

Računalna Animacija – 3. Laboratorijska Vježba

Lovro Joja 0036525822

Ak. God. 2023./24.

Vizualizacija Ulaza Zvuka

1. Pokretanje

Program se najjednostavnije može pokrenuti putem konzole s linijom „py MicrophoneVisualize.py“. Za pokretanje programa potrebna je Python verzija 3.x (x = 6 ili više), te biblioteke :

numpy

pyaudio

PyQt5

pyqtgraph

scipy

matplotlib

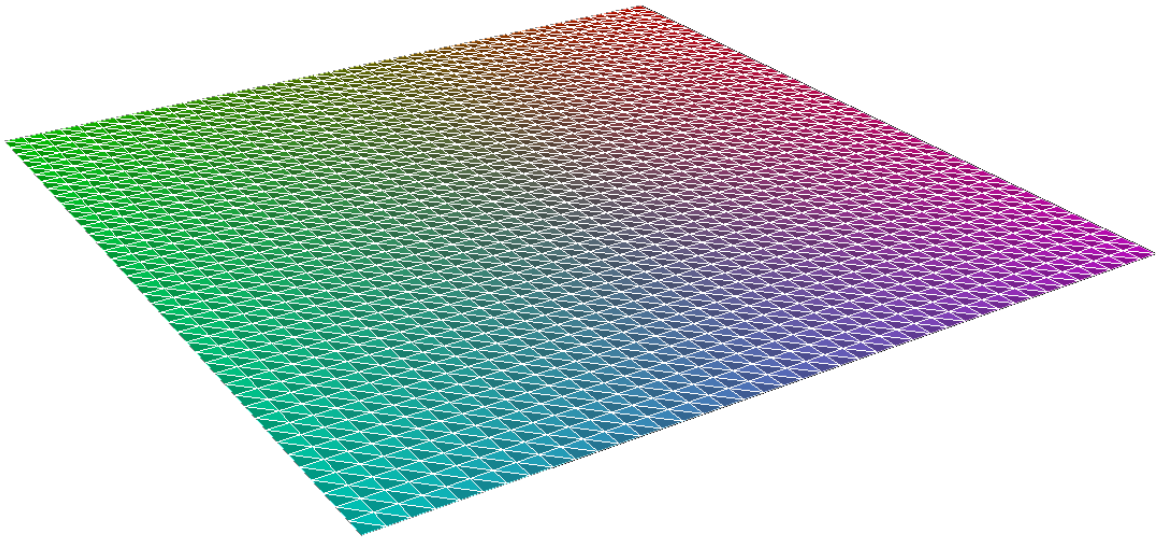
opensimplex

2. Opis rada

2.1 Geometrija

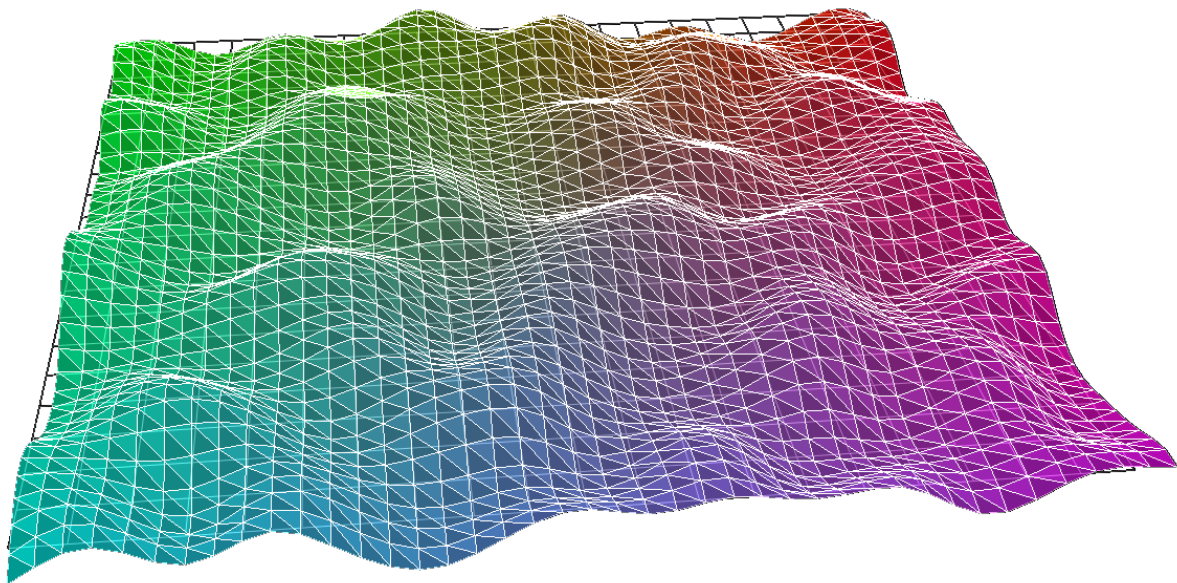
Polazna točka je običan mesh item kojim definiramo 2d ravninu s vrhovima čije koordinate x i y dolaze u intervalu od -20 do 20. Time je definirana mreža kvadrata koja je dodatno

podijeljena na trokute radi finijih rezultata. Svakome je vrhu također trenutno dodana placeholder boja.



Slika 1. Početna 2D mreža

Svaki vrh trenutno ima vrijednost z koordinate 0, kako bi ga transformirali u 3D teren svakom vrhu ćemo z koordinatu izračunati kao šum. Običan Gaussov šum radi, ali su prijelazi među susjednim vrhovima prenasumični i preoštri, pa teren ne izgleda prirodno. Zato je pomoću biblioteke openterrain dodan Perlinov šum i time dobivamo rezultat na slici 2. Kako bi animirali ovo generiranje mesha je strpano u jednu metodu koju funkcija update poziva, pri generiranju su x i y koordinate vrhova uvijek iste, a z koordinata se računa pomoću varijable offset koja se proslijedi generatoru šuma i time dobivamo prirodno kretanje animiranog terena

