

Óbudai Egyetem

Neumann János Informatikai Kar

Szoftvertervezés és –fejlesztés Intézet



TUDOMÁNYOS DIÁKKÖRI DOLGOZAT

Folyóvizek felszínformáló erejének modellezése

Szerző: Klope Bálint Dániel
mérnökinformatikus Bsc. szak, IV évf.

Konzules: Prof. Dr. Szénási Sándor
egyetemi tanár

Budapest, 2022.

Tartalomjegyzék

1.	TDK dolgozat rövid bemutatása	4
2.	Földrajzi háttér	5
2.1.	Geomorfológia.....	5
2.2.	Külső-belső felszínformáló erők	5
2.3.	A folyóvíz felszínalakító munkája	6
3.	Digitális tartalom az oktatás és kutatás területén	7
3.1.	IKT eszközök az oktatás és kutatás területén	7
3.2.	Hagyományos szemléltetés kiváltása a felsőoktatásban	8
4.	Korábbi megoldások a vízerózió szimulálására	9
4.1.	Fraktális modellek	9
4.2.	Diffúziós modellek	9
4.3.	Valósághű eredmények	10
4.4.	GPGPU megvalósítások	10
4.5.	Interaktív megoldások	11
4.5.1.	Bevezető	11
4.5.2.	Első interaktív megoldás	11
4.5.3.	Összetettebb megoldás	12
4.5.4.	Szemléletes megoldás, CPU és GPU összehasonlítás	13
4.5.5.	Legújabb megvalósítás	13
5.	Megvalósítandó program terve	14
5.1.	Feladat specifikációja	14
5.2.	Eszközök kiválasztása	15
5.3.	Eljárás	16
5.3.1.	Vízszint növelése	18
5.3.2.	Fluxus és sebesség számítása	18
5.3.3.	Vízszint aktualizálása	18
5.3.4.	Erózió-depozíció számítása	19
5.3.5.	Párolgás	19
5.3.6.	Talaj elaprózódás.....	19
5.3.7.	Hordalék szállítása	20
5.3.8.	Termálerózió.....	20
6.	Megvalósítás.....	21
6.1.	Adatok	21
6.2.	Implementáció	22
6.2.1.	Áttekintés	22

6.2.2.	Beállítások	23
6.2.3.	Szimuláció CPU	24
6.2.4.	Szimuláció GPU - Compute Shader.....	25
6.2.5.	Megjelenítés	26
7.	Megvalósult program bemutatása	27
8.	Értékelés	30
8.1.	Tesztek futtatása	30
8.2.	Eredmények értékelése.....	30
9.	Továbbfejlesztési lehetőségek.....	31
9.1.	Eszközök, effektek	31
9.2.	Szimuláció bővítése	31
9.3.	Lehetséges felhasználás	31
Hivatkozások.....		32
Ábrajegyzék		34

1. TDK DOLGOZAT RÖVID BEMUTATÁSA

Régóta foglalkoztat a felszín és a felszín változásainak szimulálása, esztétikus és valóságghű ábrázolása. Ezért választottam a víz felszínformáló erejének modellezését dolgozatom témájaként.

A víz viselkedésének, mozgásának, munkájának szimulálása népszerű téma az informatika területén. A legvalóságghűbb megoldást a részecskealapú rendszerek jelentik, ugyanakkor ezeknek van a legmagasabb számításigénye is. Ez olyan nagy teljesítményű, költséges hardware-t igényel, ami a mindennapi felhasználók számára nem érhető el. Sok megoldás született a víz eróziós munkájának bemutatására, viszont ezek elérése nehézkes, valamint a felhasználó csak korlátozottan tud beavatkozni a program működésébe.

Munkám keretében egy, a folyóvíz eróziós munkáját bemutató interaktív, valós idejű szimulációt készítettem el. Az eszköz felhasználására sor kerülhet földrajzi kutatások során, az oktatásban (mind tanárok, mind diákok részéről), de szórakoztató-ismeretterjesztő programként akár a szabadidő hasznos eltöltését is szolgálhatja. Integrálható játékprogramba vagy más tervező eszközökbe. Esetleg hasznosíthatják különböző, a földrajzi ismeretekre is építő társtudományok művelői is (pl. helymeghatározás régészek számára).

2. FÖLDRAJZI HÁTTÉR

2.1. Geomorfológia

A „földfelszín formakincsének elemzésével, kialakulásával, rendszerezésével és térképezésével” foglalkozik a geotudományok egyik klasszikus ága, a geomorfológia. [1] A Föld felszínének vizsgálata régre nyúlik vissza, igazi fejlődésnek azonban a 19. század végén indult, ekkor került kidolgozásra számos, ma is érvényes tudományos nézet alapja. A 20. század második felében a felszínformák vizsgálatában erősödött a mérnöki szemlélet: kvantitatív kutatási módszerek, fizika törvényei szerinti leírások kerültek bevezetésre. Napjainkban a térinformatika nyújtotta lehetőségek (3D-s modellezés, szimuláció) segítik a tudományág fejlődését. [2]

2.2. Külső-belső felszínformáló erők

A földfelszín állandóan változik, alakul, fejlődik. A változások okozója két erő: külső és belső erő együttes, de egymással ellentétes munkája. A belső erők erőteljesebben hatnak viszonylag rövidebb idő alatt, a külső erők kevésbé erőteljesek, így viszonylag hosszabb idő alatt érik el hatásukat („kis erő – nagy idő” elv). [3]

A belső (endogén) erők a Föld belső energiájából táplálkoznak, munkájuk eredménye a kőzetlemezek mozgása, vulkánosság, hegységek kiemelkedése, medencék és árkok besüllyedése. A külső (exogén) erők a Nap energiáját hasznosítják, és kívülről formálják a felszínt. Munkájuk három részfolyamata a lepusztítás, a szállítás és a felhalmozás. A külső erők közé tartoznak az időjárási elemek (napsütés, hőmérséklet-változás, csapadék, szél), felszíni vizek (óceánok, tengerek, folyók), jég (földi jégtakaró, gleccserek), tömegmozgások, élővilág. [3]

Minden tényező modellezése virtuálisan lehetetlen, ezért én a felszínformáló külső erők közül a folyóvíz építő-romboló munkáját mutatom be. A folyóvízi erózió a mérsékelt övben a legfontosabb felszínformáló erő.

2.3. A folyóvíz felszínalakító munkája

Folyóvíz, vízfolyás vagy lefolyó víz alatt a meghatározott pályán, vonal mentén (lineárisan), tulajdonképpen mederben lefolyó vizet értjük. [4] A magasabban fekvő területekről az alacsonyabban lévők irányába halad, vízmennyisége, vízjárása általában ingadozik. A felszíni vízfolyások vize részben a napvilágra lépő forrásokból, részben pedig közvetlenül a csapadékból származik. [5]

A külső erőkre jellemző hármas törvény, a pusztítás-szállítás-felhalmozás a folyókra is érvényes. A felszín formálásában az általuk szállított, különböző szemcse nagyságú hordalék: iszap, homok, kavics játszik szerepet. A hordalékszállítás módja a szemcsemérettől függően lehet görgetés, ugráltatás (szaltáció), lebegtetés (flotáció). A folyók munkavégző képessége függ a vízhozamtól, az áramlási sebességtől, melyet a meder esése határoz meg. [3] A folyók felszínalakító munkája a vízhozammal egyenesen, a sebességgel pedig négyzetesen arányos, vagyis a nagy esésű, kisebb hozamú folyó eróziós munkája jelentősebb, mint a kisebb esésű, de nagyobb hozamú folyóké. [6]

A folyók felszínformáló munkája a felszín esése szerint csoportosítható:

A felső szakasz jellegű folyó a nagy lejtésű területen formálja a felszínt. Sebessége nagy, munkavégző képessége nagyobb, mint amekkora a hordalék elszállításához szükséges. Hordalékával pusztítja, mélyíti jellegzetesen V alakú völgyét (bevágódó szakaszjelleg).

