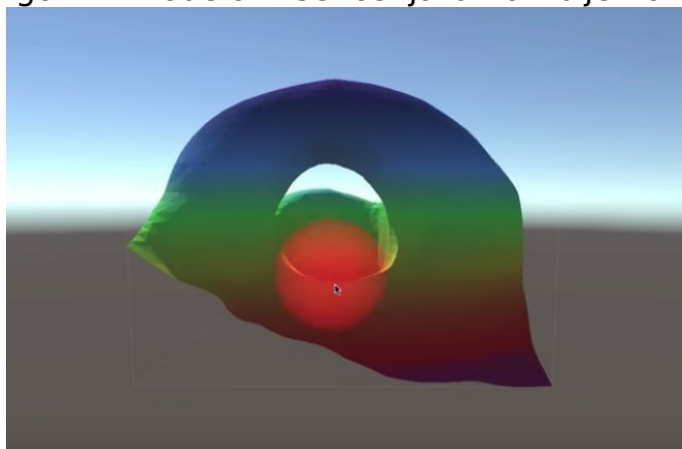


## Drugi domaći zadatak

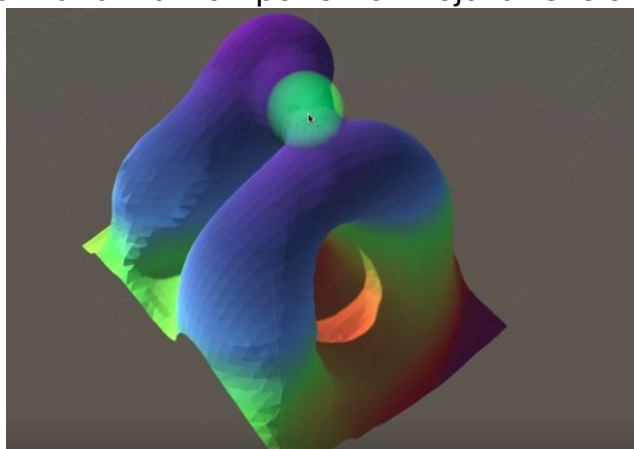
Zadatak se radi upotrebom razvojnog alata Unity3D. Drugi domaći zadatak studenti rade samostalno. Za sve elemente postavke zadatka koji nisu dovoljno precizno definisani postavkom, studenti treba da usvoje razumne pretpostavke i primene ih prilikom rešavanja zadatka.

### Postavka zadatka

Realizovati alat za izradu 3D terena uz pomoć četke. Teren može da ima i konkavnih delova poput pećina, lukova, tunela i litica. Zbog mogućnosti postojanja ovakvih delova terena potrebno je primeniti algoritam marširajućih kocki (eng. *marching cubes algorithm*) [2] koji deli vidljivi deo prostora na pojedinačne voksele [3], kako bi se pronašla odgovarajuća geometrijska mreža koja opisuje površ terena (eng. *isosurface* [4]). Alat treba da ima mogućnost zadavanja nivoa izdeljenosti prostora na voksele (potrebno je razumeti detalje algoritma), kao i poluprečnika četke koja je sfernog oblika. Pritiskom na levo dugme miša treba da se generišu novi delovi terena u prostoru koji zahvata četka, dok pritiskom na desno dugme treba da se uklanjaju delovi zahvaćeni četkom. Teren je moguće uveličavati ili smanjivati (*zoom in/out*), kao i rotirati oko koordinatnog početka pomeranjem pritisnutnog miša (*mouse drag*) ili putem strelica na tastaturi. Alat treba da pruži i mogućnost poravnavanja celokupnog terena na zadatu visinu, kao i mogućnost da se generiše početni teren na slučajan način. Tačke na terenu treba da budu obojene u zavisnosti od visine tako da najniže imaju crvenu, a najviše ljubičastu boju (iskoristiti HSV model boja). Teren osvetliti direkcionalnim izvorom svetla Fongovim modelom senčenja uz ambijentalnu i difuznu komponentu. Boja direkcionalnog



Slika 1



Slika 2

svetla takođe može da se menja iz inspektora. Na slikama je prikazan teren u trenutku kada se bio dodavao, odnosno uklanjao jedan deo četkom.

### Marching cubes algoritam - ukratko

Ideja marching cubes algoritma je da se 3D prostor obuhvati kockom koja se dodatno deli na voksele kroz koje može da prolazi, ali i ne mora, geometrijska mreža terena sastavljena od trouglova. Svaki voksel ima ukupno 8 skalarnih vrednosti u svojim temenima kojima se određuje koja temena vokselu su unutar terena, koja van terena

(recimo u vazduhu ili pećini), a koja na samoj površi terena. Geometrijska mreža terena koji se pravi odgovara funkciji  $F(x, y, z) = f$ , gde su  $x, y$  i  $z$  koordinate temena mreže, a  $f$  je konstanta sa kojom se porede skalarne vrednosti voksel (često je nula). Nakon što se prvim korakom algoritma odredi pripadnost temena, sledeći korak je pronalaženje stranica voksel koje seče površ terena (uz pomoć *edgeTable* tabele [2]). Nakon pronalaženja stranica, narednim korakom se uz pomoć *triTable* tabele [2] i interpolacijom traže tačke preseka površi terena sa stranicama voksel. Tačke preseka se uz pomoć poslednje tabele grupišu u trouglove koji su deo geometrijske mreže terena. Rezultat algoritma je geometrijska mreža terena sačinjena od trouglova koji se efikasno prikazuju rasterizacijom. Prilikom svake promene terena, potrebno je tekuće vrednosti voksel (tačnije 8 vrednosti), čuvati u 3D teksturi. Detaljno objašnjenje algoritma sa suštinskim delom koda koji može da se iskoristi je dato u [2], dok se preporučuje i sekcija 1.2 u [1].

## Literatura

- [1] H. Nguyen, in *GPU Gems 3*, 2007, pp. prva glava, prvo poglavlje.
- [2] P. Bourke, "Polygonising a scalar field," [Online]. Available: <http://paulbourke.net/geometry/polygonise/>. [Accessed 4 10 2022].
- [3] M. ' . Jan, "Pixels and voxels, the long answer," [Online]. Available: <https://medium.com/retronator-magazine/pixels-and-voxels-the-long-answer-5889ecc18190>. [Accessed 5 10 2022].
- [4] "Isosurface," [Online]. Available: <https://svi.nl/IsoSurface>. [Accessed 5 10 2022].