**Univerzita Hradec Králové**

**Fakulta informatiky a managementu**

**Katedra informačních technologií**

**Simulace vybraných jaterních funkcí**

Bakalářská práce

Autor: Lukáš Hromek

Studijní obor: Aplikovaná informatika, bakalářský

Vedoucí práce: doc. Ing. Hana Tomášková, Ph.D.

Hradec Králové Leden 2024

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 28.1.2024 Lukáš Hromek

Poděkování:   
Děkuji vedoucí bakalářské práce doc. Ing. Haně Tomáškové, Ph.D. za vedení práce a hodnotné tipy, rady a trpělivost během zpracovávání této práce. Dále bych rád poděkoval Ing. Brunu Ježkovi Ph.D. za rady ke zpracování takovéto simulace.

Anotace:

Bakalářská práce se zabývá simulací činností jater. Cílem je prozkoumat nejlepší možnosti simulace orgánů v počítačovém prostředí a přiblížit čtenáři funkci jater až po samotné programování prostředí (jater), krve a částic v ní. Práce se bude zabývat spíše technickou částí, jak vytvořit simulaci orgánu. Nebude se tolik řešit biochemická správnost pochodů.

Annotation:

*Title:* *Simulation of selected liver functions*

The bachelor's thesis deals with the simulation of liver activities. The goal is to explore the best possibilities of organ simulation in a computer environment and to introduce the reader to the function of the liver up to the very programming of the environment (liver), blood and particles in it. The work will deal more with the technical part of how to create a simulation of the organ. The biochemical correctness of the marches will not be addressed so much.

Obsah

1. Úvod
2. Cíl Práce
3. Metodika
4. Játra
   1. Stavba Jater
      1. Hepatocyty
      2. Kupfferovy buňky
      3. Endotelové buňky
      4. Itóovy buňky
   2. Krev a její složení
      1. Plazma
      2. Červené krvinky
      3. Bílé krvinky
      4. Krevní destičky
   3. Jaterní onemocnění
      1. Cirhóza
5. Možnosti simulace orgánů
   1. Zpracování modelu
      1. Triangulace
      2. Generování na model
      3. MetaBall
   2. Zobrazení
      1. Real-Time
      2. Předpočítávání
   3. Pohled na tkáň
      1. Celý orgán
      2. Kus tkáně
      3. Mikroskopický pohled
6. Modely
   1. Modelování
   2. Zakoupení
7. Výběr Prostředí
   1. Nástroje pro 3D vývoj a programování  
      5.1.1 Unity  
      5.1.2 Unreal Engine 5  
      5.1.3 Open Gl
   2. Vytyčení kritérií pro hodnocení
   3. Hodnocení
8. Praktická část
   1. Zpracování modelů pro práci
      1. Čistá triangulace
      2. Generace objektů na model
      3. Metaballs
   2. Model krve
      1. Částicový systém
   3. Zpracování změn
      1. Zasažení stěny
      2. Časové
   4. Chování krve
   5. Věrnost realitě
9. Seznam použité literatury
10. Seznam Ukázek kódu
11. Seznam obrázků

1.Úvod

Všichni lidé vědí, že jejich tělo je tvořeno orgány, kde každý z nich plní svoji jedinečnou funkci. Už méně lidí ale ví, jak jejich orgány fungují a co jim prospívá. Játra jsou jedním z nejdůležitějších orgánů, který čistí tělo od všech tělu neprospěšných či nezpracovatelných látek, a přemění je látky které se dají zpracovat.

V dnešní době snadno přístupného alkoholu, karcinogenních látek a až přílišné konzumaci potravy, jsou játra čím dál tím víc zatěžována. Díky věrným simulacím by mohli lidé i lékaři lépe porozumět povaze jaterních nemocí a zlepšit povědomí o látkách a potravinách, které jim obecně škodí, či naopak prospívají. Díky širšímu pochopení veřejnosti svých jater, by se mohlo spoustě těchto problémům předejít

V první části, práce seznámí s Játry. Jedná se o jaterní stavbu, jak které části fungují a co se simuluje

Druhá část přibližuje tvorbu či jiné získání 3D modelů, které se dají v práci použít. Mezi nejdůležitější se počítá model jater a způsob zobrazení krve

Třetí část práce se zabývat porovnáním nejznámějších prostředí, která jsou vhodná pro 3D simulace. Jsou to Unity, Unreal Engine 5, a open Gl.