A középszakasz jellegű folyó a mérsékelt lejtésű területeken folyik, munkavégző képessége akkora, mint amekkora a hordalék elszállításához szükséges. A lerakott hordalékból létrejövő zátony eltéríti a folyó medervonalát, így a folyó kanyargásaival (meanderezés) szélesíti völgyét (oldalazó szakaszjelleg). Kanyarulatainak homorú partját pusztítja, a domborút építi.

Az alsó szakasz jellegű folyó a kis lejtésű területekre jellemző. Munkavégző képessége kisebb, mint amennyi a hordalék elszállításához szükséges. Hordalékát lerakja, zátonyokat, szigeteket épít belőle a medrében, melyeknek köszönhetően szétágazik, majd elhagyja medrét. [7]

Tulajdonságok	Szakaszjelleg		
	Felső	Középső	Alsó
domborzati viszony	hegyvidék	dombvidék	síkvidék
esés	nagy	közepes	kicsi
sebesség	nagy	közepes	kicsi
hordalékszállítás	jelentős	közepes	kicsi
munkavégzés	pusztít	pusztít és épít	épít
mederforma	V alakú	kiszélesedő	tál alakú

1. táblázat A szakaszjellegek jellemzői táblázatos formában [8]

A folyók szakaszjellege forrásuktól a torkolatukig a domborzattól függően többször változik [9].

3. DIGITÁLIS TARTALOM AZ OKTATÁS ÉS KUTATÁS TERÜLETÉN

3.1. IKT eszközök az oktatás és kutatás területén

Napjainkra a digitális eszközök használata az élet minden területén természetessé vált, az ún. IKT (információs és kommunikációs technológiák) alkalmazása a közoktatásban mégis erősen korlátozott az Európai Unió országaiban. Az Európai Bizottság felmérése szerint az eszközpark kiépítésén túl nagyban segítené az új technológiák intenzívebb használatát, ha a megfelelő minőségű digitális tananyagok száma növekedne. [10]

Az IKT eszközök kiválóan alkalmasak szemléltetésre, információszerzési, -feldolgozási és -alkalmazási képesség elsajátítására [11] interaktív használatukból fakadóan számos kompetencia fejlesztésére, valamint segítségükkel a hagyományos (időhöz és helyhez kötött) oktatás is rugalmasabbá tehető. [12] Ez utóbbi tulajdonság jelentőségére a 2020-ban kitört koronavírus járvány is felhívta a figyelmet: egyik napról a másikra kellett átállni a frontális oktatásról a távoktatásra, mely óriási kihívások elé állította a tanárokat. [13] Mindez jóval gördülékenyebben mehetett volna végbe, amennyiben a megfelelő IKT eszközök használata a közoktatásban a mindennapi gyakorlat része lett volna.

A természettudományos tárgyak, így a földrajz is, egyre kevésbé népszerűek a diákok körében. Az IKT eszközök használata az aktív tanulói részvételnek, interakciónak köszönhetően örömszerzésre is lehetőséget nyújt, ezzel segíthet felkelteni az érdeklődést a tantárgy iránt. [14]

Az oktatásban használható digitális tananyagok, eszközök fejlesztésére tehát nagy szükség van, számos projekt keretében érdemes lenne ezt figyelembe venni, és úgy alakítani, hogy a tanítás-tanulás során is felhasználható legyen a szoftver.

3.2. Hagyományos szemléltetés kiváltása a felsőoktatásban

A felsőoktatásban több évtizede használatos fluviális geomorfológiai folyamatok modellezésére a terepasztal. Előnye, hogy látványos, térben kicsinyítve és időben felgyorsítva vizsgálható a folyóvízi felszínformálás. Hátránya azonban, hogy a demonstráción túl komolyabb kutatási célokra nem alkalmas, illetve a felszínfejlődés sem modellezhető teljeskörűen. Ennek egyik oka, hogy a homokon kívül más alapkőzet nem biztosított a kísérletekhez. Másrészt a kutatásokhoz megfelelő méretű, a klimatikus tényezőket, vízjárást, vízhozamot, illetve a belső erők hatását szimuláló, bonyolult, pontosan szabályozható és mérhető vízellátó, illetve emelő-süllyesztő szerkezetre van szükség (ilyen terepasztallal csak a Debreceni Egyetem Földtudományi Intézetének folyóvízes laboratóriuma rendelkezik). [15] Jóval egyszerűbb, költséghatékonyabb lenne megfelelő számítógépes program használata, mely távoktatásban, de akár terepen, gyakorlati oktatás keretében is kiválóan felhasználható lenne nemcsak szemléltetésre, de kutatási célokra is.

Az ennek segítségével elvégzett kísérletek egészen pontosan paraméterezhetők, tetszőleges pontossággal be lehet állítani a környezeti és egyéb szimulációs paramétereket. Ugyanez igaz a végeredményre, az egyes kísérletek közbülső illetve végállapotát nagy pontossággal tudjuk megmérni. Az így végzett kísérletek bármikor reprodukálhatók, visszajátszhatók akár az idő mesterséges gyorsításával-lassításával is.

4. KORÁBBI MEGOLDÁSOK A VÍZERÓZIÓ SZIMULÁLÁSÁRA

4.1. Fraktális modellek

Számos algoritmus született az erodált táj, illetve folyók tanulmányozására. Az 1990-es évek elején jelentek meg az első fraktális szabályokon alapuló, ad hoc változtatásokkal létrejött megvalósítások. Kezdetben külön folyamatban hozták létre a felszíninformát, illetve a folyót. Egyes megoldások már előre meghatározott (megépített) folyórendszer köré generáltak tájat, más alternatív megoldások fraktális hegyeken alkottak folyókat víz erózió szimulálásával.

Az 1993-ban Prusinkiewicz és munkatársa [16] által fejlesztett projektben már egy közös folyamatban építették fel a különböző felszíninformákat és a folyót. Ezek a programok csak a folyók felszíninformáló munkájának eredményét jelenítették meg fényképszerűen, magát a folyamatot nem.

4.2. Diffúziós modellek

Ezt követően már több algoritmus a fizika törvényein alapuló modellel próbálja szimulálni az eróziós folyamatot, és módosítani egy már létező, mesterségesen létrehozott felszínt.

Nagashima és munkatársai [17] megvalósítása realisztikus völgyeket eredményezett előre definiált folyók eróziós munkáját szimulálva, amiknek létrehozása a felszíntől függetlenül történt, viszont ez csak gyors folyású (nagy esésű) folyók esetére működött.

Benes és Forsbach [18] munkája is hasonló elven alapult. Előrelépést jelentett, hogy a folyamatot külön lépésekre bontotta, így lehetőség nyílt ezeknek a megismétlésére, így szimulálva például különböző évszakokat. Új eljárást alkalmaztak, megjelent a párolgás, mint folyamat. Az elpárolgott víz által szállított hordalék lerakódott.

A diffúziós modell alapján készült megoldások nem megfelelően modellezték a víz valós mozgását, a törmelék felvételét-szállítását-lerakását. A víz mozgása és törmelékszállítása szorosan kapcsolódik a folyó áramlási sebességéhez. Ebből következően számos kutató víz szimulációs eljárásokat integrált az eróziós modellbe.

