

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

Listopad 2024

Prohlášení:

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

V Hradci Králové dne 28.10.2024

Lukáš Hromek

Poděkování:

Děkuji vedoucí bakalářské práce doc. Ing. Haně Tomáškové, Ph.D. za vedení práce a hodnotné tipy, rady a při zpracování této práce. Dále bych rád poděkoval Ing. Brunu Ježkovi Ph.D. a Bc. Dominiku Kolouchovi za rady a konzultace při zpracování praktické části práce.

Slovník pojmů

Organismus – člověk, zvíře, rostlina, či jiná forma života

Živiny – látky potřebné pro výživu organismu

Hormon – přenašeč informací mezi tkáněmi

Krev – tekutina v těle. Roznáší živiny, hormony...etc.

Sraženina – ztuhnuté kousky krve

Žíla – trubice rozvádějící krev, vede do srdce

Tepna – trubice rozvádějící krev, vede od srdce

Céva – menší trubice, rozvádějí krev

Metabolismus – přeměna látek v těle, většinou na energii

Imunitní systém – chrání organismus před infekcemi

Zánět – obranná reakce imunitního systému

Fibróza – zvětšení koncentrace vazivové(pojivé) tkáně, většinou po zánětu.

Amino-kyseliny – organické sloučeniny, stavební jednotka proteinů

Protein – základní kámen buněk, skládá se z amino-kyselin

Vitamín – organická sloučenina, má rozmanité funkce dle typu

Tuk – zásobárna energie. Potřebné pro správné fungování organismů

Sacharid – živina

Cukry – zdroje energie

Glykogen – zpracovaný, uložený cukr v těle

Glukóza – zdroj energie pro buňky

Kyslík – část vzduchu, vstřebáván plicemi, klíčový pro tvorbu energie

Oxid uhličitý – odpad po zpracování kyslíku

Alkohol – toxická látka, populární pro své účinky

Sval – specializovaná tkáň

Cholesterol – látka v těle, pomáhá zpracovávat tuky

Nádor – skupina buněk, která se oddělila od organismu a parazituje na ostatních

Epitel – výstelka dutin

Endotel – výstelka krevních cest

Enzym – molekula urychlující reakce v organismu

Sinusoida – drobná žíla v játrech

Biomateriál – přírodní nebo umělý materiál, který interaguje s biologickými systémy

Hydrogel – látka která má polo-tekutý, polo-pevný stav a jejíž základem je voda

Fenotyp – soubor pozorovatelných vlastností a znaků organismu

Simulace – napodobování skutečnosti

Logický obvod – provádí výpočet dle struktury

Čip – destička s logickým obvodem

2D – dvourozměrný

3D – trojrozměrný

Vektor – prvek mající směr a velikost

Normála – kolmý vektor ku ploše

Mesh – kolekce vrcholů a hran tvořících 3 D objekt

Tag – vlastnost objektu v unity, pomáhá shlukovat objekty

Iterace – přidání k hodnotě 1

Engine – nástroj, stroj

Renderer – vykreslovací nástroj

Headset – komunikační aparát na hlavě

Mezipaměť – dočasné úložiště dat

Gradient – směr růstu, v simulaci měří změny

GPU – grafická karta

CPU – procesor

Anotace:

Bakalářská práce se zabývá simulací činností jater. Cílem je prozkoumat funkce jater, jejich části a chování. Dále se zaměří na nejlepší možnosti simulace orgánů a tkání v počítačovém prostředí. Práce se bude zabývat 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 aim is to explore the functions of the liver, its parts and behaviour. Furthermore, it will focus on the best possibilities for simulating organs and tissues in a computer environment. The work will deal with the technical part of how to create an organ simulation. The biochemical correctness of the marches will not be addressed so much.

1 Úvod	10
2 Cíl práce	11
3 Metodika.....	12
4 Orgány	13
4.1 Játra	13
4.2 Stavba Jater	14
4.2.1 Hepatocyty	14
4.2.2 Cholangiocyty	15
4.2.3 Další buňky	15
4.2 Krev	16
4.3 Jaterní Funkce.....	17
5 Počítačové Simulace	18
5.1 Historie a vývoj počítačových simulací	18
5.2 Typy simulací	18
5.2.1 Simulace založené na rovnicích	19
5.2.2 Agentní simulace	20
5.2.3 Víceúrovňové simulace.....	20
5.2.4 Monte Carlo simulace.....	21
5.3 Vizualizace simulací	21
5.3.1 2D vizualizace	22
5.3.2 3D vizualizace	22
5.3.3 Částicové systémy.....	23
5.3.4 VR a AR	24
5.4 Optimalizace	24
5.4.1 Optimalizace simulací.....	25
1 Hledání gradientu	25
2 Stochastická optimalizace.....	25
3 Metoda responsivního povrchu.....	25
4 Heuristické metody.....	25
5 Statistické metody	26
5.4.2 Optimalizace vizualizací	27
1 Level of Detail	27
2 GPU Akcelerace	27
3 Cachování a paměťová optimalizace	28
4 Metakoule a organické modely	29
6 Počítačové simulace v biologii	30

6.1 Buněčné simulace.....	30
6.2 Tkáňové simulace	31
6.3 Orgánové simulace	31
6.4 Nástroje	32
6.4.1CompuCell3D.....	32
6.4.2PhysiCell.....	33
6.4.3Morpheus	33
6.4.4Chaste.....	33
6.5 State-of-Art.....	34
6.5.1 Organ-on-chip.....	34
6.5.2 Organoid	35
6.5.3 3D Biotisk.....	35
7 Výběr prostředí pro simulaci	36
7.1.1 Unity	36
7.1.2 Unreal engine 5	37
7.1.3 OpenGL.....	37
7.2 Porovnání.....	37
7.3 Hodnocení	38
9 Tvorba simulace v Unity	39
9.1 Prvky pro práci.....	39
9.1.1 GameObject.....	39
9.1.2 Kamera.....	39
9.2 Datové struktury.....	39
9.2.1 Liver	39
9.2.2 Clustery	40
9.2.3 CellTriangle	40
9.2.4 BloodCell.....	41
9.3 Model jater	41
9.3.1 Zpracování modelu pro práci.....	42
9.4 Nedokonalosti modelu	43
9.4.1 Propletenost krevních cest	43
9.4.1 Nespojené krevní cesty.....	44
9.5 Průběh simulace	44
9.5.1 Pohyb krve	45
9.5.2 Předání vlastností	46
9.5.3 Sledování změn.....	46

9.6 UI a Ovládání	46
10 Shrnutí výsledků	48
10.1 Budoucí rozšíření	48
11 Závěr	49
Seznam Použité literatury.....	50
Seznam obrázků	54
Seznam ukázek kódu.....	54
Použité modely.....	54

1 Úvod

Znalost lidského těla pomáhá činit zdravá rozhodnutí a přijmout vhodná opatření, když se objeví příznaky problému. Při pokusech o přiblížení pochodů lidského těla je občas náročné najít hranici mezi detailní simulací chování buněk a obecnou simulací, která pomůže divákovi obecně lépe pochopit chování orgánu. Simulace buněčných struktur a funkcí bývají většinou náročné a zaměřeny pro publikum, které se už v daném tématu má aspoň základní znalosti. Je ale důležité poskytnout vhled do fungování složitých systémů, který je přehledný i pro nezainteresovanou osobu. [1][2]

V první části, práce seznámí s orgány a proč je důležité se jimi zabývat. Následně se zaměří na bližší pochopení jater, jejich složek a funkcí.

Druhá část přibližuje, co jsou to počítačové simulace, k čemu a proč se používají. Dále popíše typy simulací, jejich fungování a využití. Také jsou zde uvedeny možné zobrazení v počítačovém prostředí. Uvede i využití počítačových simulací v biologii a State-of-Art.

Třetí část práce se zabývá porovnáním nejznámějších prostředí, která jsou vhodná pro 3D simulace. Jsou to Unity, Unreal Engine 5, a Open GL. Proběhne zde seznámení s prostředím, vytyčením kritérií hodnocení a vyhodnocení.

Poslední část dokumentuje vytváření prostředí a zpracování modelů pro simulaci. Také jsou zde uvedeny části kódu při vytváření simulace.

2 Cíl práce

Cíl práce je přiblížit proces a metody vytváření simulací jater a jiných orgánů či organismů. Pojednává o důležitosti jater, jejich funkcích a složení. Dále se zabývá možnostmi počítačových simulací, jejich rozdělením, typy a využitím. Následně vybírá a porovná prostředí, které je vhodné pro tvoření 3D simulace. Součástí práce je i tvorba 3D simulace, která zahrnuje vytvoření modelů jater a krve, a naprogramováním jejich chování.

3 Metodika

Metodika bakalářská práce spočívá ve kvalitativním sběru informací o orgánech, především o játrech, a o počítačových simulacích organických pochodů, z článků, odborných textů a pojednání. Tyto informace jsou prezentovány v teoretické části pro vnesení čtenáře do problematiky.

Cílem je přiblížit čtenáři fungování jater a možnostmi jejich simulací.

Úkolem je na základě sesbíraných informací zpracovat simulaci jater.

Sesbírané informace by měly by odpovědět na následující otázky vyplívající ze zásad pro zpracování:

1. Co jsou to játra a jaké mají funkce?
2. Jaké simulace jsou vhodné pro simulování orgánů?

Tyto otázky jsou zpracovány teoretickou částí práce.

Umělá inteligence je používána k dohledávání relevantních zdrojů informací.

