

Drugi domaći zadatak

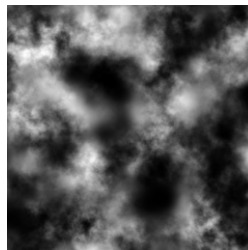
Zadatak se radi upotrebom razvojnog alata Unity. Drugi domaći zadatak studenti rade samostalno. Za sve elemente postavke zadatka koji nisu dovoljno precizno definisani, studenti treba da usvoje razumne pretpostavke i primene ih prilikom rešavanja zadatka.

Postavka zadatka

Realizovati program za interaktivno simuliranje vodene površi. Simulaciju vršiti korišćenjem mapa visina. Mapu visine kreirati u vreme izvršavanja, u svakom frejmu pre samog crtanja vodene površi, korišćenjem *Compute* programa za senčenje. Postupak kreiranja je opisan sledećim pseudokodom:

```
for ( int i = 0; i < height_map_width; ++i ) {  
    for ( int j = 0; j < height_map_height; ++j ) {  
        int sum = current_height_map[i - 1][j]  
                + current_height_map[i + 1][j]  
                + current_height_map[i][j - 1]  
                + current_height_map[i][j + 1];  
  
        tmp[i][j] = sum / 4 - current_height_map[i][j]  
  
        tmp[i][j] *= damp_factor;  
    }  
}  
  
for ( int i = 0; i < height_map_width; ++i ) {  
    for ( int j = 0; j < height_map_height; ++j ) {  
        next_height_map[i][j] = current_height_map[i][j] + tmp[i][j];  
    }  
}
```

Početnu mapu visina inicijalizovati proizvoljnim vrednostima. Vredost **damp_factor**, kao i dimenzije mape visina odrediti empirijski. Sledeća slika predstavlja ilustraciju na kojoj je prikazana mapa visina. Bela boja predstavlja najveću, a crna najmanju moguću elevaciju.



Pored mape visina potrebno je kreirati i mapu normala na isti način. Mapa normala je istih dimenzija kao i mapa visina. Kreiranje mape normala je ilustrovano sledećim pseudokodom:

```

for ( int i = 0; i < height_map_width; ++i ) {
    for ( int j = 0; j < height_map_height; ++j ) {
        height_left  = next_height_map[i - 1][j] * MAXIMUM_WATER_HEIGHT;
        height_right = next_height_map[i + 1][j] * MAXIMUM_WATER_HEIGHT;
        height_up    = next_height_map[i][j - 1] * MAXIMUM_WATER_HEIGHT;
        height_down  = next_height_map[i][j + 1] * MAXIMUM_WATER_HEIGHT;

        dx = height_right - height_left;
        dy = height_down - height_up;

        normal_map[i][j] = normalize (
            float3 ( -dx * strength, 1, -dy * strength )
        );
    }
}

```

Parametar **strength** odrediti empirijski. Prikazivanje vodene površi vršiti mrežom trouglova odgovarajućih dimenzija. Visine temena postaviti na odgovarajuće vrednosti čitanjem iz odgovarajuće teksture u programu za senčenje temena. Omogućiti postavljanje boje vodene površi. U vodi se ogleda ambijent (*cube map*) kao i spekularna komponenta usmerenog izvora svetla. Obezbediti da se prozirnost vode smanjuje sa porastom ugla koji čine normala površi i vektor pogleda. Ne treba simulirati kaustiku, refrakciju ni druge vizuelne pojave. Korisnik može da interaguje sa vodenom površi i izazove njeno talasanje tako što će kursorom "šarati" po površi, kao da provlači prst kroz vodu. Intenzitet talasa koji ovako nastaju kontroliše se iz Unity Editora. Sledeća slika ilustruje rešenje ovog zadatka.