4.3. Valóság-hű eredmények

Az első megvalósításban Chiba és munkatársai [19] hegyek és völgyek erózió általi kialakulását szimulálták részecske alapú rendszerrel. A vízrészecskék sebességét dinamikusan frissítették a gravitációtól és a lokális dőlésszögektől függően. Ezután a felszín módosult a vízrészecskékkel való ütközéskor az átadott energia alapján. Ezt a munkát Benes és munkatársai [20] fejlesztése követte, tradicionális Navier-Stokes Egyenleteket (NSE) alkalmazott a modelljéhez 3D-s rácsszerkezetben. A fő hátránya ezeknek az algoritmusoknak, hogy a számítási igényük magas, főleg a folyadék szimulálása.

Neidhold és munkatársai [21] előálltak egy egyszerűsített newtoni fizikán alapuló modellel a sebesség számításához 2D-s euklideszi térben. A sebesség minden cellában változik a gravitációnak és a lokális dőlésszögnek megfelelően. A víz másik cellába való mozgásakor a két sebesség vektort egy empirikus formula alapján összegzi. Ez 256x256 cellán 4 fps-sel futott, ez túl kevés egy interaktív megvalósításhoz. A sebességszámítás és az eróziós folyamatok algoritmusai között adatfeldolgozási függőség áll fenn, ezért nem lehet jól párhuzamosítani, nem optimalizálható GPU-n.

4.4. GPGPU megvalósítások

A hardware gyors ütemű fejlődésével egyre többen fordultak a GPU felé általános számítási problémák (GPGPU – General Purpose Computing on Graphics Processing Unit) megoldására, főleg a folyadék szimuláció terén. 2D-ben már sikerült hatékonyan NSE-t megoldani GPU-n [22, 23], viszont a framework kibővítése, hogy 3D-ben NSE-t oldjon meg, mindig limitált egy kis térre. Li és munkatársai [24] a rács-Boltzmann módszert implementálták grafikai hardware-en, hogy folyadékkal kapcsolatos jelenségeket szimuláljanak, de a teljesítmény itt is korlátozott volt a 3D textúrák használata miatt. A felszíni vízmozgás leírására optimális megoldás 2D-ben a 'Shallow Water Equations' (SWE). Az eljáráshoz több iteráció szükséges a 2D rács térben minden lépésnél, ami nem hatékony a nagyméretű területek interaktív megvalósítására. Benes [25] 2007-es munkájában alkalmazta ezt az eljárást valós-idejű erózió szimulációhoz, ami 5 fps-sel futott 300x300 rácson.

1995-ben O'Brien és Hodgins [26] bemutatott egy új eljárást hydrostatic pipe-model néven, amibe integrálva a SWE-t lehetőség nyílt egy interaktív szimuláció megvalósítására.

4.5. Interaktív megoldások

4.5.1. Bevezető

A mai videokártyák magas párhuzamos számítási sebessége kiaknázzható a megfelelő algoritmus alkalmazásával. A GPU megvalósítások esetében a videokártya véges (kisméretű) memóriája szabja meg a határt.

A CPU megoldások esetében nagyobb komplexitás elérésére van lehetőségünk, viszont a magas számítási igény miatt ezek a megvalósítások időigényesek, elveszik az interaktivitás lehetősége. Ezeknél a megoldásoknál inkább a folyamat eredménye, és nem a folyamat megfigyelése a cél. Ugyanakkor pont emiatt a tudományosabb megközelítések CPU-n működnek, mivel a komplex mérések és pontos eredmények mellett kevésbé jelentős a ráfordított idő.

4.5.2. Első interaktív megoldás

Az első GPU megvalósítás interaktív frame rate mellett Mei és munkatársai [27] munkája volt. Alapja az O'Brien és Hodgins [26] munkájában ismertetett hydrostatic pipe-model, egy rektilineáris oszlopokból álló rácsszerkezet, ahol egy-egy oszlop azonos méretű területet jelent, és tartalmazza a víz és a talaj állandó fizikai jellemzőit; talajszint (b), vízszint (d), szállított anyag (s), kiáramló fluviális fluxus (f^L , f^R , f^T , f^B) és sebesség vektort (u,v). Ezeket a tulajdonságokat a GPU 2D height field-ek formájában dolgozza fel.

A szimuláció öt lépésben zajlik:

1. vízszint növelése (ez lehet forrás vagy eső)
2. folyam szimuláció SWE alkalmazásával, majd a víz sebességének és szintjének a frissítése
3. hordalékanyag felvétele vagy lerakása a sebesség függvényében
4. anyag szállítása
5. vízszint csökkentése párolgás következtében

A sorrend fontos, mivel minden következő lépés az azt megelőző eredményeivel dolgozik.

A korábbi megvalósításokkal ellentétben a szimulációjuk nagyobb mezőkön is képes volt interaktív frame rate mellett realisztikus folyamatot bemutatni, azonban közel lapos terepen (lassú vízsodrással) az eljárásuk nem a valóságnak megfelelő jelenségeket produkált. A szimuláció korlátait itt egyértelműen a GPU memóriájának a mérete jelentette. A megoldás csak a víz eróziós hatását szimulálta, és nem adott lehetőséget a felhasználónak a talaj, a víz vagy más egyéb fizikai változók módosítására.

4.5.3. Összetettebb megoldás

Egy ezt követő megoldás Stava és munkatársai [28] munkája, ami számos előrelépést mutatott. A szimulációjukban megjelentek talajrétegek, így növelve a komplexitást, valamint realiztikusabb folyamatot eredményezve. A rétegek vertikálisan feküdtek egymáson meghatározott sorrendben. A legalsó réteg volt a legsűrűbb, a víz a hordalékát mindig a felső, laza réteghez adta hozzá. Az így megvastagodott rétegekből átsorolás történt az alattuk lévő sűrűbb rétegek felé. A legfelső, laza réteg a víz és saját maga nyomása hatására az alatta lévő sűrűbb rétegbe „keményedett”, majd ezt követően a következő réteg is, egészen addig, amíg el nem érte a legalsó réteget. A talajra hatással volt a gravitáció is, az anyagtól függően egy bizonyos fokig a víz hatása nélkül is képes volt elmozdulni. A memória igényre megoldásként a terepet felbontották további kisebb egységekre, így nagyobb területek feldolgozására is lehetőség nyílt a megfelelő kisebb részegységek kiválasztásával és szimulálásával. A felszínalakítás modellezésére három különböző számítást alkalmaztak a szakaszjellegnek megfelelően: gyors mozgású, lassú, illetve (szinte) álló vizekre. Ez jelentette a megoldást az előző projektben felmerülő, a víz sík talajon való hibás viselkedésének a problémájára.

Habár egy hasonló rendszeren futtatva a folyamat valamelyest lassabban teljesített Mei és munkatársai [27] előzőleg ismertetett megoldásánál, egy sokkal komplexebb és pontosabb szimulációt eredményezett interaktív frame rate mellett. Lehetőség volt különböző eszközökkel globális, illetve lokális változtatásokat végezni elfogadható futásidő alatt, mint például források hozzáadása, eső, a víz felszárítása vagy párologtatás, illetve könnyen erodáló réteg (a legfelső réteg) hozzáadása.

4.5.4. Szemléletes megoldás, CPU és GPU összehasonlítás

Jákó [29] az előző két megoldás nyomán Mei és munkatársai [27] munkájának egy továbbfejlesztett változatát valósította meg. Az eljárást modern architektúrákra, mind CPU-n, mind GPU-n alkalmazta. A megfigyelés az volt, hogy GPU-n az eljárás legalább harmincszor, de akár százszor gyorsabban működött.

