Vizualizace a simulace vodní hladiny

Patrik Chukir <xchuki00@stud.fit.vutbr.cz>

31. prosince 2017

1 Úvod

Projekt se zabývá *real-time* implementací simulace vodní hladiny, jak tvorbou a chováním vln, tak zrcadlením okolí a zvětšení věcí pod hladinou. Takovéto simulace můžou být vhodné pro zobrazení menších(rozuměno rybníků, bazénů) vodních ploch například v počítačových hrách.

2 Teorie

Teorie v tomto projektu se dá rozdělit na dvě části - Vizuální a Fyzikální chování hladiny. U vizuálních jevů v rámci tohoto projektu nebyla užita žádná zajímavá teorie, proto je v této kapitole přeskočím a budu se zabývat pouze teorií výpočtu vln.

Výpočet vln využívá algoritmus *Height field water*[MF14]. Vychází se ze čtvercové sítě bodů reprezentující hladinu, ale pracuje se pouze s výškou a kinetickou energii bodu dle vzorce:

$$f = \frac{c^2}{h^2} * (u[i-1] + u[i+1] + u[i-width] + u[i+width] - 4 * u[i])$$

$$u_{new} = u[i] + velocity[i] * f * \delta t$$

- i - index aktuálního bodu
- uau_{new} - výchozí pole výšek bodů
- $width, c, h$ - počet bodů v řádku, rychlost šíření,velikost čtverce sítě
- $velocity$ - kinetická energie sloupce/bodu

Ze vzorce vyplývá, že nová poloha bodu je dána průměrným rozdílem výšek a energii z předchozích iterací. Tato energie nám představuje v podstatě setrvačnost kapaliny.

3 Popis řešení

Jako výchozí nástroje jsou použity knihovny GPUEnginy, SDL2 a glm. Jako kostra projektu je použita adresářová struktura cvičení z předmětu PGPa a příslušné CMakeListy. Pro zviditelnění vizuálních efektů byla použita textura z fotobanky, více ./imgs/credist.txt.

3.1 Instalace

Instalace:

1. Spustit CMake v kořenovém adresáři aplikace.
2. Nakopírovat do složky ./build SDL2.dll a SDL2d.dll, pro realese a debub překlad
3. Přeložit vygenerovaný projekt.

3.2 Ovládání

- W - pohyb vpřed
- S - pohyb vzad
- A - pohyb doleva
- D - pohyb doprava
- Levé tlačítko myši+držení - rotace kamery
- 1-6 - vyvolání vln na přednastaveném místě

3.3 Implementace vizuálních efektů

Vizuální stránka je realizována pomocí postupné ho vykreslování, *alfa blendingu a stencil bufferu*. Kdy se do pouze do *stencil bufferu*[Ove12] vykreslí vodní hladina s hodnout 2, a přivrácené stěny vodního tělesa s hodnout 1. Potom efekt zrcadlení je oříznut na 2 a zvětšení na >0, tedy je vidět i skrze boční stěny. Samotné efekty jsou pouhé další vykreslení objektů.

- zrcadlení = převrácení a posun od dvojnásobek vzdálenosti k vodní hladině po kolmici
- zvětšení = zřejmé

Ovšem, aby se v scéně nezobrazovaly které se nemají zrcadlit a tak, tak je ještě použito oříznutí pomocí clip plane nastavené do úrovně vodní hladiny.

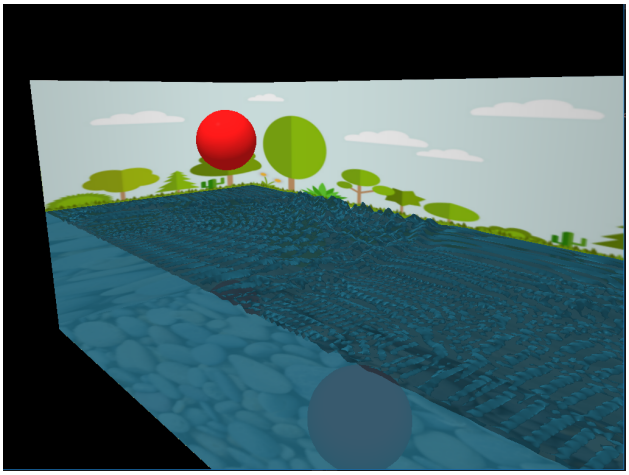
3.4 Implementace fyzikálních efektů

Jde o výpočet dle výše zmíněného vzorce, realizuje jej *compute shader* přímo nad daty, které jsou potom předávány *vertex shaderu*. Ovšem z důvodů odrazu světla je potřeba spolu z výškou měnit i normály bodů, což se provádí při stejném průchodu jako výpočet výšky. V případě normál je vektorový součin vektorů z počítaného bodu do jeho 4 sousedních bod. Jako výchozí materiál pro práci s *compute shadery* posloužila prezentace "How to Use and Teacj OpenGL Compute Shaders"[Bai14]

4 Vyhodnocení

Aplikace běží opravdu rychle i pro velký počet bodů sítě vodní hladiny, jen setrvačnost systému není úplně vyladěná(pokud se jednou začne hýbat nedojde již zpět do klidového stavu). Co se týče vizuálních efektů bylo dosaženo uspokojivého výsledku na rychlost zpracování. Jedině stěny nádrže se v rámci efektu deformace/zvětšení nechovají 100% věrohodně.

Obrázek 1: Zrcadlení na klidné hladině



Obrázek 2: První vlny

Reference

[Bai14] Mike Bailey. How to use and teach opengl compute shaders, 2014. [Online; accessed 27-December-2017].

[MF14] Matthias Müller-Fischer. Fast water simulation for games using height fields, 2014. [Online; accessed 27-December-2017].

[Ove12] Alexander Overvoorde. Opengl - depth and stencils, 2012. [Online; accessed 27-December-2017].