Poslední část pojednává o implementování těchto postupů a porovnávání jejich vhodnosti k simulaci

2. Cíl práce

Cíl práce je přiblížit možnosti simulace jater, které jsou schopny reprezentovat jejich fyziologické procesy. Simulace by měla být použitelná pro pochopení a přiblížení práce jater v našem těle.

Přesnější cíle se dají shrnout do pochopení jater a jejich případných poškození. Seznámení se s nejznámějšími způsoby, jak pracovat vypracovat tuto simulaci. Výběru prostředí nejvhodnějším pro zpracování této simulace. Popsáním samotného modelu jater a jeho zpracováním který se se dá použít pro simulace a pozdější rozšíření, a nakonec porovnání již seznámených způsobů, jak vytvořit takovou simulaci

3. Metodika

Metodika zpracování bakalářská práce se zaměřuje na sběr informací o játrech z článků a pojednáních. Dále na porovnání třech prostředí pro tvorbu Her či simulací a proběhne seznámení s vybranými prostředími, a vytyčení kritérií a vyhodnocení.

Data jsou získávány z článků o počítačových simulacích a o anatomii jater

4. Játra

Játra jsou důležitý orgán, který plní důležitou funkci pro udržení zdraví organismu. Zpracovávají živiny vstřebané ze střev a přeměňují je na jiné, pro tělo použitelné substance. Provádí i plno dalších majoritních funkcí, jako detoxifikaci, metabolizaci léků a produkování žluči. Žluč slouží ke správnému trávení a absorpci vitamínů rozpustných v tucích.

4.1 Stavba Jater

Játra jsou tvořeny 2 jaternímy laloky (lobus dexter, lobus sinister). Zde probíhají všechny funkce jater, prováděny jaternímy buňkami(hepatocyty). Do jater je přiváděna čerstvá krev Jaterní tepnou (arteria hepatica), která je okysličuje a vyživuje. Portálová žíla (vena portae) přivádí krev z jater s natrávenými částmi potravy, které se později v játrech zpracovávají. Jaterní žíla (vena hepatica) Odvádí všechnu krev z jater, která byla do jater přivedena přes portálovou žílu. Z jater je sbírána žluč do žlučníku (vesica biliaris).

Obsah obrázku klipart, kreslené, ilustrace, design

Popis byl vytvořen automaticky

Obrázek 1 Stavba jater, převzato [1]

4.1.1 Hepatocyty

Základním stavebním kamenem jater jsou Hepatocyty. Tvoří téměř 80 % objemu jater, který se může s poškozením snižovat. Pokrývají stěny všech kánálků v játrech, kde na vnitřní straně vstřebávají krev ze střev, a na vnější jsou vyživovány okysličenou krví ze srdce.

Metabolizují živiny způsobem, kdy přemění glukózu z jídla na glykogen pro skladování, a nazpět uvolňují glukózu přu hladovění. Dále detoxifikují a neutralizují škodlivé látky přijaté tělem, jako léky a toxiny. Zpracovávají alkohol metabolizováním acetaldehydu, a přeměnou na méně škodlivé látky, které se dají vyloučit. Produkují žluč pro absorpci tuků a vylučování odpadů z těla. Ukládají v sobě vitamíny a minerály, aby jich v případě potřeby tělo nemělo nedostatek těchto látek. Spolupracují s Kupfferovými Buňkami na obraně těla a regulují také krevní srážlivost.

4.1.2 Kupfferovy buňky

Jsou imunitní buňky, které mají za úkol čištění krve od patogenů a starých či poškozených červených krvinek. Při jejich nadměrné aktivaci třeba při přílišné konzumaci alkoholu dochází k mnoha reakcím, které způsobují ztučnění jater, první stádium poškození jater, které vede k cirhóze.

4.1.3 Endotelové buňky

Endotelové se nacházejí v krevním řečišti. Tvoří bariéru mezi krví a okolní tkání a řídí tak které látky vstoupí do tkáně a které ne. Podílejí se také na růstu cév a hojení. Regulují průtok a srážlivost krve.