4.5.5. Legújabb megvalósítás

Az egyik legújabb megvalósítást McLeod és munkatársai [30] ugyancsak Mei és munkatársai [27] munkája alapján készítették. Olyan felhasználói eszközöket implementáltak, ami lehetőséget ad források elhelyezésére, és a táj tetszőleges alakítására. Mindezen túl realisztikus víz vizualizáció, szimulált hullámok, irányított fények és talaj textúra gazdagítja megoldásukat. Stava és munkatársai [28] ötlete alapján ők rétegek helyett Perlin-zaj (heterogén anyag, kisebb-nagyobb kavicsok, kövek) és a magasság alapján (a hegyek közetei ellenállóbbak) határoztak meg egy oldódási együtthatót.

5. MEGVALÓSÍTANDÓ PROGRAM TERVE

5.1. Feladat specifikációja

Cél a már létező megoldásokkal szemben egy olyan program létrehozása, ami megnövelt komplexitással rendelkezik, könnyen bővíthető, szemléletesebb, illetve több eszközt megvalósít a felhasználó számára, egyszerűen alkalmazható (kutatási célokra, de akár a köz-, felső-, és távoktatásban is) mindezt interaktív framerate mellett úgy, hogy a szimuláció továbbra is megfeleljen a valóságnak.

Az általam fejlesztett programban a szimuláció szabadon bejárható, valós időben történik. A rendelkezésre álló, betölthető térképeken túl saját térképek is hozzáadhatók. Indítás előtt a térkép különböző kezdeti paraméterekkel manuálisan módosítható.

A felhasználó már jóval komplexebb beállítási paraméterekkel találkozhat. A szimuláció alatt több módosításra is lehetősége lesz a következők szerint.

Talaj módosítása: a talaj ellenállása változtatható, így szimulálva különböző keménységű kőzeteket. Talaj elvételével vagy hozzáadásával hegyeket emelhetünk vagy völgyeket hozhatunk létre.

Eső és párolgás szimulálása, ami a térkép teljes területére kiterjed. Ezen túl lokálisan is hozzáadhatunk vagy elvehetünk vizet „vödör eszköz” segítségével. Elhelyezhetünk forrásokat és víznyelőket, amiknek hozama és mérete szabadon kiválasztható.

5.2. Eszközök kiválasztása

Az általam választott programozási nyelv a C#. A választás szinte egyértelmű, hiszen ez a nyelv tudja a legnagyobb teljesítményt nyújtani amellet, hogy számos egyéb szolgáltatást is egyszerűen lehet a segítségével implementálni (megjelenítés, stb.). A szimuláció számításigényes részei GPU kernelek formájában lettek implementálva, így itt sem okoz problémát a C# esetleges teljesítmény hátránya az alacsonyabb szintű nyelvekhez képest.

A program kódnyelve a fentiekből adódóan elsősorban C#. Maga a szimuláció egy Compute Shaderen, speciális GPGPU felhasználásra optimalizált HLSL shaderen (High-Level Shader Language) fut, ami egy C stílusú nyelv. A kernelek indítása és az adatok másolása buffereken keresztül a CPU feladata, a CUDA-hoz hasonlóan.

Számos megjelenítő motor áll rendelkezésre a szimuláció köztes és végeredményének megjelenítésére. Ezek többsége megfelelne céljaimnak, ezért azt választottam, ami a legjobban alkalmazkodik az eddigi környezetbe. A fenti döntések következményeként a megjelenítésre az elterjedt Unity keretrendszert használtam. A Unity egy általános célú engine, valós idejű fejlesztői platform, ahol a fejlesztés elsősorban C# nyelven zajlik. Rugalmas, platform független, gyors, és minden szükséges eszközt egy helyen biztosít (Integrated Development Environment). 2D és 3D grafikát is támogat, széles körben szolgál szimulációk, videójátékok, filmek és más egyéb vizuális programok alapjaként. A mögötte álló jelentős közösségnek köszönhetően bármilyen problémára számtalan megoldás, eszköz és oktatóanyag áll rendelkezésre.

5.3. Eljárás

A szimuláció matematikai alapjául O'Brien és Hodgins [26] által bevezetett hydrostatic pipe-model szolgál, ami igény és lehetőség szerint módosítható a környezethez és az elvárásokhoz igazítva. Ez az eljárás kétdimenziós mátrixokkal dolgozik, így jól párhuzamosítható, könnyen implementálható GPU-s környezetben. A mátrixon belül minden cella a talaj és a víz konstans tulajdonságait - mint egy „oszlop” - tárolja. Az oszlopokat virtuális csövek kötik össze mind a 4 szomszédjukkal. A szomszédos oszlopok között a nyomás alapján határozzuk meg, hogy a csövek merre és mennyi vizet (és a vízzel együtt hordalékot) szállítanak.

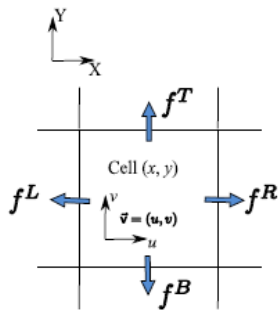
Belátható, hogy a háromdimenziós megoldásokhoz képest ennek sokkal kisebb a memória és a számítás igénye, a műveletek könnyen elvégezhetőek a szomszédok között, kevesebb iterációval. [32]

Hátrány viszont, hogy sík és üres (vízmentes) terepen sok fölösleges adatot tárolunk; magasabb területeken kényes a felbontásra, alacsony felbontás esetén pontatlanságokat figyelhetünk meg, a felbontás növelésével viszont négyzetesen növekszik a memóriaigénye is.

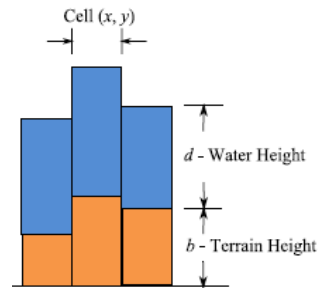
A hátrányok kiküszöbölésére a felhasználó számára lehetőség nyílik a felbontás kiválasztására. A memóriaigény csökkentésének megoldására a felszínt kisebb területekre osztom fel, ahol szükség szerint végezhető el a számítások.

A modell a talajt és víztömeget vertikális rektilineáris oszlopokba rendezi állandó fizikai tulajdonságokkal:

- talaj magassága b
- víz magassága d
- szállított hordalék mennyisége s
- talaj keménysége r
- sebesség vektor $\vec{v}$
- kiáramló fluviális fluxus $F_f(f_f^L, f_f^R, f_f^T, f_f^B)$
- kiáramló termáleróziós fluxus $F_s(f_s^L, f_s^R, f_s^T, f_s^B)$



1. ábra Rektilineáris oszlop felülről



2. ábra Rektilineáris oszlop oldalról

Az oszlopok területe konstans, ezeket virtuális csövek kötik össze négy szomszédjukkal. Nyomáskülönbség hatására, amit az oszlopok közötti vízszint különbsége, vagy más külső erők (pl. gravitáció) okoznak, a víz átáramlik ezeken a csöveken.