4 Orgány

Studium lidského zdraví pomáhá lépe pochopit prevenci a léčbu nemocí. V dnešní době vyčíslených lékařů se lidé musí začít spoléhat i na vlastní znalosti o zdraví, aby nezatěžovali lékaře banálními příznaky. Příznaky se nejčastěji projevují bolestí, vyrážkou či jinými vjemy které jsme schopni zpracovat smysly, a nejčastěji mají příčinu ve vnitřních orgánech. [3][4]

Orgány jsou důležitou součástí těla organismů, kde vykonávají specifické funkce nezbytné pro život. Každý orgán se skládá z tkání, které mají za úkol plnit jejich určitou biologickou funkci. V lidském těle se nacházejí různé druhy orgánů, které zodpovídají za širokou škálu činností, od dýchání, přes cirkulaci krve, trávení potravy až po vylučování odpadních látek. Orgány jsou obvykle sdružovány do orgánových systémů, jako je například trávicí, dýchací, oběhový, nervový či vylučovací systém. Každý systém plní komplexní biochemické funkce, kde jednotlivé orgány spolupracují, aby zajistily rovnováhu organismu. [5]

Tkáň je uskupení buněk s podobnou stavbou a společně plní jeden či více účelů. Prostory mezi buňkami vyplňuje materiál, nazývaný mezibuněčná matrice. Mezibuněčná matrice může obsahovat speciální látky které jsou jedinečné pro konkrétní tkáň a propůjčují této tkáni unikátní vlastnosti. [6]

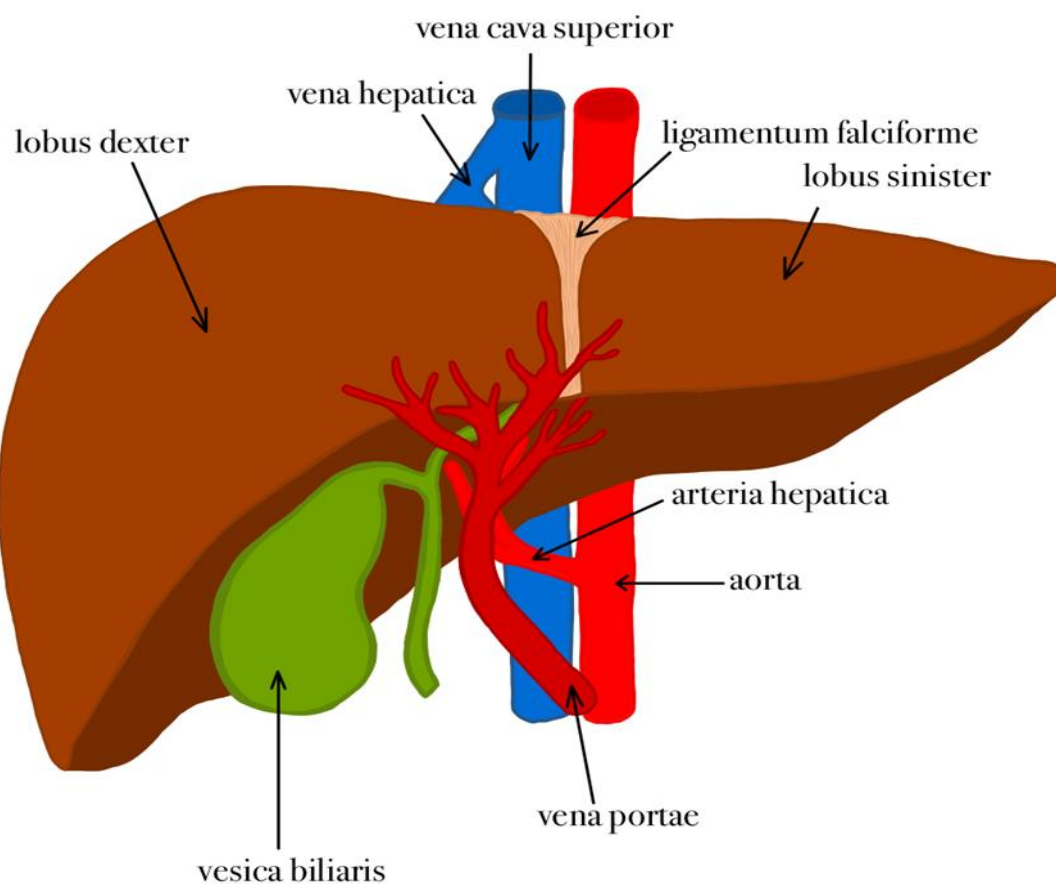
Buňka je základní strukturou všech živých organismů. Je schopna vykonávat všechny základní životní procesy a to sama, nebo v kooperaci s jinými buňkami. Každá buňka je složená z různých chemických látek, které umožňují její přežití, růst a reprodukci. Uvnitř buňky probíhá množství biochemických procesů, které zajišťují její funkce, a buňky tvoří tkáň a orgány v tělech komplexnějších organismů. [7]

4.1 Játra

Játra jsou jedním z nejvýznamnějších orgánů lidského těla. Jsou zodpovědná za celou řadu tělesných procesů a díky jejich funkci přicházejí do styku s látkami získanými z prostředí, a tak mohou poukazovat na spoustu různých problémů v organismu. Nacházejí se v pravé horní části břišní dutiny a mají mnoho metabolických, filtračních a sekrečních funkcí. Vstřebávají živiny z krve a produkují žluč, což je tekutina, která podporuje trávení tuků a zpracování bílkovin či sacharidů. Játra také ukládají látky jako glykogen a vitamíny, které ovlivňují srážlivost krve, a odstraňují z krve toxiny a staré červené krvinky, které mohou být vyloučeny močí nebo stolicí. Slouží také jako úložiště pro energii ve formě glykogenu, který se v případě potřeby přeměňuje na glukózu a uvolňuje do krve. [8][9]

4.2 Stavba Jater

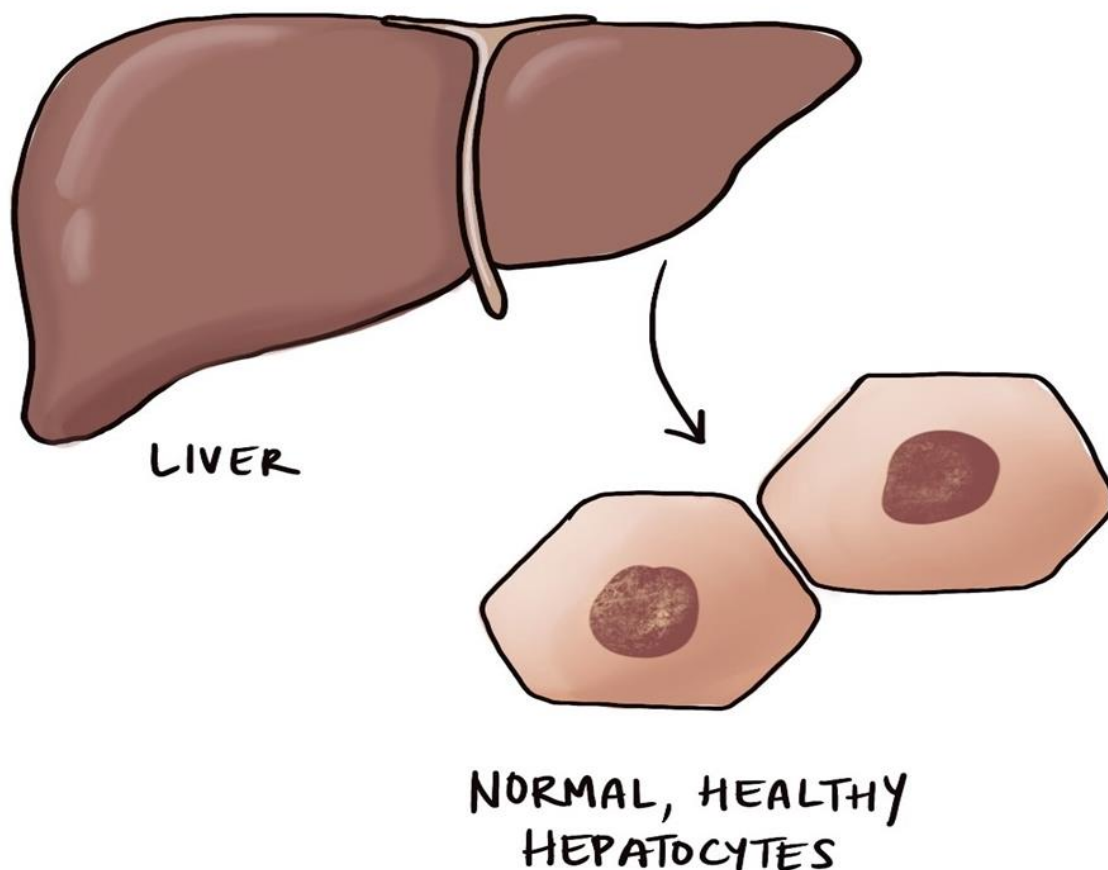
Játra jsou rozdělena na dva hlavní laloky – pravý a levý, označované jako *lobus dexter* a *lobus sinister*. Zásobení okysličenou krví ze srdce je zajištěno tepnou *arteria hepatica*. Žíla *vena portae* zase do jater přivádí krev bohatou na živiny ze střev. O odvádění krve z jater zase stará žíla *vena hepatica*, která směřuje do dolní duté žíly (*vena cava superior*), což umožňuje distribuci živin po těle. Kromě toho játra produkují žluč, která je odváděna do žlučníku (*vesica biliaris*) kde je skladována. [10]



Obrázek 1 Stavba jater, převzato [1]

4.2.1 Hepatocyty

Hepatocyty představují hlavní složku jater. Tvoří až 80 % objemu všech jaterních buněk a plní jejich hlavní symbolické funkce. Jsou také zodpovědné za filtrování lipidů, růstových hormonů a dalších krevních molekul. Mají schopnost schopnosti pohltnout velké množství toxinů a následně se z nich zregenerovat. V této funkci jsou hepatocyty náchylné k poškození zejména při nadměrném vystavení škodlivým látkám, jako jsou tuky či alkohol. Také jsou velmi citlivá na napadení viry. [11]



Obrázek 2 Stavba Jater Hepatocyty, převzato

4.2.2 Cholangiocyty

Cholangiocyty jsou buňky epitelu, tvořící výstelku žlučových vodů a mají zásadní podíl na produkci a udržování rovnováhy žluči. Mají různé velikosti i funkce, na čemž závisí jejich schopnost transportu rozpuštěných látek ve žlučovém traktu. Starají se o regeneraci jater, pokud je regenerace hepatocytů negativně ovlivněna, například u trvalých jaterních onemocněních. [12]

4.2.3 Další buňky

V játrech se nachází i další důležité buňky.