4.2.3 Itóovy buňky

4.2 Krev a její složení

Krev je vitální tekutina pro chod organismu. Přenáší se v ní téměř všechny látky, které tělo potřebuje pro svou výživu a správnou funkčnost. Je tvořena v kostní dřeni, je hnána srdcem a okysličována plícemi. Krev je složena ze 4 hlavních složek které mají svou specifickou roli pro udržení chodu organismu.

4.2.1 Plasma

Plasma je tekutina, která tvoří přes 55% objemu krve. Asi 90% jí tvoří voda, která rozpouští ostatní složky. Nachází se v ní také Proteiny, Elektrolyty, Živiny, Odpadní látky a Hormony potřebné pro regulaci tělesných funkcí.

4.2.2 Červené krvinky

Červené krvinky jsou bezjaderné buňky, které tvoří téměř celou druhou polovinu krve. Starají se hlavně o přenos kyslíku k buňkám, a odvod oxidu uhličitého zpět do plic.

4.2.3 Bílé krvinky

Zastupují necelé 1% krve, ale hrají důležitou roli v obranyschopnosti těla. Je jich více druhů a zastupují funkce jako neutralizace bakterií a patogenů, tvorba protilátek, přeměna na makrofágy a mírnění alergických reakcí.

4.2.4 Krevní destičky

V případě poškození, se zapojují krevní destičky. I jako nepatrná část krve, jsou krevní destičky nezbytné pro zastavení krvácení.  
4.3 Jaterní onemocnění

Většina jaterních onemocnění se dá vysvětlit patogeny, špatným životním stylem, genetikou nebo vším dohromady. Jedno z nejznámějších onemocnění je cirhóza, která vzniká ukládáním a hromaděním tuku v játrech. Má několik fází:

1. Steatóza – hromadění tuků v játrech
2. Alkoholová či nealkoholová Steatohepatitida – poškození jaterních buněk a zánět
3. Fibróza – tvorba jizevnaté tkáně, dá se zvrátit
4. Cirhóza – zjizvená tkáň nahradí normální tkáň jater. Většinou nezvratná

5. Možnosti simulace orgánů

Existuje několik možností, jak přistoupit k takovéto simulaci. Asi základní kritérium správné určení metody zobrazení, je určit si co chceme simulovat.

5.1 Možnosti zpracování modelu

5.1.1 Triangulace

Triangulace hraje zásadní roli v počítačové grafice a výpočetní geometrii tím, že rozkládá složité tvary a povrchy na jednodušší, nepřekrývající se trojúhelníky. Tato technika je nepostradatelná kvůli snadnosti a účinnosti vykreslování trojúhelníků, což jsou nejzákladnější polygonální tvary schopné definovat rovinu v trojrozměrném prostoru. Trojúhelníky slouží jako základní komponenty pro konstrukci 3D modelů, protože důsledně leží v jedné rovině, což zaručuje jasnost jejich zobrazení. Většina grafických nástrojů a rozhraní API, jako je OpenGL a DirectX, je speciálně optimalizována pro vykreslování trojúhelníků. Aplikace triangulace zahrnují generování sítě pro 3D modelování, modelování terénu využívající triangulované nepravidelné sítě (TIN), analýzu konečných prvků (FEA) ve strojírenství a 3D rekonstrukci v počítačovém vidění. K dosažení účinné a přesné triangulace se používají různé metody, jako je Delaunayova triangulace, ear clipping a Marching cubes. Je však třeba překonat výzvy, jako je správa složitých geometrií, nalezení rovnováhy mezi detaily a výkonem a zajištění stability.