A szimuláció több lépésben zajlik, melynek sorrendje meghatározott:

1. vízszint növelése (eső)
2. vízmozgás, sebesség kiszámítása
3. vízszint aktualizálása
4. eróziós-depozíciós folyamat
5. párolgás
6. talaj elaprózódás
7. hordalék szállítása
8. termálerózió

5.3.1. Vízszint növelése

A szimuláció minden iterációjában, minden cellában a megadott konstans értékkel növeljük a vízszintet. d_1 a jelenlegi vízszint, d_2 az új számított érték.

$$d_2(x, y) = d_1(x, y) + K_r$$

5.3.2. Fluxus és sebesség számítása

Annak érdekében, hogy a vízszintet frissíteni tudjuk, ki kell számítani a fluxusokat. Minden iterációban a fluxust szorozzuk az oszlopok közti nyomáskülönbséggel. Ha a kiáramló fluxus negatív, az értéket K_f arányszámmal csökkentjük. A_p a virtuális cső keresztmetszete, l a virtuális cső hossza, g a gravitáció, h a szintkülönbség.

$$K_f = \min\left(1, \frac{d_1 * l_x l_y}{(f^L + f^R + f^T + f^B) * \Delta t}\right)$$

$$f_{f(t+\Delta t)}^i(x, y) = \max\left(0, f_{f(t)}^i(x, y) + \Delta t * A_p * \frac{g * \Delta h^i(x, y)}{l}\right), i = L, R, T, B$$

$$f_{f(t+\Delta t)}^i(x, y) = K_f * f_{f(t)}^i(x, y), i = L, R, T, B$$

5.3.3. Vízszint aktualizálása

A kimenő és bejövő fluxusok különbsége adja meg a víz V_l volumenjének változását.

$$\Delta V_l(x, y) = \Delta t * (\sum F_{in} - \sum F_{out})$$

A vízszint új magasságát ennek ismeretében a következő képlet adja meg:

$$d_2(x, y) = d_1(x, y) + \frac{\Delta V(x, y)}{l_x l_y}$$

5.3.4. Erózió-depozíció számítása

A folyó víz erodálja a talajt, felveszi, szállítja, majd valahol lerakja. A jelenséget elsősorban a víz hordalékszállítási kapacitása (C) befolyásolja.

$$C(x, y) = K_c * \sin(\alpha(x, y)) * |\vec{v}(x, y)|$$

ahol K_c a hordalék kapacitás konstans, $\alpha(x, y)$ a lokális dőlés szög, és $|\vec{v}(x, y)|$ a sebességvektor nagysága. Az így kapott értéket összehasonlítjuk a jelenleg szállított hordalék értékével (s). Amennyiben C kisebb vagy egyenlő ezzel, depozíció; másik esetben erózió történik.

5.3.5. Párolgás

A szimuláció alatt a nyomás és a hőmérséklet változatlan, a számításokban nem kapnak szerepet. K_e a párolgás konstans.

$$d_{t+\Delta t} = d_2(x, y) * (1 - K_e * \Delta t)$$

5.3.6. Talaj elaprózódás

A víz folyamatos közetre kifejtett erejének eredménye a talaj elaprózódása, puhulás. K_s hordalék elaprózódási konstans.

$$r_{t+\Delta t}(x, y) = r_t(x, y) - \Delta t * K_s * (s_t - C)$$

5.3.7. Hordalék szállítása

Ez a lépés a következő advekcíós egyenlettel írható fel:

$$\frac{\partial s}{\partial t} + (\vec{v} * \nabla s) = 0$$

A szemi-Lagrange advekcíós módszert felhasználva:

$$s_{t+\Delta t}(x, y) = s_1(x - u * \Delta t, y - v * \Delta t)$$

Amennyiben ez a határon kívülre esik, a négy legközelebbi szomszéd interpolációja alapján számítjuk ki a képletet.

5.3.8. Termálerózió

Első lépésben összevetjük az oszlop és szomszédjai magasságát. Ennek segítségével meghatározzuk mozgatandó mennyiséget. A_c az oszlop keresztmetszete, K_t a termáleróziós ráta, H az adott oszlop és legalacsonyabb szomszédja magasságának különbsége.

$$\Delta f_{s(t+\Delta t)}^i(x, y) = A_c * \Delta t * K_t * r_t(x, y) * \frac{H}{2}$$

Amennyiben az anyag viszkozitása nagyobb a lejtőszögből meghatározott határértéknél, ez a mennyiség az alacsonyabb oszlopba vándorol. A folyamat hosszútávú eredménye egy szabályos, a szög által meghatározott lejtő.

6. MEGVALÓSÍTÁS

6.1. Adatok

A szimuláció adatait 4 írható-olvasható textúra tartalmazza, a textúrák pixelei az egyes csatornákon (R,G,B,A) egy szimulált oszlop tulajdonságait tárolják.

- T1 magasságtérkép: R – talajszint, G – vízszint, B – szállított hordalék mennyisége, A – talaj ellenállása
- T2 sebesség vektor: R – x komponens, G – y komponens
- T3 víz fluxus térkép: R – bal, G – jobb, B – felső, A – alsó
- T4 talaj fluxus térkép: R – bal, G – jobb, B – felső, A – alsó

A magasságtérképet a T1 első két csatornájára töltjük be. A textúra közös, az első két csatornát használja a megjelenítési réteg a pálya kirajzolásához.

A talajszint és a felette elhelyezkedő vízoszlop adják meg a teljes magasságot. A szállított hordalék mennyisége a szimuláció elején mindig 0. A talaj ellenállása a választott beállítás alapján lesz meghatározva minden oszlopra, ez bizonyos mértékig megváltoztatható futásidő alatt. Ezt különböző értékekre beállítva szimulálható közetfajta, talajösszetétel, flóra megléte.

A sebességvektor a víz mozgásának irányát és nagyságát adja meg.

A fluxus térkép az adott cellából kiáramló víz mértékét tárolja. A befolyó víz mértékét a szomszédok értékeiből határozhatjuk meg.

A talaj fluxus térképe ugyanezt a feladatot látja el, ez viszont a talaj (anyag) mozgására vonatkozik.

6.2. Implementáció

6.2.1. Áttekintés

A Unity terében objektumok léteznek, ezekhez komponensek és scriptek (általunk létrehozott komponensek) vannak csatolva, amik ezeknek a viselkedését határozzák meg, más komponenseket módosítanak.

A programnak három fő, jól elkülönített része van: beállítások, szimuláció (logika és alkalmazási réteg) és megjelenítés. A beállítások betöltése, mentése és átadása a Settings Manager objektum feladata. A legfontosabb objektum a Simulation, a szimuláció inicializálásáért és működéséért felel. A megjelenítés procedurális dinamikus meshek segítségével történik, amit a Mesh Generator hoz létre.

Ezekon felül vannak még kisebb, segédjellegű szkriptek, amiket a UI-hoz használunk:

- Collapse
- Dropdown
- SimUIElements
- Slider
- Tabs

6.2.2. Beállítások

Simulation settings: A szimuláció összes paraméterét tartalmazza, nem tartalmaz logikát.

Settings manager: Egy simulation settings instance-t és az ehhez tartozó logikát tartalmazza; beállítások mentése és betöltése fájlból, UI elemekkel való kapcsolat, beállítások rendelkezésre bocsátása a szimuláció felé. A konfigurációs fájl (config.cfg) a főkönyvtáron belül található, módosítható, elvesztése vagy sérülése esetén alaphelyzetbe állítható.

A felhasználó a különböző UI elemekkel futásidő alatt módosíthatja a szimuláció beállításait

A térképek betöltéséért Map Manager felelős. Új térképeket a főkönyvtárból elérhető a „\Maps” mappába helyezve adhatunk hozzá (a program „*.jpg” és „*.png” formátumokat tud feldolgozni).