Kupfferovy buňky, nacházející se ve stěnách jaterních sinusoid. Mají převážně imunitní a čistící funkci, kdy na sebe váží různé odpadní částice a bakterie. Spolupracují s dalšími jaterními buňkami a na udržování vnitřní rovnováhy tuků a spouští reakce na toxiny. Pokud jsou játra poškozena, přispívají opravě tkání, ale při velkém poškození mohou spouštět proces zvaný fibróza. [13]

Endotelové buňky mají vícero funkcí, které se mohou lišit v závislosti na jejich specifickém umístění v těle. Nachází se v epitelu krevních cest a díky své rozmanitosti, mohou efektivně udržovat průtok krve v systému. Udržují tekutost krve, vypouštěním látek, které regulují počet aktivních částic v krvi, což redukuje její srážlivost. Kromě toho jsou endotelové buňky rozhodujícím faktorem propustnosti látek skrz stěny krevních cév. Tato propustnost se může při poškození měnit. [14]

Itovy buňky jsou především zodpovědné za ukládání tuků a ukládání vitamínu A v nich rozpustném. Navíc pomáhají při různých jaterních zraněních. Itovy buňky se nacházejí v Dissově prostoru, což je místo mezi Endotelovými buňkami a hepatocyty. [15]

4.2 Krev

Krev je tekutina klíčová pro život. Doručuje kyslík a živiny do buněk a následně odvádí oxid uhličitý a další odpadní produkty. Většina složek se tvoří v kostních dutinách a následně je pumpována srdcem do plic pro načerpání kyslíku a zbavení se oxidu uhličitého. Živinami a dalšími látkami natrávené potravou je obohacena ve střevech, kde následně prochází játry, v nichž probíhá zpracování, detoxifikace a ukládání důležitých látek. Na krev se dá pohlížet i jako na tkáň, neboť se skládá z nejrůznějších buněk. Hlavní části krve se dají rozdělit na:

1. Plasma
2. Červené krvinky
3. Bílé krvinky
4. Krevní destičky

[16]

Plazma, je tvořena zhruba 90 % vodou a je syntetizována játry. Obsahuje proteiny, živiny (například glukózu a aminokyseliny), odpadní látky, lipidy a minerály. Plazma také roznáší hormony a enzymy klíčové pro výživu, imunitu a metabolismus. [16]

Červené krvinky, jsou zodpovědné za přenos kyslíku z plic do tělesných tkání díky obsahu proteinu hemoglobinu. Dožívají se kolem 120 dní a poté se rozpadají. [16]

Bílé krvinky hrají klíčovou funkci při obraně a obnově těla. Dělí se na více druhů, kde každý je vhodný pro specifický úkol (např. regenerace, likvidace specifického patogenu, etc.). [16]

Krevní destičky jsou nejmenší buňky v krvi, co vyrovnávají svým množstvím. Mají složitější metabolismus a strukturu než červené krvinky. Pomáhají při krvácení tím, že tvoří hmotu v místě poranění. Při nedostatku krevních destiček může dojít k prodlouženému krvácení. Krevní destičky mají v krevním řečišti životnost 10 dní. [16]

4.3 Jaterní Funkce

Do dnes bylo nalezeno více než 500 důležitých funkcí jater. Mezi nejdůležitější patří udržování rovnováhu většiny chemických hladin v krvi. Nejdůležitější funkce se dají shrnout do:

- Zpracování živin
- Detoxifikace
- Syntéza proteinů
- Produkce žluči

[17]

Játra hrají zásadní roli při zpracování živin. Transformují sacharidy na glukózu, dle potřeby vytváří/rozkládá lipidy. [18]

Toxiny jako například alkohol jsou třeba neutralizovat, aby nepoškodili vitální buňky v těle. I když malé procento se odvede z těla jinými způsoby, játra mají hlavní zodpovědnost za zpracování alkoholu. Účinnost a rychlost tohoto procesu silně závisí na zdraví jater a lécích. [19]

Proteiny jsou důležité pro správný růst, neboť je tělo využívá pro opravu a tvorbu nových buněk. Proteiny se rozloží na amino-kyseliny a ty jsou následně v játrech použity na tvorbu nových proteinů. Přebytké amino-kyseliny jsou posílány do svalů, kde pomáhají udržovat svaly zdravé. [18][20][21]

Žluč je tekutina, plnící mnoho různých funkcí. Je produkována hepatocyty, a následně modifikována cholangiocyty. Zodpovídá za regulaci tuků, a v nich rozpustných vitamínů. Neutralizuje cholesterol a bilirubin, což je látka vznikající rozložením červeného barviva krve. [22]

5 Počítačové Simulace

Počítačová simulace je virtuální model, který kopíruje nebo přibližuje systém reálného světa. Díky tomu že jsou zpracovány digitálně, poskytují větší prostor pro vytváření komplexních procesů, které by jinak byly moc složité na pochopení a orientaci. Tím vnáší do systému faktor náhody, který lépe odráží očekávání, jak se systémy budou chovat v podmínkách reálného světa, jako jsou změny v prostředí, například poptávka na trhu [23]

5.1 Historie a vývoj počítačových simulací

Historie počítačových simulací sahá do 40. let 20. století s ranými aplikacemi ve vojenském a vědeckém kontextu. Průkopníci jako John von Neumann vývojem počítače typu ENIAC poskytli významný pokrok v této oblasti. Během desetiletí se simulace vyvíjeli a nacházeli aplikace v různých oblastech, jako je strojírenství, biologie a ekonomie. Zavedení důmyslnějších algoritmů a výpočetního výkonu na konci 20. století dále zvýšilo jejich schopnosti, takže simulace se staly nezbytnými nástroji pro analýzu a rozhodování ve složitých systémech. [24]

Simulace se stali důležitou částí znalostní základny pro odborníky v nejrůznějších odvětvích. Pokrok ve vizualizaci a grafice revolučně změnili sektor simulací uživatelsky přívětivým modelováním za dostupnou cenu. V 90. letech se simulace se stala nezbytnou optimalizační metodou provozu závodu. Do roku 1998 se software vyvinul do stavu, který umožňoval modelovat komplexní systémy bez pokročilých znalostí programování. Charakteristiky moderního simulačního softwaru jsou uživatelsky přívětivé prostředí, integrací další systémů a rychlé testování a ověření. [25]

5.2 Typy simulací

Existuje vícero druhů simulací, které se dělí na 2 hlavní modely: deterministické modely a stochastické modely. Každý má charakteristické vlastnosti, dle kterých se řídí jejich účel. [26]

Deterministické modely se vyznačují předpovídáním událostí bez prvku náhody. Pro jejich relevanci jsou zapotřebí kompletní a ideálně rozsáhlá data pro přesné předpovídání výsledků. Deterministické modely mají hlavní výhodu v jejich jednoduchosti a snadném pochopení, za to však neberou v úvahu všechny aspekty a proměnné reálných systémů. [26]

Klíčové vlastnosti:

- Při stále stejných vstupních parametrech dávají vždy stejné výsledky.
- Jsou jasné vztahy mezi vstupy a výstupy.

- Bez manuálního zásahu nemohou změnit chování.
- Výsledek je pevně daný, nebývá uváděn jako pravděpodobnost.
- Deterministické modely vynikají především v úkolech, kde je potřeba preciznost a objektivita. Používají se především v matematice, ovládání senzorů či financích.
- Pevná logika poskytuje stabilitu a spolehlivost, když jsou pravidla jasná.

[26]

Stochastické modely pracují na předpovídání pravděpodobnosti různých výsledků za různých podmínek. Předpovídá výsledky pomocí faktoru náhody, což zajišťuje lepší simulaci reálných prostředí. Mají aplikace napříč průmyslovými i vědeckými odvětvími. [27]

Klíčové vlastnosti:

- Předpovídá pravděpodobnost výsledků za různých podmínek.
- Představuje data a předpovídá výsledky pomocí faktoru náhody.
- Lepší napodobení systémů reálného světa

[27]

Pochopením rozdílů mezi modely může významně přispět k pochopení fungování reálných systémů.

5.2.1 Simulace založené na rovnicích

Nejčastější využití “Equation-based” simulací je ve fyzikálních vědách a dalších oborech které jsou definovány matematickými rovnicemi. Tyto simulace mohou být buď částicové nebo založené na poli. [26]

Částicové simulace se dají definovat jako soubor těles a rovnic řídící jejich interakce. Pomocí částicových simulací se dají například simulovat formování galaxií, kde se počítá s gravitační silou mezi objekty.

Simulace založené na poli se zakládají na souboru rovnic, které deformují či řídí interakce ploch v závislosti na prostředí či čase. Příkladem užití je simulace tekutiny nebo bouře. [26]

5.2.2 Agentní simulace

Agentní simulace(agent-based) jsou počítačové simulace které fungují tak, že představují jednotlivé entity nebo agenty, které interagují v prostředí. Využívají se především ke studiu chování mezi lidmi, věcmi, místy a časem. Jsou to stochastické modely budované zdola nahoru, což znamená, že jednotlivým agentům jsou přiřazeny určité atributy. Těm je dále naprogramováno chování k interakci mezi jinými agenty či prostředím. Agentní simulace se liší od tradičních metod v tom, že umožňují zkoumání složitých systémů vykazujících závislost jednotlivců. Tento způsob se hodí na provádění experimentů, které jsou jinak náročné, nemožné nebo neetické provádět v reálném prostředí. Parametry dat (jako například rychlost množení buněk či šíření chorob) jsou často obtížné podložit v literatuře. [26][28]

Modely založené na agentech poskytují účinný nástroj, který je zvláště užitečný, když se předpokládá že se jednotliví agenti navzájem ovlivňují, nebo když experimenty v reálném světě nejsou vhodným řešením. Modely založené na agentech poskytují důležité poznatky simulováním systémů tím, že zkoumají důsledky různých akcí na větší systémy. [26][28]