5.1.2 Generování na povrch

5.1.3 Metaballs s Raymarching

Metaballs a raymarching představují sofistikované metodologie používané ve 3D simulacích k vytvoření plynulých, přirozených tvarů a složitých vizuálních efektů. Metakoule, neboli izoplochy, se vyznačují implicitními povrchy, které se hladce prolínají, když jsou v těsné blízkosti, což vede k měkkým tvarům podobným kapkám. Tato charakteristika prolínání činí metakoule vhodné pro emulaci tekutin, poddajných těles a dalších organických struktur v rámci počítačově generovaných snímků. Naopak raymarching slouží jako způsob vykreslování, který sleduje paprsky v celé scéně a počítá vzdálenost k nejbližšímu povrchu pomocí funkce vzdálenosti se znaménkem (SDF). Tato technika umožňuje efektivní vykreslování složitých, matematicky definovaných povrchů, jako jsou metakoule, bez spoléhání se na konvenční polygonální sítě. Prostřednictvím integrace metaballů a raymarchingu mohou 3D simulace dosáhnout vyšší úrovně složitosti a realismu, zejména ve scénářích zahrnujících dynamické interakce tekutin, transformace tvarů a hladké přechody mezi formami. Využití těchto metodologií zvyšuje vizuální kvalitu a přizpůsobivost 3D simulací a činí je použitelnými pro použití ve vizuálních efektech, vědecké vizualizaci a interaktivních médiích.

5.2 Zobrazení

5.2.1 Simulace v reálném čase

Simulace v reálném čase klade značné nároky na zdroje Grafické karty. Grafická karta je vyžadována pro konzistentní výpočet komplexních efektů, jako je osvětlení, stíny a fyzika pro každý snímek v dynamicky se měnících scénách, což vyžaduje značný výpočetní výkon, aby bylo zajištěno plynulé a přesné vykreslování v reálném čase. Na rozdíl od předkalkulačních technik, které vkládají efekty do statických textur nebo map, simulace v reálném čase zahrnuje průběžné výpočty, což vede k vyšší spotřebě energie a potenciálně sníženému výkonu. Přesto usnadňuje vývoj dynamičtějších a interaktivnějších prostředí, přičemž změny osvětlení, pohybu a dalších proměnných se okamžitě projeví v renderované scéně.

5.2.2 Předpočítání

Předvypočítávání obrazů, nabízí ve srovnání se simulací v reálném čase významnou výhodu z hlediska úspory zdrojů GPU. V aplikacích, jako jsou videohry nebo interaktivní simulace, které vyžadují zpracování v reálném čase, je GPU neustále zapojen do výpočtu osvětlení, stínů a různých efektů pro každý snímek, což vyžaduje značné množství výpočetních zdrojů. Předpočítáním těchto prvků a uložením výsledků jako statických textur nebo map je GPU vyžadováno pouze k vykreslení předem vypočítaných dat namísto provádění složitých výpočtů na místě. Tento přístup snižuje zátěž GPU, což má za následek plynulejší výkon, nižší spotřebu energie a schopnost pracovat na méně výkonném hardwaru při zachování vysoce kvalitního obrazu.

5.2.3 Video

5.3 Typ tkáně

5.3.1 Celý orgán

Celý orgán je nejširší pohled na orgán. Dají se na něm dobře základní funkce, jako model proud krve, případné rozsáhlejší poškození a Ideální zobrazení je na 3 D, Ale vyvstává tu problém s grafickou náročností při velké míře detailu.

5.3.2 Kus tkáně

Kus tkáně je menší detail, kde se již dají dobře zobrazit jednotlivé části krve a dobře ukazuje kooperaci buněk na jejich úrovni. Dá se s ním pracovat jako s 2D i 3D modelem.

5.3.3 Mikroskopické zobrazení

Zobrazení do nejmenšího detailu, které se dá rozdělit do více úrovní. Na té nejmenší, by se daly simulovat i jednotlivé prvky atomů jak do sebe zapadají a je proto vhodné pro biochemické reakce.

6.Modely

K simulaci jater je vhodné vytvořit nebo jinak získat modely. Modely vytvořené předem usnadní práci s objekty díky věcem jako pevný tvar, mesh a collidery. Všechny tyto prvky se hodně využívají v samotné simulaci. Další možnosti simulace je přímo vstřebávání složek krve do hepatocytů, kde se jedná spíše o tkáň, a není zapotřebí tak detailního modelu.

Obsah obrázku obloha, snímek obrazovky, kreslené

Popis byl vytvořen automaticky

7. Výběr prostředí

Některé nástroje se hodí pro jisté simulace více než jiné, a proto je vhodné vytvořit porovnání těchto nástrojů. Vhodně zvolený nástroj může silně urychlit celý proces a zlepšit výslednou simulaci jak na úrovni kódu, po vizuální stránce.