Name	Type	Size
Maps	File folder	
MonoBleedingEdge	File folder	
RiverSim_Data	File folder	
Screenshots	File folder	
config.cfg	CFG File	1 KB
RiverSim.exe	Application	639 KB
UnityCrashHandler64.exe	Application	1,208 KB
UnityPlayer.dll	Application exten...	27,648 KB

3. ábra. Főkönyvtár

6.2.3. Szimuláció CPU

A Simulation osztály felel az adatok és a kernelek inicializálásáért, a szimuláció működéséért. Lekéri a beállításokat a Settings managertől, amennyiben nem áll rendelkezésre, létrehozza az alapértelmezett beállításokat. Létrehozza a 4 írható-olvasható textúrát, amik a szimulált terület adatait fogják tárolni.

```
stateTexture = new RenderTexture(width, height, 0, RenderTextureFormat.ARGBFloat)
{
    enableRandomWrite = true,
    filterMode = FilterMode.Bilinear,
    wrapMode = TextureWrapMode.Clamp
};
```

4. ábra Textúra létrehozása

Átmásolja a kiválasztott térképet a fő textúrára. Inicializálja a compute shadert, beállítja a globális változókat, minden kernelnek átadja a textúrákat.

```
ErosionComputeShader.SetTexture(j, "HeightMap", stateTexture);
ErosionComputeShader.SetTexture(j, "VelocityMap", velocityTexture);
ErosionComputeShader.SetTexture(j, "FluxMap", waterFluxTexture);
ErosionComputeShader.SetTexture(j, "TerrainFluxMap", terrainFluxTexture);
```

5. ábra Textúrák átadása a kerneleknek

A fő textúrát hozzárendeli a talaj és víz megjelenítését szolgáló materialokhoz. A kerneleket (magát a szimulációt) a program ideje alatt ciklusosan futtatja. A beállításokat minden kernel indítás előtt beállítjuk, mivel ezek futásidő alatt megváltoztathatók.

6.2.4. Szimuláció GPU - Compute Shader

Maga a szimuláció egyetlen Compute Shaderen történik. A kód HLSL-ben íródott, ami a GPU által értelmezhető kód. A GPU nem objektum-orientáltan működik, adatokat másolunk át a memóriájába, és ezeken egyszerű utasítások segítségével matematikai műveleteket hajtunk végre. A GPU textúrákon (mátrixok) lebegőpontos számokkal dolgozik. A műveleteket az átadott textúrákon 8 kernel végzi. A 4 textúra egyes csatornái (RGBA) az egyes tulajdonságok, minden pixel pedig egy adott oszlop (terület) fizikai jellemzőit tárolja.

Az elindított kernelek minden pontban leolvassák az adatokat a textúrákról, elvégzik a matematikai műveleteket (a hydrostatic pipe-model lépéseit), majd az új aktuális állapotokat visszaírják a textúrákra, a 8 kernel mindegyike a hydrostatic pipe-model egy-egy lépését [5.3] valósítja meg.

```
[NUMTHREADS]
void Rain(uint3 id : SV_DispatchThreadID)
{
    float4 currentState = CURRENT_SAMPLE(HeightMap);
    WATER_HEIGHT(currentState) += timeDelta * rainRate;
    CURRENT_SAMPLE(HeightMap) = currentState;
}
```

6. ábra. Eső, vízszint növelése

Az eső a legelső és legegyszerűbb kernel. Első lépésben a megadott helyen kiolvassuk a cella állapotát, megnöveljük a vízszintet az eső állandóval, amit az eltelt idővel arányosítunk, majd az új állapotot visszaírjuk a textúrára. A többi kernel pedig a többi 7 lépés képleteit implementálja.

A szimuláció bővítése egyszerű, bármikor beiktatható egy újabb kernel.

6.2.5. Megjelenítés

A Mesh Generator egy megadott méretű procedurális mesh-t (lapot) hoz létre a térben (pontok és élek halmaza, aminek eredménye komplex 3D-s alakzat).

A Chunk Generator azonos méretű chunk-kokra osztja fel a térképet, a Mesh Generatorral mindegyikhez létrehoz egy mesh-t, beállítja a LOD (Level of Detail) értékeket, és hozzájuk rendeli a talaj és víz megjelenítését szolgáló materialokat.

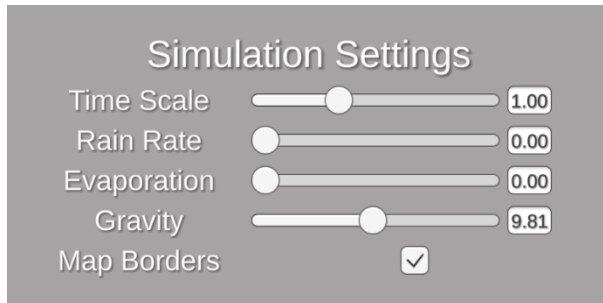
Surface Shader: szintén HSL nyelvű shaderek, de struktúrájuk más, felhasználásuk leginkább grafikai jellegű.

Terrain és Water (materialok és surface shaderek): a materialok és az azokhoz tartozó surface shaderek dinamikusan változtatják a procedurális mesheket (alakját, kinézetét) a hozzájuk rendelt stateTexture változásai alapján.

7. MEGVALÓSULT PROGRAM BEMUTATÁSA

A felhasználónak a menüben lehetősége van a globális paraméterek beállítására. Legtöbbjük futásidő alatt is módosítható, és azonnal érvénybe lép. A szimuláció paraméterei három nagy csoportba sorolhatók:

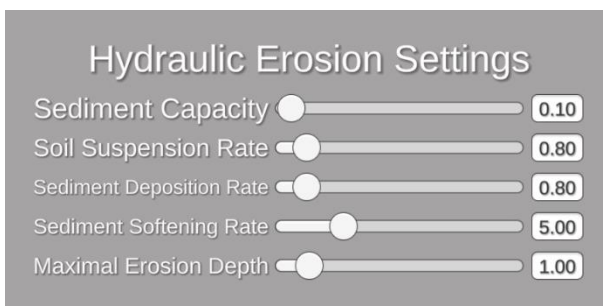
1. Szimuláció általános beállításai:



7. ábra. Szimuláció általános beállításai

- Time Scale – idő múlásának szorzója [0-3x]
- Rain Rate – eső mértéke
- Evaporation – párolgás mértéke
- Gravity – gravitáció
- Map Borders – a pálya határok, ne folyjon ki a víz a pályáról

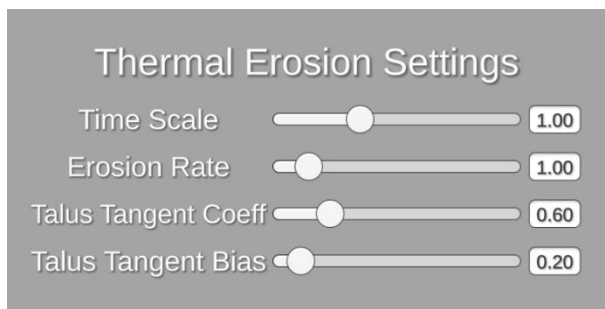
2. Vízerózió beállításai:



8. ábra. Vízerózió beállításai

- Sediment Capacity – hordalék kapacitás konstans
- Soil Suspension Rate – hordalék felvételének mértéke
- Sediment Deposition Rate – hordalék lerakásának mértéke
- Sediment Softening Rate – hordalék oldódásának mértéke
- Maximal Erosion Depth – erózió mélységének mértéke

3. Termálerózió beállításai

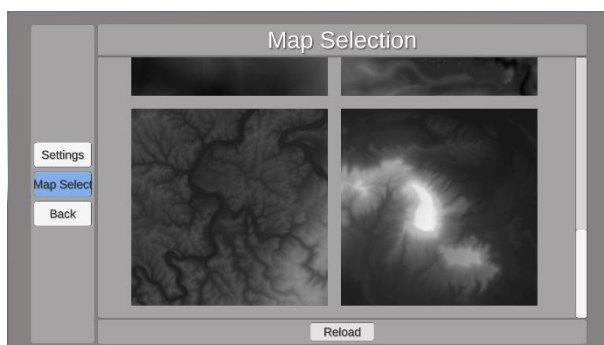


9. ábra. Termálerózió beállításai

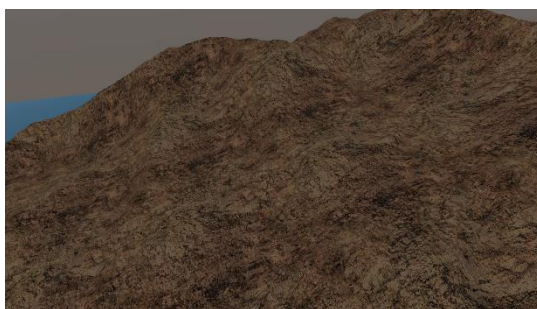
- Time Scale (Thermal Erosion) – idő múlásának szorzója [0-3x] (földmozgás)
- Erosion Rate – eróziós ráta (földmozgás)
- Talus Tangent Coeff – lejtőszög tangens együttható
- Talus Tangent Bias – lejtőszög tangens torzítás