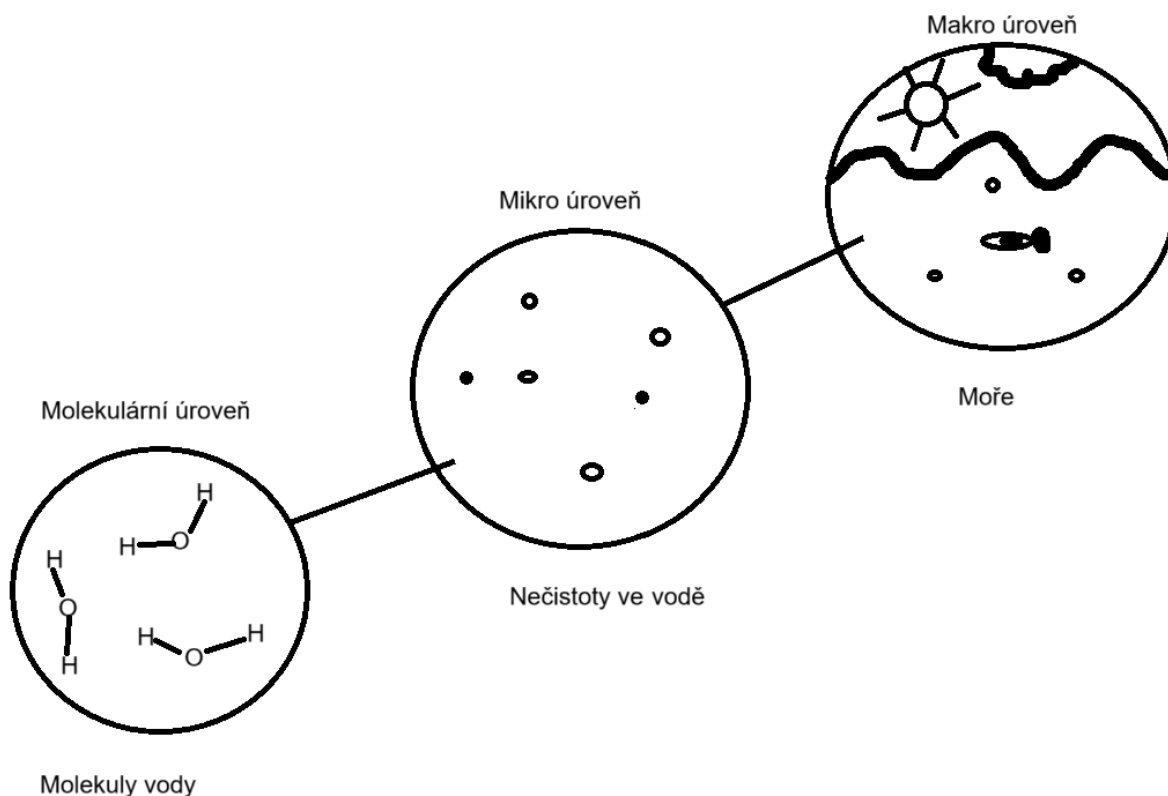
5.2.3 Víceúrovňové simulace

Víceúrovňové(multiscale) modelování spočívá v použití vícero modelů v různých měřítcích, což je většinou závislé na úrovni detailu. Tyto modely většinou kombinují poznatky z mikro a makro světa. Hlavní důvod pro použití víceúrovňové modelu nastává, když makro modely nemají dostatečnou přesnost či hloubku, a mikro modely jsou příliš složité, nepřehledné nebo neefektivní. Zapojením obou pohledů umožňuje propracovanější simulaci, která je přesná i účinná. [26][29]

Zahrnují tři klíčové prvky:

1. Analýza, která zkoumá vztahy mezi modely v různých měřítcích.
2. Modely, které spojují tato měřítká.
3. Algoritmy, které využívají tyto myšlenky ve výpočetních procedurách.

Tradiční metody modelování se obvykle zaměřují pouze na jedno měřítko, a často se spoléhají na zástupné reprezentace jevů na nižších úrovních, což může být problematické zejména u složitých systémů. [26][29]



Obrázek 3, Příklad úrovní víceúrovňové simulace

5.2.4 Monte Carlo simulace

Monte Carlo je algoritmus, ke generování výsledků používá jistý rozsah hodnot pro vstupy. Byla vyvinuta za druhé světové války Johnem von Neumannem a Stanislawem Ulamem. Název je odvozen od jeho zpracování prvku náhody (kasino). Monte Carlo se používají v oblastech jako umělá inteligence, ceny akcií a řízení projektů. Mezi její hlavní výhody patří zjištění citlivosti systému na změny. Tato metoda je vhodná pro přesné a dlouhodobé předpovědi, a generují vícero možných výsledků. Jednoduchá aplikace je například u výpočtu pravděpodobností hodu kostkou. [26][30]

Aplikace metody Monte Carlo spočívá v nastavení modelu, určením pravděpodobnosti pro proměnné a vstupy a následném opakováním spouštěním simulace. Měnící se parametry mohou ovlivnit výsledky a analýza rozptylu a směrodatné odchylky poskytuje pohled na rozložení dat. [30]

5.3 Vizualizace simulací

Vizualizační technologie umožňují zobrazovat data, která jsou získána simulací. Mezi běžné typy vizualizace patří 2D a 3D grafika, graf, animace či obrázek. Poskytnutím vygenerovaných dat ve snadno srozumitelném formátu i bez technický znalostí se dají

efektivně interpretovat a analyzovat data. Tím vizualizace pomáhá urychlit pracovní postupy a usnadňuje komunikaci. [31]

V této kapitole se více rozvedou vizualizační techniky a uvedou populární nástroje používané za tímto účelem.

5.3.1 2D vizualizace

Principy 2D vizualizace se zaměřují na jednoduchost a přehlednost. Tato metoda je výhodná pro prezentaci prostorových vztahů, usnadňuje rychlé pochopení systému a výsledů a podporuje srovnání mezi různými soubory dat. To je užitečné zejména ve vědeckých a technických kontextech, kde je kritická přesná reprezentace dat. 2D vizualizace bývají dostupnější a snadněji se vytvářejí než 3D modely. [32]

Výhody:

- Efektivita: 2D vizualizace považována za efektivní techniku pro modelování a tvorbu modelů. Je většinou jednodušší a rychlejší než práce ve 3D, a proto je stále některými odborníky upřednostňována.
- Časová náročnost: modelování ve 2D většinou zabírá méně času než ve 3D. Existují ovšem nástroje jako WITNESSVR, který umožňuje rychlé vytvoření modelu v 2D a následně ho automaticky převede do 3D.

[32]

5.3.2 3D vizualizace

Principy 3D vizualizace zlepšují reprezentaci dat tím, že poskytují hloubkový a prostorový kontext, což umožňuje lepší vhled do struktur a vztahů. Tato metoda je zvláště výhodná v oborech, jako je medicína, strojírenství a architektura, kde vizualizace komponent ve třech rozměrech podporuje lepší vhled a rozhodování.[32]

Výhody:

- Porozumění: 3D vizualizace jsou účinnější než 2D při prezentaci modelů a výsledků simulace. Nástroje jako FlexSim poskytují realistický náhled na data, což umožňuje větší efektivitu předávání informací.
- Model: Nabízí větší množství nástrojů pro rozvoj modelů, díky čemuž splňuje aktuální trendy. Vývoj ve 2D je sice rychlejší, ale 3D poskytuje širší možnosti.

- **Validace:** Díky realističtější vizualizaci, umožňuje snazší detekci chyb pro zvýšení kvality simulace.
- **Experimentování:** Většinou bývá preferována pro experimentální účely, protože poskytuje lepší výkon při analýze chování modelů a v „co kdyby“ analýzách.

[32]

5.3.3 Částicové systémy

Částicové systémy v simulacích jsou navrženy tak, aby modelovaly složité jevy zahrnující mnoho malých částic, jako jsou tekutiny, kouř nebo oheň. Částice je pouze 3D bod a v simulacích se pohybuje nepřímo, prostřednictvím externích virtuálních sil, jako je třeba vítr či gravitace. Principy částicových systémů zahrnují flexibilitu umožňující širokou škálu chování a interakcí spolu se snadnou manipulací pro vizuální efekty. Mezi výhody patří realistický pohyb a vzhled, efektivní manipulace s velkým množstvím částic a schopnost vytvářet dynamické, vyvíjející se scény. Tento přístup je zvláště účinný ve vizuálních efektech a animacích. [33]



Obrázek 4, Ukázka využití částicového systému v aplikaci sand:box

Částice jsou generovány emitery a ihned po vytvoření jsou na ně aplikovány podmínky prostředí. Tyto komponenty dohromady vytvářejí částicový systém, který lze

pozorovat v reálném čase. Úprava nastavení simulace umožňuje okamžitou zpětnou vazbu o změnách. [33]

5.3.4 VR a AR

Virtuální realita (VR – Virtual Reality) je nasimulované 3D prostředí, se kterým mohou uživatelé interagovat pomocí specializovaných pomůcek, což vytváří silně realistický zážitek.

