 120K triangoli; framerate inaccettabile (10-15 su Intel UHD 620); bisogna valutare possibili ottimizzazioni.
  + Usare dei cubi non è per niente efficiente, la maggior parte delle facce sono inutili, sono nascoste e potrebbero benissimo non essere renderizzate al fine di aumentare di molto le performance.

**Obiettivi del giorno:** Ottimizzazione della generazione del terreno, lato geometry.

* **Soluzione:** inizialmente con una heightmap di 100x100 istanziavamo 10K blocchi nella scena, portando ad un elevato overhead nella gestione di un alto numero di mesh. La soluzione attuata unisce i mesh di tutti i cubi in un unico mesh. Il numero di triangoli non cambia, ma WebGl si trova a lavorare con un unico mesh (frutto del merge degli altri).
  + - * **Possibili problemi:** Ora abbiamo un unico mesh, non possiamo applicare materiali diversi… Si possono applicare diverse texture a diverse porzioni del mesh? Da sperimentare e verificare se la soluzione utilizzata è valida.
      * **Altre considerazioni:** L’implementazione più corretta nella generazione del terreno non dovrebbe utilizzare dei cubi ma delle facce, l’attuale implementazione.

*Le due immagini riportano le differenze di framerate ottenuto, senza variare il risultato finale:***TEST ESEGUITO SU GPU INTEGRATA INTEL UHD620 (heightmap 100x100):**

* Settato il renderer in high performance.
* Successive modifiche: I cubi in superficie (erba) vengono mergati in un unico mesh. Allo stesso modo, quelli aggiunti successivamente per coprire i “buchi” vengono mergati in unico mesh “terra”.

*Primi test con le texture del terreno:*

* Vari test di performance su UHD620, Nvidia 940MX, RTX 2070.