4. Térkép betöltése

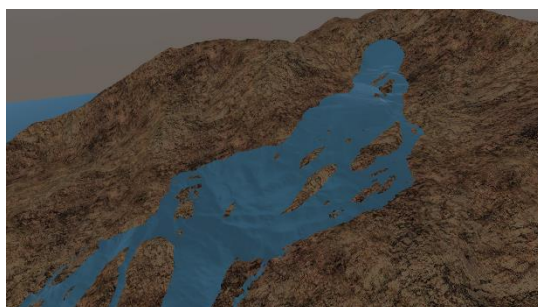
A felhasználónak lehetősége van meglévő pályákból választani vagy különböző domborzati térképek hozzáadására, amiből aztán a Map Select menüben választhat. Alternatív módon a felhasználó létrehozhat egy procedurális pályát is pár kezdeti paraméter beállításával. A felhasználó különböző eszközök segítségével módosíthatja a domborzatot valós időben (anyag hozzáadásával vagy elvételével hegyek, völgyek képzése), lehetőség van egy „vödör” segítségével víz hozzáadására, forrás, illetve víznyelő lehelyezésére.



10. ábra Térkép kiválasztása



11. ábra. Hegyoldal erózió előtt



12. ábra. Létrejött folyó.

A víz-, és termálerózió hatása észlelhető



13. ábra. A folyó kiapadása után az erózió hatása jól látható

A képek a programba beépített képernyőkép mentővel készültek, képernyőkép a 'C' betű lenyomásával készíthető, a képek a főkönyvtár „\Screenshots” mappában található.

Az elkészült programot bemutató videók megtekinthetők az alábbi linkre kattintva:
<https://youtube.com/playlist?list=PLcWGQUKLD-GciG8Y0d7hj2vVjEGDXt7Y4&feature=shares>

8. ÉRTÉKELÉS

8.1. Tesztek futtatása

A fejlesztés során célom volt egy alacsonyabb gépigényű program létrehozása, hogy minél szélesebb körben felhasználható legyen illetve minél nagyobb és komplexebb szimulációkat lehessen vele lefuttatni.

Az alábbi hardware konfigurációkon végeztem teszteket. A programot elindítottam, betöltöttem különböző pályákat, az összes kernelt futtattam, kipróbáltam a felhasználói eszközöket, procedurálisan létrehoztam egy térképet. Stabil és interaktív framerate mellett (30 fps) az elvárásoknak megfelelően működött.

GPU	CPU	RAM	Eszköz
AMD Radeon R7 200 series	Intel Core i5-3570	8 GB	PC
AMD Radeon RX 580 series	Intel Core i5-9400F	16 GB	PC
Intel Iris Xe	Intel Core i7-1255U	16 GB	Laptop
AMD Radeon 530	Intel Core i7-7500U	16 GB	Laptop
AMD Athlon II X3 455	AMD Radeon R7 200	8 GB	PC
NVIDIA GeForce GTX 970	Intel Core i5-6600k	32 GB	PC

2. táblázat Tesztelt eszközök specifikációi

8.2. Eredmények értékelése

A cél egy olyan szimuláció létrehozása volt, ami az átlagember számára is elérhető, ugyanakkor rendelkezik olyan matematikai pontossággal, hogy az a tudományok terén is helyt álljon.

A program futtatható bármilyen számítógépen, ami rendelkezik videokártyával. Eszközök széles tárháza áll rendelkezésre a felhasználó számára: a különböző paraméterek módosítása, a térkép valós idejű manipulálása, magasságtérképek betöltése, procedurális pályagenerálás, folyók létrehozása, eső. A szimuláció felbontása és pontossága igény szerint igazítható a célhardware-hez.

Az egyszerű kezelés, alacsony gépigény, nem helyhez kötött (laptopon történő) használat lehetőséget teremt a kutatás mellett az oktatás különböző szintjein, területein történő alkalmazásra: hagyományos tanórán, távoktatásban, akár terepen is.

9. TOVÁBBFEJLESZTÉSI LEHETŐSÉGEK

A program számos továbbfejlesztési lehetőséget rejt magában. Ezek megvalósítására a rendelkezésre álló időben nem kerülhetett sor, azonban a későbbiekben mindenképp érdemes lehet foglalkozni velük.

9.1. Eszközök, effektek

Létrehozható egy pályaszerkesztő, amivel személyre szabott térképek készíthetők és exportálhatók, amik nem csak a talaj magasságát, de más szimulációs adatokat is tartalmazhatnak, pl. talaj összetétele. Hosszabb távú fejlesztés esetén más paraméterek is felmerülhetnek (pl. kőzetfajták, rétegzettség, flóra).

Lehetséges esztétikai fejlesztések részeként új vizuális effektekkel bővíthető a program: hab és fröccsenés nagylejtésű területeken, buborékok, tükröződés és egyéb fényeffektek.

9.2. Szimuláció bővítése

A logikai réteg teljesen külön van választva a program többi részétől. Megfelelő matematikai és földrajzi (és minimális informatikai) ismeretek birtokában a szimuláció bővíthető, módosítható a shaderek segítségével. Lehetséges teljesen új Compute Shaderek hozzáfűzése, amiknek a kerneljeit a hozzájuk tartozó konfigurációs fájlból lehet majd betölteni és ütemezni, ezzel kiterjeszteni a szimulációt. Ezáltal lehetőség lesz a szimulációt új paraméterekkel kiegészíteni, amik módosítják a meglévő funkciókat, vagy akár teljesen új funkciók részét képezhetik (pl. hőmérséklet, nyomás).

Egy könnyen megvalósítható fejlesztés lenne más folyadékok, képlékeny anyagok és azok munkájának modellezése a már elkészült szimulációs környezetben (pl. vulkán, vulkánkitörés, láva).

Egyetlen új kernellel szimulálható sárlavina, földcsuszamlás. Ez egy aktuális és súlyos probléma, mely komoly károkat okoz világszerte.

9.3. Lehetséges felhasználás

A klímaváltozás következtében nagyon heves zivatarok is kialakulnak, hirtelen nagy mennyiségben hullik csapadék olyan területekre, ahol korábban csak kevés eső esett, így az ember birtokba vette. Településvédelmi, katasztrófa megelőzési céllal érdemes modellezni, mit okoz adott területen a különböző mértékű és időtartamú eső.

Napjainkban a videójáték-iparban egyre nagyobb az igény a különböző víztömegek valóságghű megjelenítésére, szimulálására. A modell fluviális szimuláció része jól integrálható játékokba optimális számítás- és memóriaigénye miatt.