Vylepšená realita (AR – Augmented Reality) překrývá prvky reálného světa digitální vrstvou díky čemuž vytvářejí směs digitálních a skutečných vjemů. [34]

Zásadní rozdíl mezi VR a AR je v jejich povaze. VR nabízí kompletně umělé prostředí vhodné pro hry a jinou zábavu. AR, oproti tomu vylepšuje pohled na reálný svět, což nachází využití především ve výuce či navigaci. Mezi hlavní typy těchto realit patří:

- imerzní VR – plně interaktivní 3D prostředí
- nepohlcující VR – založené na obrazovkách
- AR založená na značkách – používá kódy ke spouštění digitálního obsahu
- AR nezaložená na značkách – pomocí počítačového vidění rozpoznává objekty okolního světa

V posledních letech zažívají VR a AR významné pokroky, díky jejich zvyšující se dostupnosti a investicím společností do těchto technologií. Headsety VR si získaly popularitu díky jejich schopnosti ponořit diváka do virtuálního světa, zatímco AR je snadno dostupné na celé řadě chytrých telefonů a tabletů. Jejich využití ve vzdělávání nabízí možnost personalizovaného učení pomocí AI, chytré výukové systémy, které poskytují okamžitou zpětnou vazbu a analýzu dat pedagogů a zrychlení efektivitu administrativních úkonů. [34]

VR a AR mají velký potenciál zažehnout revoluci ve vzdělávání, vytvářením poutavých a interaktivních výukových materiálů, simulací a praktických lekcí. S neustálým vývojem těchto technologií, mají šanci modernizovat postupy v mnoha oblastech, jako je zábava či zdravotnictví. [34]

5.4 Optimalizace

Optimalizace je proces zdokonalování systémů a procesů do jejich nejlepší možné podoby. Je důležitá především v oborech jako strojírenství, finance a informatika. V informatice se optimalizace může vztahovat na obvody, software či celý počítač. Optimalizace softwaru se zabývá zefektivněním algoritmů, kódu a maximalizací využití hardwaru. Proces optimalizace

spočívá v identifikaci problému, vytvoření řešení a následnou implementací, například úpravou zdrojového kódu. [35]

Optimalizace má zvlášť důležitou pozici v aplikacích obsahujících vizualizaci (zvláště pak 3D vizualizaci) a grafické operace, například u videoher, leteckých simulátorů, virtuální reality, či grafickém modelování. Prostřednictvím optimalizace mohou tyto aplikace dosáhnout lepšího výkonu a snížit potřebné zdroje [35]

5.4.1 Optimalizace simulací

Optimalizace simulací jsou rozsáhlé pozornost přitahující téma. V této práci se pojednává o 5 hlavních technikách optimalizace. [36]

1 Hledání gradientu

Princip fungování hledání gradientu spočívá v tom, že změny vstupních hodnot ovlivňují výsledky simulace, aby pomohly optimalizovat model. K tomu používá následující metody:

- Metoda konečných rozdílů – porovnává změny na výstupech pro odhad gradientu
- Metoda pravděpodobnostních poměrů – spočítá gradient dle vstupních hodnot
- Perturbační analýza – odhaduje citlivost výsledků na vstupy jedním spuštěním simulace.
- Frekvenční analýza – mění vstupy a analyzuje výstupy.

[36]

2 Stochastická optimalizace

Stochastická optimalizace je založena na hledání nejlepšího možného řešení pro funkci, jejíž hodnoty nejsou známy přesně, ale lze je odhadnout nebo změřit. Funguje na principu neustálého opakování s odhady gradientu. [36]

3 Metoda responsivního povrchu

“Response Surface Methodology” (RSM) slouží k optimalizaci výstupu simulace pomocí řady regresních modelů, což jsou modely odhadující vztah mezi vstupy a výstupy. Oproti metodám založeným na hledání gradientu potřebuje RSM méně experimentů. [36]

4 Heuristické metody

Tyto metody fungují tak, že hledají přibližná řešení optimalizačních problémů pomocí různých strategií jako jsou

- Genetické algoritmy – inspirované biologickou evolucí, kombinují prvky náhody s výběrem nejlepších řešení. Jedinci v populaci jsou reprezentováni řetězci, které

vyjadřují různé hodnoty proměnných. Pomocí operátorů, jako výběr, křížení a mutace, GA generují nové „potomky,“ kteří dědí vlastnosti od „rodičů.“ Jednotlivci s vyšší „fitness“ (vhodnost pro prostředí) mají větší šanci na přežití a předání vlastností dál. Jsou rozsáhlé a efektivní hlavně pro komplexní problémy.

- Evoluční strategie – fungují na principu mutace, kdy starší generace (rodič) vytváří potomky drobnými změnami parametrů. Poskytuje účinný nástroj pro optimalizaci problémů se složitými parametry.
- Simulované žíhání – umožňují vyhnout se uváznutí v lokálních minimech, protože i když se postupně zkoumají méně příznivá řešení, je jim přidávána klesající pravděpodobnost. Tento přístup je zvláště účinný pro kombinatorické problémy a je vhodný při hledání globálních optimálních řešení.
- Hledání tabu – zabráňuje uváznutí v lokálních minimech tím, že udržuje "Tabu seznam" dříve prozkoumaných pohybů, čímž zabráňuje v návratu, k již projitým řešením. Vyniká schopností nalézt kvalitní řešení s rozumnými výpočetními nároky. Pomáhá překonat omezení jiných metod, které se mohou zaseknout v lokálních optimech.
- Metoda Nelder-Mead – začíná s body v simplexové oblasti, což jsou možné hodnoty řešení) a postupně vybere nejhorší bod, vyloučí ho a nahradí novým bodem odraženým od středu zbylých bodů. Tato metoda má omezení, zejména předpoklad o konvexní oblasti a problémy s implementací omezení.
- Asynchronní tým (A-tým) - kombinace různých strategií.

[36]

5 Statistické metody

Dělí se na:

- Vzorkování důležitosti – princip je změnit pravděpodobnostní rozdělení během simulace tak, aby se zvýšila pravděpodobnost výskytu vzácné události. Hlavním účelem je najít vhodnou změnu pro vyvolání vzácné události. Výsledky jsou následně upraveny korekčním faktorem, aby odhady zůstaly nestranné.
- Metody hodnocení a výběru – využívají techniky, jako je hodnocení podobných výsledků jako jednoho.
- Metoda vícero porovnání s nejlepším – vybírá nejlepší alternativu z možností která může zahrnovat i další měřítka (podmnožiny).

[36]

5.4.2 Optimalizace vizualizací

U modelu auta, výrobce potřebuje znát přesnou velikost a umístění každého šroubu v sestavě, což znamená vysoké množství detailů. Stejná míra detailu auta ve videohře, je po hráče zbytečná, a znamená pouze zátěž na zařízení. Optimalizace je nezbytná k zajištění plynulého zobrazení v reálném čase. Existuje 6 hlavních důvodů pro optimalizace modelů:

1. Výkon – optimalizace plynulého chodu zejména v aplikacích s vysokou snímkovou frekvencí, jako například VR.
2. Odezva – aplikace v reálném čase vyžadují rychlou reakci na uživatelské vstupy a změny prostředí.
3. Efektivita paměti – menší nároky na paměť, což je důležité zejména pro kompatibilitu mezi různými zařízeními.
4. Šířka pásma a úložiště – snížení požadavků na šířku pásma a úložiště.
5. Kompatibilita – konzistentní zobrazení na různých zařízeních.
6. Škálovatelnost – optimalizace dat v aplikacích umožňují vytvářet větší a složitější scény při zachování výkonu.

[37]

1 Level of Detail

Dosažení rovnováhy mezi realistickým zobrazením a výkonem v 3D scénách je neustálou výzvou při práci v 3D prostoru. Řešení nabízí "Level of Detail" (LOD), což je technika pro vytvoření více verzí 3D modelu. Úroveň detailů (LOD) se používá k zobrazení různých verzí 3D modelu na základě vzdálenosti mezi modelem a kamerou. Jak se objekty vzdalují, detaily se stávají méně patrnými. [38]

Toho se dosáhne vytvořením více verzí stejného 3D modelu s různou úrovní detailů. Tyto verze (LOD), sahají od modelů s vysokým rozlišením po zjednodušené verze s menším počtem polygonů. [38]

2 GPU Akcelerace

Akcelerace GPU neboli akcelerace grafického procesoru je výpočetní technika, která zprostředkovává výkon grafických karet k výraznému zvýšení výkonu aplikací. Používá možnosti paralelního zpracování GPU což znamená, že může zpracovávat více úloh současně.[39]

CPU oproti GPU vynikají ve zpracování postupných úloh a obecných výpočtů, zatímco GPU jsou stavěny pro zpracování více úloh naráz, což znamená, že hlavní využití mají u úloh, které vyžadují paralelní provádění. GPU jsou význačné architekturou s tisíci jádry, zatímco CPU jich mají méně. [39]

CUDA (Compute Unified Device Architecture) a CuDnn (CUDA Deep Neural Network) jsou architektury společnosti NVIDIA pro výpočty GPU a úkoly hlubokého učení. Akcelerace GPU funguje tak, že se jsou jisté výpočetní úlohy předávány z CPU na GPU. Úkoly, které lze paralelizovat, jsou zadány GPU, které využívá svých více jader pro simultánní zpracování. Tato funkce paralelního zpracování je výhodná především pro vykreslování grafiky, vědecké simulace a umělá inteligence [39]

Výzvy v tomto oboru představuje programování GPU, problémy s kompatibilitou a vysoká spotřeba energie. [39]

3 Cachování a paměťová optimalizace

Jedná se o ukládání do mezipaměti. To je proces, kdy jsou data uložena na dočasné umístění pouze po určitou dobu. Má pozitivní vliv na výkon všech typů aplikací. Data jsou uložena v hardwaru s rychlým přístupem, jako je RAM (paměť s náhodným přístupem). Populární mezipaměti jsou například Memcached a Redis. [40]

Výhody:

- Ukládání do mezipaměti umožňuje vývojářům značně zlepšit výkon.
- Dočasně ukládá data pro softwarové aplikace, webové prohlížeče či servery, což zajišťuje, že uživatelé nemusí stahovat informace pokaždé, když vstoupí na web nebo aplikaci.
- Datový model v mezipaměti může být odlišný od datového modelu v databázi. To znamená, že generovaná odpověď v důsledku více dotazů může být uložena proti jednomu identifikátoru.
- Rychlosti CPU jsou poměrně vysoké ve srovnání s přístupovou dobou hlavní paměti. Výkon procesoru je tedy omezen pomalou rychlostí hlavní paměti.