7.1 Nástroje pro 3D vývoj a programování

Pro porovnání jsem vybral 3 nástroje, ve které mají předpoklady pro takový projekt. Jedná se o Unity, Unreal engine a knihovnu Open GL. Všechny jsou známé a vysoce používané

7.1.1 Unity

Unity je silný a multifunkční nástroj sloužící především pro vývoj 2D i 3D her, animací i virtuální reality. Od svého začátku roku 2005 se Unity stala jedním z nejpopulárnějších nástrojů pro tvorbu her za což vděčí svému uživatelskému prostředí, spoustou funkcí a možností exportovat na spoustu platforem. Unity používá jazyk C#, který je dobře zdokumentovaný a popsaný.

7.1.2 Unreal engine

Unreal Engine, vytvořený společností Epic Games, je software primárně určený pro herní účely. Proslulý svou špičkovou grafikou a pohlcující herní atmosférou obsahuje pokročilé technologie jako Nanite a Lumen pro složité vykreslování a dynamické světelné efekty. Vizuální skriptovací systém Blueprints zjednodušuje vývoj herní logiky bez nutnosti kódování, i když uživatelé mají také možnost využít k programování C++. S výkonnými nástroji pro animaci a simulaci fyziky, jako je Control Rig a MetaHuman Creator, mohou uživatelé bez problémů vytvářet realistické postavy a prostředí. Unreal Engine je kompatibilní s různými platformami, včetně mobilních zařízení a systémů VR/AR, a nabízí nástroje pro optimalizaci výkonu.

7.1.3 Open Gl

OpenGL, také známé jako Open Graphics Library, bylo vytvořeno společností Silicon Graphics v roce 1992 jako průmyslové standardní grafické rozhraní API. Tento výkonný nástroj umožňuje vývojářům snadno navrhovat interaktivní 2D a 3D grafiku. OpenGL nabízí širokou škálu funkcí, jako jsou geometrické transformace, texturování, osvětlení a stínování, a stalo se oblíbenou volbou pro vykreslování grafiky napříč různými zařízeními a operačními systémy. Jeho všestrannost a efektivita z něj udělaly řešení pro širokou škálu aplikací, včetně vývoje her, CAD softwaru, simulací a vizualizace vědeckých dat.

7.2 Porovnání

Pro srovnání byli zvoleny následující kategorie:

1. Programovací jazyk
2. Komunita
3. Cena
4. Grafické schopnosti
5. Uživatelská přístupnost

Tyto Parametry budou ohodnoceny 1-3 dle toho, který nabízí nejlepší podporu pro danou kategorii. Nejvíce bodů znamená nejlepší. Jako jazyk je preferovaný C#. Je dobře zdokumentovaný a má velkou podporu. Velká komunita znamená, že je mnohem snadněji dostupná pomoc. Cena hraje roli v případě rozšiřování simulace, kdy pro lepší nástroje je někde třeba si připlatit. Grafické schopnosti velmi usnadňují práci v daném nástroji. Uživatelská přístupnost definuje snadný pohyb v prostředí.

7.3 Hodnocení

Pro programování v C# se nejlépe hodí Unity. Je to nativní jazyk tohoto nástroje. Oproti tomu Unreal engine je psaný v C++ nebo blueprintech, které se ani jeden nehodí pro budoucí možnou návaznost práce. Open Gl je jen api a má podporu C#, jen není tak rozšířená. Komunitu má nejrozšířenější Unity a Unreal, ale unity o něco více. Cenově je nejvýhodnější Open Gl, jelikož má všechny funkce dostupné zdarma. Následuje Unreal engine a nejhůře dopadlo Unity s více možnostmi licencí. Graficky je na tom pro 3 D práci nejlépe Unreal Engine. Open Gl chybí mnoho nadstavbových funkcí. Uživatelsky nejpříjemnější se jeví Unity. Lze se v něm snadno pohybovat a vše je po ruce. V Open GL by se téměř všechny nástroje musely dodělávat, či hledat doplňkové knihovny.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Unity | Unreal Engine | Open Gl |
| Jazyk | 3 | 1 | 2 |
| Komunita | 3 | 2 | 1 |
| Cena | 1 | 2 | 3 |
| Grafické schopnosti | 2 | 3 | 1 |
| Uživatelská přístupnost | 3 | 2 | 1 |
| Celkem bodů | 12 | 10 | 8 |

8. Praktická část

V této práci je použit již vytvořený model jater [3], který je dodán do práce. Model je již rozdělen na základní části. Nejdůležitější pro vykreslování jsou Pravý a levý jaterní lalok. Na tyto části budou vykreslovány změny, které se v játrech budou odehrávat, v závislosti na složení krve, které bude protékat žílami. Spolu s laloky jsou do modelu rovnou zavedeny žíly a tepny. Některé části původního modelu byly vypuštěny, z důvodu nepodstatnosti pro simulaci. Byly ale ponechány pro možnou budoucí návaznost práce. Jedná se například o žlučník.