Hivatkozások

- [1] S. Gönczy és K. Szalai, Geomorfológiai fogalomgyűjtemény (oktatási segédanyag a földrajz szakos főiskolai hallgatók számára), Beregszász: Kárpátaljai Magyar Pedagógusszövetség Tankönyv- és Taneszköztanácsa, 2004, p. 22.
- [2] D. Lóczy, „Geomorfológia,” in *Földtudományi alapismeretek*, G. Konrád, Szerk.
- [3] A. Nemerkenyi, Általános természetföldrajz a középiskolák I. osztálya számára, Budapest: Nemzeti Tankönyvkiadó, 1997, pp. 142-148.
- [4] J. Lóki és J. Szabó, A külső erők geomorfológiája, Debrecen: Debreceni Egyetem Kossuth Egyetemi Kiadó, 2006, p. 93.
- [5] I. Bihariné dr. Krekó és G. Kanczler, Természetismeret – I. Természetföldrajz az ELTE TÓK hallgatóinak, Budapest: ELTE Eötvös Kiadó, 2017, pp. 36-39.
- [6] T. Petró, A magyarországi árvízvédelmi fejlesztések, a kapcsolódó lakosságvédelmi feladatrendszer újszerű megközelítése Doktori (PhD) Értekezés, Budapest: Nemzeti Közszerzői Egyetem Hadtudományi és Honvédtisztviselői Kar Katonai Műszaki Doktori Iskola, 2017, p. 202.
- [7] I. Arday és E. Rózsa, Földrajz B Társadalomföldrajz természetföldrajzi alapozással a szakközépiskolák I. osztálya számára, Budapest: Műszaki Könyvkiadó, 1998, pp. 44-45.
- [8] K. Körmendy, *A szárazföldek felszíni vizei - óravázlat*, Budapesti Egyetemi Katolikus Gimnázium és Kollégium.
- [9] J. Dr. Chelnoky, „A folyók szakasz-jellegének összefüggése a szabályozással és öntözéssel,” in *Vízügyi Közlemények I. sz.*, Budapest, M. Kir. Földművelésügyi Minisztérium Vízügyi Műszaki Csoportja, 1939, pp. 5-25.
- [10] B. András, IKT és oktatás - Együtt vagy egymás mellett?, Belvedere Meridionale, 2017.
- [11] A. Békési, Az IKT eszközök az oktatásban, Debrecen: Debreceni Egyetem Matematikai és Informatikai intézet, 2010.
- [12] É. Tót, Számítógépek az iskolában, Budapest: Oktatókutató Intézet, 2001, p. 7.
- [13] F. Csongor és E. Galgóczi, „Tünetkezelés - 2020-21 Pedagógus kutatás gyorsjelentés,” 21 Kutatóközpont, 2021.
- [14] Z. Varga és H. Czédli, „Gondolatok a természettudományi tárgyak oktatásához,” in *Térinformatikai Konferencia és Szakkiállítás*, Debrecen, 2019 május 23-24.
- [15] L. Bertalan, C. A. Dr. Tóth és G. Dr. Szabó, „Fluviális felszínfejlődési folyamatok modellezési lehetőségei folyóvízes laboratóriumban,” in *Az elmélet és a gyakorlat találkozása térinformatikában IV Térinformatikai Konferencia és Szakkiállítás*, 2013 május.

- [16] P. Prusinkiewicz és M. Hammel, „A Fractal Model of Mountains with Rivers,” in *Proceedings of Graphics Interface '93*, Toronto, Canada, 19-21 May 1993, pp. 174-180.
- [17] K. Nagashima, „Computer generation of eroded valley and mountain terrains,” *The Visual Computer*, %1. kötetVol. 13, %1. számIssue 9-10, pp. 456-464, 1998.
- [18] B. Benes és R. Forsbach, „Visual Simulation of Hydraulic Erosion,” *Journal of WSCG*, %1. kötet10., %1. szám1-2., pp. 79-86., 2002.
- [19] N. Chiba, K. Muraoka és K. Fujita, „An Erosion Model Based on Velocity Fields for the Visual Simulation of Mountain Scenery,” *The Journal of Visualization and Computer Animation*, %1. kötet9, pp. 185-194, 1998.
- [20] B. Benes, T. Vaclav, J. Hornys és S. K. Bhatia, „Hydraulic erosion,” *Computer Animation and Virtual Worlds*, %1. kötet17, pp. 99-108, 2006.
- [21] B. Neidhold, M. Wacker és O. Deussen, „Interactive physically based Fluid and Erosion Simulation,” Ireland, 2005.
- [22] M. J. Harris, „Fast Fluid Dynamics Simulation on the GPU,” in *GPU Gems*, %1. kötet1, Assison-Wesley Professional, 2004, pp. 637-666.
- [23] E. Wu, Y. Liu és X. Liu, „An improved study of real-time fluid simulation on GPU,” *Computer Animation and Virtual Worlds*, %1. kötet15, pp. 139-146, 2004.
- [24] W. Li, W. Xiaoming és K. Arie, „Implementing lattice Boltzmann computation on graphics hardware,” *The Visual Computer*, %1. kötet19, pp. 444-456, 2003.
- [25] B. Benes, „Real-time erosion using shallow water simulation,” Dublin, 2007.
- [26] J. F. O'Brien és J. K. Hodgins, „Dynamic simulation of splashing fluids,” in *Computer Animation '95*, Switzerland, 1995. április 19-21.
- [27] X. Mei, P. Decaudin és B.-G. Hu, „Fast Hydraulic Erosion Simulation and Visualization on GPU,” United States, 2007.
- [28] O. Stava, B. Benes, M. Brisbin és J. Krivanek, „Interactive Terrain Modeling Using Hydraulic Erosion,” 2008.
- [29] B. Jákó, „Fast Hydraulic and Thermal Erosion on the GPU,” 2011.
- [30] D. Mcleod, H. Chen és N. Hu, „Interactive Hydraulic Erosion Simulator,” 2020. [Online].
- [31] J. I. Agulleiro, F. Vazquez, E. M. Garzon és J. J. Fernandez, „Hybrid computing: CPU+GPU co-processing and its application to tomographic reconstruction,” *Ultramicroscopy*, %1. kötet115., pp. 109-114., 2012.
- [32] T. Telbisz, B. Székely és G. Timár, Digitális terepmodellek - Adat, látvány, elemzés, Budapest: ELTE TTK Földrajz- és Földtudományi Intézet, 2013.
- [33] A. Buda, IKT és oktatás - Együtt vagy egymás mellett?, Szeged: Belvedere Meridionale, 2017.
- [34] C. Fényes és E. Galgóczi, Tünetkezelés - 2020-21 Pedagógus kutatás gyorsjelentés, 21 Kutatóközpont, 2021.

ÁBRAJEGYZÉK

1. ÁBRA REKTLINEÁRIS OSZLOP FELÜLRŐL.....	17
2. ÁBRA REKTLINEÁRIS OSZLOP OLDALRÓL.....	17
3. ÁBRA. FŐKÖNYVTÁR	23
4. ÁBRA TEXTÚRA LÉTREHOZÁSA	24
5. ÁBRA TEXTÚRÁK ÁTADÁSA A KERNELEKNEK	24
6. ÁBRA. ESŐ, VÍZSSZINT NÖVELÉSE.....	25
7. ÁBRA. SZIMULÁCIÓ ÁLTALÁNOS BEÁLLÍTÁSAI	27
8. ÁBRA. VÍZERÓZIÓ BEÁLLÍTÁSAI	27
9. ÁBRA. TERMÁLERÓZIÓ BEÁLLÍTÁSAI	28
10. ÁBRA TÉRKÉP KIVÁLASZTÁSA.....	28
11. ÁBRA. HEGYOLDAL ERÓZIÓ ELŐTT.....	29
12. ÁBRA. LÉTREJÖTT FOLYÓ.	29
13. ÁBRA. A FOLYÓ KIAPADÁSA UTÁN AZ ERÓZIÓ HATÁSA JÓL LÁTHATÓ	29