[40]

4 Metakoule a organické modely

Metakoule a organické modely jsou techniky generování 3D modelů. Metakoule jsou speciální geometrické tvary, které se podobají tekutinám a spojují se do hladkých ploch. Díky tomuto spojování jsou používány v simulacích kapalin a v 3D modelech orgánů.[41]

Spočívá v definici návrhového prostoru a aplikací analýzy konečných prvků (FEM – final element method), která rozdělí objekt na menší prvky. Poté se používá genetický algoritmus, který optimalizuje strukturu podle mechanických omezení a požadované hustoty. Výsledné metakoule tvoří spojitou síť buněk. [41]

6 Počítačové simulace v biologii

Počítačové simulace v biologii se používají pro zkoumání komplexních systémů a interakcí mezi molekulami. Pomáhají vědcům modelovat dynamické procesy jako například skládání proteinů, což je složitý proces, při kterém se řetězce aminokyselin skládají do 3D struktur. Simulace umožňují detailně prozkoumat procesy na molekulární úrovni bez potřeby fyzických experimentů, které by byly časově, finančně nebo technicky náročné. [42]

Tvorba realistického počítačového modelu spočívá ve výběru správných rovnic pravidel pro popsání dané vlastnosti a simulaci zkoumaného jevu. To může zahrnovat Newtonovy pohybové zákony pro výpočet drah částic nebo výpočty pro modelování interakcí mezi atomy a molekulami. V případě proteinů jsou často použity jednodušší modely, kde je protein reprezentován jako „náhrdelník“ tvořený „korálky“. Každý korálek zastupuje skupinu atomů s různými vlastnostmi a umožňuje sledovat skládání molekuly. [42]

Dají se rozdělit do několika kategorií, dle rozměru:

- Buněčné
- Tkáňové
- Orgánové

[43]

6.1 Buněčné simulace

Buněčné simulace se dají rozdělit na 2 druhy dle zkoumaného jevu, na strukturální simulace a simulace interakcí.

Strukturální simulace buněk jsou nástrojem pro modelování biologických procesů, které zatím stále nejsou plně prozkoumány. Využívají různé techniky, které umožňují modelovat chemické reakce a molekulární dynamiku. Mezi hlavní techniky patří:

- ODE modely – vycházejí z kinetických rovnic, které popisují změnu koncentrace látek v čase. Používají se pro simulace buněčných cyklů.
- Prostorové modely – přidávají dimenzi prostoru, se vyplatí u simulací dynamiky a umístění molekul v buňkách. Hodí se pro pozorování šíření signálů v buňkách
- Prostorově-Stochastické přístupy – kombinace stochastických a prostorových přístupů. Používá se pro simulace složitých buněčných procesů, interakce molekul na buněčných membránách.

[44]

Simulace interakcí mezi buňkami se používá pro pochopení jevů, jako je formování tkání a vzorců v mnohobuněčných systémech. Nejčastěji používají Agentní modely simulování, díky jejich schopnosti zpracování buněk jako samostatných jednotek, které spolu interagují. Využívané metody:

- ODE – popisují chování pomocí rovnic.
- Celulární automaty (CA) - simulují vztahy mezi agenty v mřížce na základě jejich sousedů.
- Cellular Potts model (CPM) - řídí chování pomocí změny energetický stavů.

[45]

6.2 Tkáňové simulace

Simulace tkání jsou důležité nástroje pro pochopení víceúrovňových interakcí mezi biochemickými sítěmi, mikroprostředími, buňkami, tkáněmi a orgány. Pochopení chování buněčných struktur v prostoru a čase je klíčové pro celou řadu oborů v medicíně a biologii.[46]

V minulosti řešily tkáňové modely problémům s poměrně úzký zaměřením modelovacích nástrojů. Většina těchto nástrojů se zaměřuje buď na tkáňovou mechaniku (fyzikální vlivy), nebo na interakce mezi buňkami. Také mají strmou křivku učení a předpokládají, že uživatel investuje čas potřebný k tomu, aby se naučil vnitřní části simulačního enginu. [46]

6.3 Orgánové simulace

Modely orgánů a simulace s nimi prováděné mají mnoho výhod. Virtuální modely orgánů mohou být užitečné mnoha způsoby. Schopnost manipulace se strukturou poskytuje zásadní zlepšení efektivity učení. Jedním nejmodernějších způsobů je vytvoření prostředí reálného scénáře, které pak může být zpracováno umělou inteligencí. Zejména chirurgické zákroky, které nejsou možné vytvořit v reálném prostředí, umožňují být zpracovány algoritmem, by mohl pomocí miliónů akcí prozkoumat všechny možnosti operace. [47][48]

6.4 Nástroje

Pro simulace buněk, tkání a orgánů jsou vytvořeny specializované nástroje. Tyto nástroje poskytují paletu funkcí, které umožňují vizualizovat data a následně s nimi manipulovat.

Nejpopulárnější jsou 4 open source nástroje

1. CompuCell3D
2. PhysiCell
3. Morpheus
4. Chaste

Tyto nástroje jsou používány mezi odborníky pro prezentaci dat. Kvůli svému velmi úzkému zaměření jsou ale náročné na práci a pochopení, a je vyžadována časová investice pro jejich efektivní využití. [46]

6.4.1 CompuCell3D

CompuCell3D je vyvinutý pro simulace buněčných a tkáňových interakcí. Umožňuje výzkumníkům modelovat komplexní biologické procesy, jako je tvorba tkání či vývoj onemocnění. Tento nástroj, založený na Monte Carlo metodě, poskytuje efektivní způsob simulace chování buněk a tkání. CompuCell3D je hojně využíván ve výzkumu pro studium vývoje nádorů či procesu hojení ran. [48]

Výhody a funkce CompuCell3D:

- Efektivita – při použití odborníky umožňuje rychlé vytvoření a úpravy komplexních biologických modelů. Využívá skripty v jazyce Python pro rozšíření funkcionality a flexibility.
- Modularita – využívá modulární přístup, kde jednotlivé moduly umožňují efektivní monitorování dějů v rámci simulace. Tento přístup umožňuje výpočetní optimalizaci a možnost rychlého přizpůsobení a škálování pro robustní simulace.
- Paralelní výpočty – podporuje souběžné výpočty na více jádrových procesorech.
- Zaměření – díky úzké specializaci, obsahuje funkce pro růst a dělení buněk. Také umožňuje modelování chemických polí a biochemických sítí, což přidává rozměr času a prostoru.
- Propojení s externími nástroji – Nabízí grafický editor pro programování simulací a nástroj CellDraw, který umožňuje prvotní nastavení buněk.

CompuCell3D představuje významný nástroj pro výzkumníky, kteří chtějí zkoumat komplexní buněčné a tkáňové dynamiky v simulovaném prostředí. [48]

6.4.2 PhysiCell

PhysiCell je nástroj pro simulování buněk založen na fyzice. Nabízí celé množství funkcí, usnadňujících simulace komplexních buněčných systémů.

Mezi hlavní výhody a funkce patří:

- Biotransport – PhysiCell je propojen s BioFVM, výkonným řešičem biotransportu, což je proces přenosu živin a jiných látek mezi biologickými systémy. Díky tomu lze sledovat a modelovat, jak různé chemické látky ovlivňují chování buněk
- Dynamika fenotypů – buňky mohou upravovat svůj fenotyp, což umožňuje realistické modelování adaptivního buněčného chování
- Paralelní výpočty – poskytuje optimalizaci pomocí OpenMP, což umožňuje efektivní souběžné výpočty. Díky tomu lze simulace snadno škálovat na milióny buněk na dostatečně výkonných zařízeních
- Nezávislost – je napsán v jazyce C++, který zajišťuje snadnou údržbu a přenositelnost mezi platformami

Tyto vlastnosti činí PhysiCell účinným nástrojem pro simulaci komplexních biologických systémů na buněčné úrovni. [49]

6.4.3 Morpheus

Morpheus je simulační prostředí pro studium víceúrovňových buněčných systémů.

Klíčové vlastnosti jsou:

- Víceúrovňová simulace – simuluje víceúrovňové mnohobuněčné systémy spojující ve 2D nebo 3D
- Deklarativní modelování – vytváří modely přímo v GUI, namísto programování
- Integrace dat – nabízí sadu FitMultiCell pro efektivní odhad parametrů z experimentálních dat

[50]

6.4.4 Chaste

Chaste (Cancer, Heart And Soft Tissue Environment) je simulační balíček pro zpracování matematických modelů fyziologie a biologie. Vývoj je veden primárně pro oblasti srdeční

elektrofyzologie, modelování tkání na základě buněk a ventilace v plicích. Srdeční elektrofyziologie (Cardiac Chaste), umožňuje simulace činností srdce. Srdeční elektrofyziologie (Cell-based Chaste) umožňuje využití různých modelů, jako je celulární automat či Cellular Potts model mají využití v aplikacích v biologii. Ventilace v plicích (Lung Chaste) spočívá v simulaci dýchacích cest v plicích. [51]

Výhody používání Chaste zahrnují:

- Flexibilita – využití vícero modelovacích rámců, umožňuje vědcům vybrat nejvhodnější nástroj pro výzkum a porovnat různé přístupy modelování.
- Jednoduchá instalace – použití Dockeru zajišťuje snadné běh na různých operačních systémech.
- Podpora HPC – optimalizace pro vysoko výkonnostní prostředí.
- Dokumentace – obsahuje rozsáhlou dokumentaci a tutoriály pro uživatele.

Díky těmto vlastnostem, představuje Chaste efektivní nástroj, který je hojně využíván v biologii a medicíně. [51]

6.5 State-of-Art

I když byly v minulých letech počítačové simulace velmi populární nástroj pro napodobování chování buněk a prostředí, v dnešní době se propracovanější metody výzkumu, které nezůstávají pouze v digitálním prostředí. Pomocí moderních technologií se dají vyrobit různé modely, které dokážou simulovat pochody buněk věrněji než čistě počítačové modely. State-of-art se dělí na 3 nejpopulárnější možnosti simulace orgánů. Jsou to:

1. Organ-on-chip
2. Organoid
3. 3D Biotisk

Tyto technologie se spolu dají propojit, aby tak vznikla ještě detailnější a propracovanější simulace prostředí.