8.1 Zpracování Modelů pro práci

Model je vhodné před samou simulací zpracovat, což usnadní budoucí manipulaci. Je možno více možností jak se tohoto úkonu zhostit, dle autorova záměru co v simulaci chce prezentovat a pro koho je určena.  
8.1.1 Triangulace

Nejpřímější metodou je triangulace objektu. Model se rozloží na menší Buňky= trojúhelníky, které reprezentují. Technicky se použije vlastnost mesh.vertices, která vrátí počet vrcholů meshe a uloží se do bufferu pro Vertexy. Dále se dají rozdělit na menší trojúhelníky dle potřeby míry detailu.Obsah obrázku snímek obrazovky

Popis byl vytvořen automaticky

void ParseMeshToTriangles()

{

MeshFilter meshFilter = GetComponent<MeshFilter>();

if (meshFilter == null)

{

Debug.LogError("MeshFilter not found!");

return;

}

Mesh mesh = meshFilter.mesh;

Vector3[] vertices = mesh.vertices;

int[] triangles = mesh.triangles;

for (int i = 0; i < triangles.Length; i += 3)

{

Vector3[] triangleVertices = new Vector3[3];

triangleVertices[0] = vertices[triangles[i]];

triangleVertices[1] = vertices[triangles[i + 1]];

triangleVertices[2] = vertices[triangles[i + 2]];

CreateTriangle(triangleVertices);

}

// Optionally, disable the original mesh renderer

GetComponent<MeshRenderer>().enabled = false;

}

void CreateTriangle(Vector3[] vertices)

{

GameObject triangle = new GameObject("Triangle");

triangle.transform.SetParent(transform);

Mesh mesh = new Mesh();

mesh.vertices = vertices;

mesh.triangles = new int[] { 0, 1, 2 };

mesh.RecalculateNormals();

MeshFilter meshFilter = triangle.AddComponent<MeshFilter>();

meshFilter.mesh = mesh;

MeshRenderer meshRenderer = triangle.AddComponent<MeshRenderer>();

meshRenderer.material = GetComponent<MeshRenderer>().material;

//meshRenderer.material = GetComponent<MeshRenderer>().material.gET

triangle.tag = "Hepatocyte";

triangle.AddComponent<MeshCollider>();

}

8.1.2 Generování objektů

8.1.3 Metaballs

Pohled na játra – celek, dle krve-

.1 Model jater

7.2 Krev

7. Zpracování modelů pro práci  
Model je třeba zpracovat před započetím práce. Pro tento účel je zvolena triangulace Objektu, kde se zvlášt triangulují levý a pravý lalok. Takto vytvořené trojúhelníky představují zvětšené budoucí hepatocyty, a bude se tedy na nich projevovat změněný stav jater. Pro větší detail je možno udělat podrozdělení – subdivizi, která rozdělí vytvořené trojúhelníky na menší.

Vytyčení měřících metrik

V grafice se obecně perou mezi sebou pojmy Míra detailu a Výpočetní náročnost.

Seznam Použité literatury  
[1] Michał Naskręt Robust simulation of internal body organs <https://sano.science/research/robust-simulation-of-internal-body-organs/>

[2]Pan, J., Yan, S., Qin, H. *et al.* Real-time dissection of organs via hybrid coupling of geometric metaballs and physics-centric mesh-free method. *Vis Comput* **34**, 105–116 (2018). <https://doi.org/10.1007/s00371-016-1317-x>

[3] <https://3dexport.com/3dmodel-liver-anatomy-479483.htm>

[4] <https://theses.cz/id/h5yupe/STAG86730.pdf>

Obrázky

[1] <https://www.wikiskripta.eu/w/Játra>