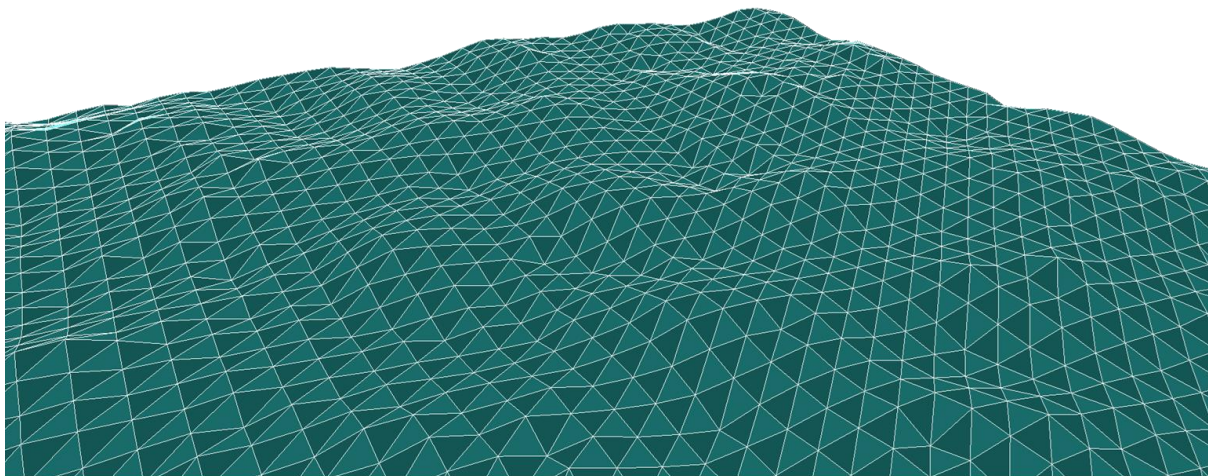


Slika 2. 3D površina dobivena korištenjem Perlinovog šuma

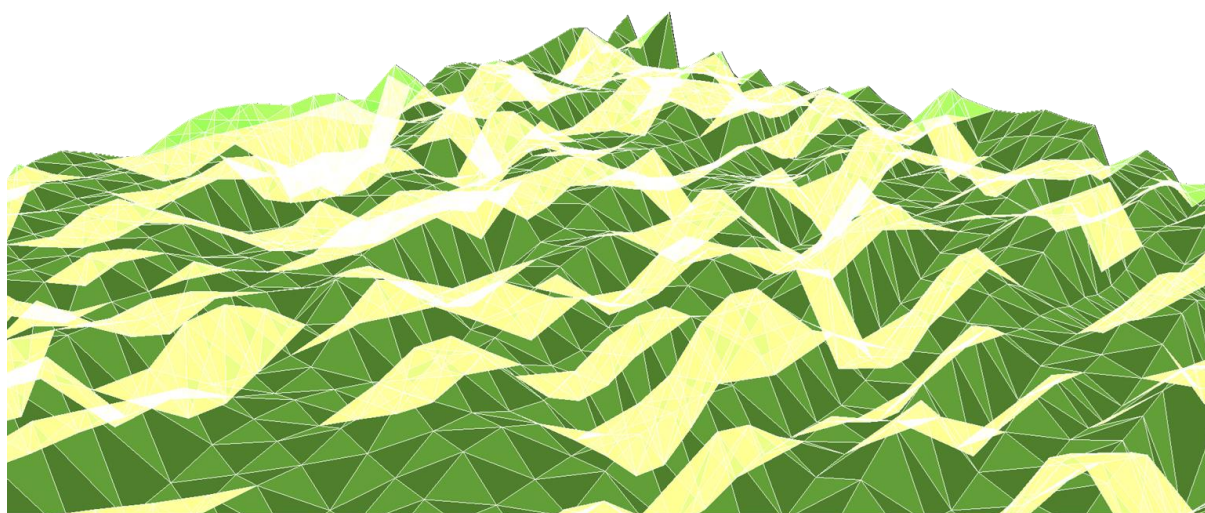
2.2 Zvuk

Sad kada postoji animirani 3D teren, može se njime manipulirati putem audio unosa kroz mikrofona. To omogućuje biblioteka `pyaudio`, pomoću koje je definiran tok u varijabli `stream`, koji u diskretnim vremenskim trenucima prima unos veličine `CHUNK` (broj vrhova u ovom slučaju). Prvo se dobiveni signal, koji ovisi o glasnoći unosa, zapisuje u varijablu `wf_data`, koju pretvaramo u `numpy` matricu te ju koristimo za određivanje koordinata z svakoga vrha i time dobivamo vizualnu reprezentaciju glasnoće kao što se vidi na slikama 3. i 4.

Zapis u `wf_data` se zatim provede kroz 1D Fourierovu transformaciju kako bi izvukli frekvencijsku informaciju o audio unosu. Frekvencija unosa ovdje služi kako bi mijenjali boju 3D površine. Za određivanje boje korišten je colormap *viridis* iz biblioteke `matplotlib`. Vrijednost boje onda ide od 0 do 1 a određuje se tako da se korijen srednje vrijednosti unosa provedenog kroz 1D FFT podijeli s korijenom maksimalne vrijednosti iz iste liste. Tako smo dobili teren koji ne reagira samo na glasnoću audio unosa, nego vizualizira i frekvenciju promjenom boje.



Slika 3. Relativna tišina



Slika 4. Osoba govori i promijenjene su visine vrhova, te se boja promijenila zbog frekvencije(korištena je mapa boja viridis iz matplotliba)