[52]

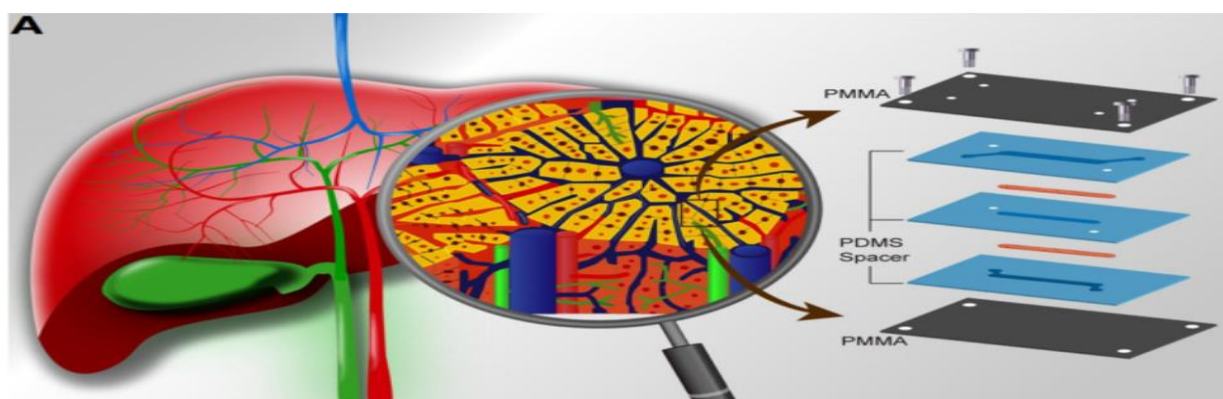
6.5.1 Organ-on-chip

Organ-on-chip technologie se zakládá na mikrofluidních čípech (Lab-on-a-chip).

Tyto čipy se skládají z mikrokanálek, které jsou vyryty či vlisovány do různých materiálů (sklo, silikon ...) a mají velikost okolo 10-100 μm . Mikrokanálky formující

mikrofluidní čip jsou navzájem propojené, aby dosáhly požadovaného výsledku. Díky své schopnosti přesně kontrolovat fyziko-chemické reakce tekutin, si našly své využití v chemii, mikrobiologii a medicíně. [53]

Díky těmto vlastnostem umožňují mimikování struktury a funkce lidských orgánů, kde dokáže kontrolovat růst a vlastnosti buněk. Dovoluje také real-time monitorování a analýzu změn na buněčné úrovni, což pomáhá k výzkumu reakcí na léky a jiné látky, či šíření nemocí. [52]



Obrázek 4 Microfluidic liver model, převzato

6.5.2 Organoid

Organoid je zjednodušená 3D verze orgánu, která strukturou připomíná anatomii orgánu či tkáně. Vytváří se z kmenových buněk, což jsou buňky, které se mohou přeměnit do jakékoliv jiné buňky. Díky tomu dokážou vytvořit téměř jakýkoliv požadovaná orgán. Mají mnoho výhod, jako například dobrou obnovu po skladování v mrazu, dlouhodobé kultivace, vysokou míru úspěšnosti, a relativně jednoduchého udržování organismu. Jsou neocenitelné při pozorování nejrůznějších virových onemocnění a při pozorování nádorů. [52][54]

6.5.3 3D Biotisk

3D biotisk je výrobní technologie, která se opírá o pokročilé biomateriály a různé tiskové mechanismy. Umožňuje přesné umístění buněk do 3D prostoru, obvykle hydrogelu, a tak věrně vytváří interakce mezi buňkami či mikroprostředím. Hydrogel může časem degradovat a buňky se postupně rostou a vyvinou do tkáně či orgánu, což má velký potenciál pro transplantace orgánů. Díky jejich prostorovému rozložení, jsou vhodné pro testování léků, díky jejich vyšší citlivosti na tyto látky. [52][55]

Praktická část

7 Výběr prostředí pro simulaci

Na internetu jsou dostupné různé aplikace poskytující prostředí a nástroje určené pro vytváření softwaru a práci s modely. Tyto aplikace usnadňují práci a tvoření kódu, neboť vytvářejí abstraktní vrstvu nad programovacím jazykem. Umožňují vývojářům soustředit se pouze na psaní kódu. Populární prostředí by měly poskytovat následující výhody: [56]

- Graficky renderovací engine, který podporuje 2D a 3D grafiku.
- Fyzikální engine podporující detekce kolizí a gravitace.
- Možnost více vláknového zpracování.
- Správu paměti.

Pro práci je vybráno prostředí, které se nespecializuje na biologické simulace. Důvodem je lepší přehlednost, srozumitelnost a snadná replikace. Nespecializované nástroje a editory jsou také lepší pro pochopení principů vizualizace dat.[46]

7.1.1 Unity

Unity od společnosti Unity Technologies je všestranná aplikace, která se díky své flexibilitě stala oblíbenou především pro vývoj her, ale také pro simulace různých procesů a prostředí. [57]

Umožňuje vývojářům vytvářet detailní 3D i 2D simulace, což je ideální pro aplikace určené pro školení, vizualizace dat či virtuální realitu. Výbava Unity zahrnuje mimo jiné o fyzikální engine. Ten umožňuje realistické simulace fyzikálních jevů, což je obzvláště užitečné pro simulace, kde je klíčová interakce objektů v prostoru, jako jsou kolize nebo gravitační efekty. Pomocí jazyka C# se mohou v Unity programovat komplexní interakce mezi objekty a událostmi díky skriptům, což je vhodné pro simulace s automatizovanými nebo předem definovanými procesy. Tento nástroj se dá snadno propojit s dalšími softwary a nástroji pro práci s daty, což je užitečné u projektů, ve kterých je potřeba analyzovat velké množství dat nebo simulovat reálná prostředí. Unity podporuje VR a AR což je ideální pro tréninkové simulace nebo interaktivní prezentace, kde si uživatel může simulované prostředí prohlédnout zblízka. [57]

7.1.2 Unreal engine 5

Unreal Engine 5 od EpicGames používá jazyk C++ či Blueprints, které umožňují tvorbu bez větších programovacích znalostí. Má rozsáhlé 3D knihovny, ale i podporu 2D simulací. Je vhodně uzpůsoben pro práci s externími nástroji pro import i export dat, což z něj dělá užitečný nástroj simulování složitých jevů. [58]

Unreal Engine 5(UE5) nabízí oproti konkurenci mnoho výhod, například integrace nástrojů jako Nanite a Lumen, díky kterým má schopnost vytvářet robustní a složité světy, které umožňují dynamickou práci se světlem v reálném čase. Funkce jako Temporal Super Resolution, která umožňuje efektivní vykreslování scén se sníženým rozlišením se zachováním stejné podobné kvality, zlepšují výkon bez obětování kvality. Společně tyto funkce zvyšují efektivitu, věrnost a kreativitu, díky čemuž je UE5 ideální pro odvětví, jako je hraní her či architektura. [58]

7.1.3 OpenGL

OpenGL, také známé jako Open Graphics Library, bylo vytvořeno společností Silicon Graphics v roce 1992 jako průmyslové 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, simulací a vizualizace vědeckých dat. [59]

7.2 Porovnání

Pro vybrání nejvhodnější aplikace pro simulaci jsou stanovena bodově hodnocená kritéria:

- Programovací jazyk
- Komunita
- Cena
- Schopnosti programu
- Uživatelská přístupnost

Programovací jazyk, v kterém se bude simulaci definovat chování, je jedním z hlavních kritérií, neboť v sobě zahrnuje i autorovy zkušenosti s těmito jazyky. Další kritérium je aktivní a dostatečně rozsáhlá komunita, která zvyšuje efektivitu vytváření kódu a přístup k informacím. Cena je také důležitým kritériem, neboť určuje náklady na vývoj a rozvíjení

programu. Schopnosti programu přímo definují komplexnost kódu a správné použití dokáže urychlit vývoj. Poslední kritérium je jednoduchost práce s nástrojem.

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. OpenGL je jen api ale má podporu C#, která však není tak rozšířená. Komunitu mají všechny nástroje podobně rozšířenou, ale Unitysv závěsu Unreal engine 5 lépe splňuje trendy posledních let. Cenově je nejvýhodnější OpenGL, jelikož má všechny funkce dostupné zdarma. Následuje Unreal engine a nejhůře dopadlo Unity s více možnostmi licencí. Graficky je na tom pro 3D práci nejlépe Unreal Engine. OpenGL chybí mnoho nadstavbových funkcí, které by se vzhledem k API povaze knihovny museli doimplementovat. Uživatelsky nejprívětivější se jeví Unity. Lze se v něm snadno pohybovat a vše je po ruce.

	Unity	Unreal Engine	OpenGL
Jazyk	3	1	2
Komunita	3	2	1
Cena	1	2	3
Nástroje	2	3	1
Uživatelská přístupnost	3	2	1
Celkem bodů	12	10	8

9 Tvorba simulace v Unity

V této části, se práce zabývá seznámením se s prostředím a vytvářením simulace v Unity podle poznatků z teoretické části.

9.1 Prvky pro práci

Nejprve je třeba přiblížit prvky Unity, se kterými se bude v simulaci pracovat.

9.1.1 GameObject

Základním prvkem pro práci v Unity je GameObject. Zastupuje téměř všechny objekty, se kterými se dá v simulaci pracovat. GameObject se skládá z různých komponentů, které mu propůjčují nejrozličnější vlastnosti a definují jeho chování. Tyto komponenty se dají přidávat a měnit dle potřeby, což se hodí pro měnící se prostředí či hry.

Nejdůležitější komponenty pro simulaci jsou:

- Zobrazovací komponenty (Renderer)
- Chování (Skripty)
- Fyzika objektů (Collider)

Také má transformace, které definují jeho polohu, rotaci či měřítko. Podobně jako jiné vlastnosti, se dají měnit za běhu programu pomocí skriptů, nebo provést počáteční nastavení přímo v Unity.

9.1.2 Kamera

Kamera se stará o zobrazování simulace a dá se přizpůsobovat. V Unity se kamera definuje jako komponent, který se dá přiřadit k jakémukoliv objektu, se kterým se dá ve scéně pohybovat. Díky možnostem různých kamer se dá přenastavit úhel pohledu, nebo vytvořit nový pohled na simulaci s jinými parametry.

9.2 Datové struktury

Pro reprezentaci jater v digitálním prostředí, jsou třeba vytvořit vhodné datové struktury.

Vhodné nastavení struktur výrazně usnadní vývoj, přehlednost a logiku kódu.

9.2.1 Liver

Třída Liver je primární datová struktura. Ukládá do sebe hlavní sledované atributy, jako kyslík, alkohol, tuk a jak jsou játra zdravá a dále s těmito atributy pracuje. Odehrává se v nich poškození alkoholem, či fibróza při velkém množství nahromaděných toxinů. Také umožňuje regeneraci, pokud játra zrovna nejsou zatížena metabolizací tuků, či zpracováváním alkoholu.

9.2.2 Clustery

Části jater jsou rozděleny do clusterů buněk. Tyto clustery reprezentují jisté množství a druhy jaterních buněk, a mají určitý typ dle jejich funkce a účelu. V práci se rozlišují 3 druhy:

- Clustery jaterních buněk
- Clustery žilních buněk
- Clustery žlučnickových buněk

Na clusterech jaterních buněk se projevují změny stavu jater. Zastupují laloky neboli jaterní tkáň. Mohou nabrat stavy zdravé tkáně (hepatocyty), poškozené tkáně (poškozené hepatocyty), ztučněná tkáň (tuk) a fibróza. Žilní clustery reprezentují buněčné struktury, které jsou zodpovědné za zobrazení žil. Také se starají o udržení krve uvnitř cév. Žlučnickové clustery reprezentují žlučové cesty, kterými cestuje žluč.

9.2.3 CellTriangle

CellTriangle je vytvořen jako reprezentace clusterů ve scéně. Drží v sobě materiál, tag, transformační vlastnosti a vrcholy trojúhelníku. Hlavní metoda je přidání trojúhelníku do scény. Ty se přidávají s vykreslovací a kolizní mřížkou, kde je každému trojúhelníku přiřazena vrstva dle tagu. Cell triangle je tvořen 3 vykreslenými trojúhelníky ve scéně a 2 trojúhelníky určujícími chování kolizí.

```
public void AddTriangleToScene(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, 2, 1, 0, 1, 0, 2};
    mesh.RecalculateNormals();

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

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

    Mesh colliderMesh = new Mesh();
    colliderMesh.vertices = vertices;
    colliderMesh.triangles = new int[] { 0, 1, 2, 2, 1, 0};
    colliderMesh.RecalculateNormals();

    MeshCollider coll = triangle.AddComponent<MeshCollider>();
    coll.sharedMesh = colliderMesh;
    coll.convex = false;

    var body = triangle.AddComponent<Rigidbody>();
    body.isKinematic = true;
    body.collisionDetectionMode = CollisionDetectionMode.Continuous;
```

```

switch (tag)
{
    case ("Lobe"):
        triangle.tag = "Lobe";
        triangle.layer = 3;
        break;
    case ("PortalVein"):
        triangle.tag = "PortalVein";
        triangle.layer = 7;
        break;
    case ("HepaticPortal"):
        triangle.tag = "HepaticPortal";
        triangle.layer = 8;
        break;
    case ("VenaCanva"):
        triangle.tag = "VenaCanva";
        triangle.layer = 9;
        break;
    case ("GallBladder"):
        triangle.tag = "GallBladder";
        triangle.layer = 10;
        break;
}
SetTriangle(triangle);
}

```

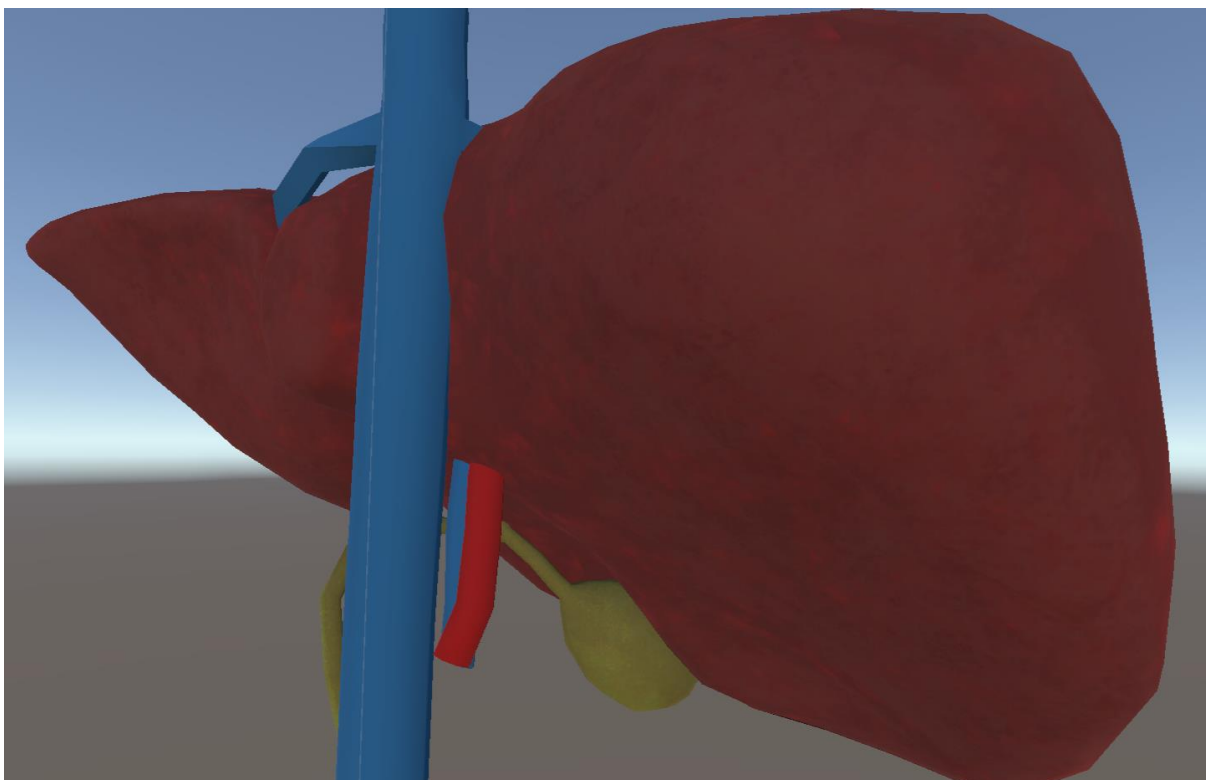
Ukázka kódu 2, přidání objektu Celltriangle do scény

9.2.4 BloodCell

O simulování vlastností krve v modelu se stará třída “BloodCell”. Ve scéně se zobrazuje jako koule a může v sobě nést hodnoty kyslíku, alkoholu a tuku, které se při kolizích přenášejí do jater.

9.3 Model jater

Pro simulování jater ve 3D, je pořízen 3D model pod licencí umožňující použití k bakalářské práci. V tomto modelu již jsou implementované základní struktury, jako žíly, tepny a žlučník. Tyto struktury jsou jednotlivé objekty tvořené meshem, které spojené dohromady tvoří model jater. [1]



Obrázek 5 Model jater v prostředí Unity [5]

9.3.1 Zpracování modelu pro práci

Pořízený model vizuálně připomíná játra, ale pro budoucí práci je v tomto stavu nevhodný, a je proto třeba ho vhodně zpracovat, což poskytne lepší kontrolu nad modelem.

Pro tento účel je vytvořen skript, který je přiřazen každé jednotlivé části jater. Tento skript projde mesh dané části, vybere z něj vrcholy, a následně je uloží do trojúhelníku, které následně dle uloží do seznamů.

```
void SetLiver()
{
    List<Vector3[]> meshTrianglesVertices = new List<Vector3[]>();

    ClusterOfLiverCells LiverCluster;
    ClusterOfVeinCells VeinCluster;
    ClusterOfGallBladderCells GallBladderCluster;

    ParseMeshToTriangles(meshTrianglesVertices);

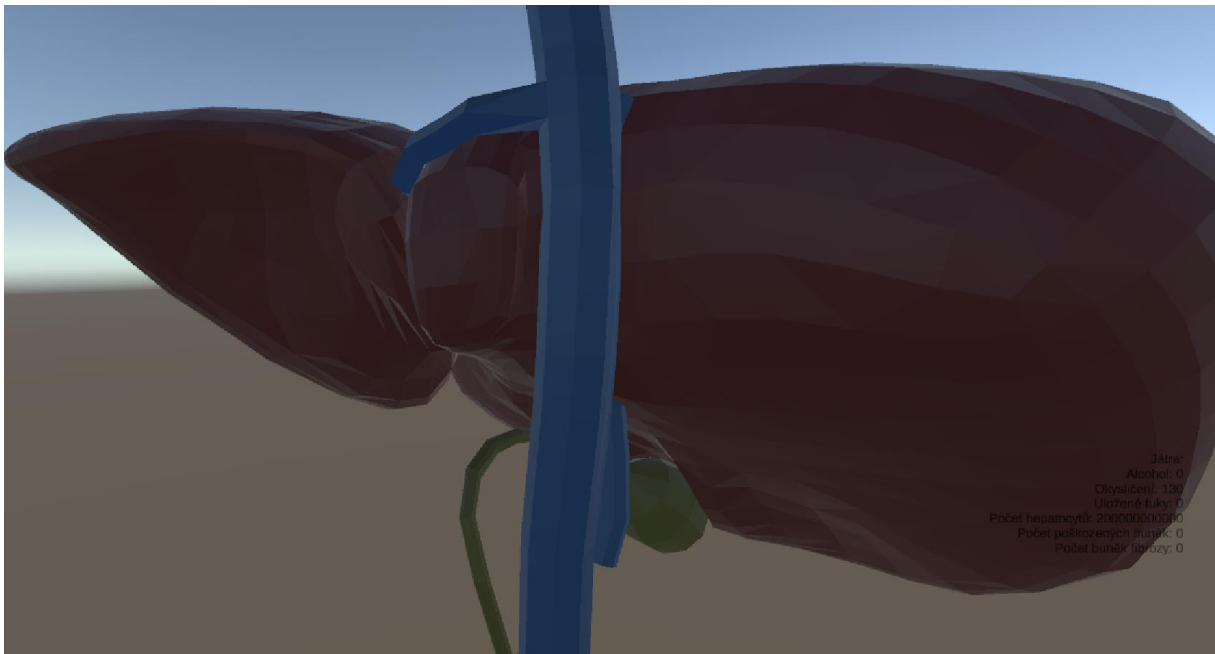
    for (int i = 0; i < meshTrianglesVertices.Count; i++)
    {
        switch (this.gameObject.tag)
        {
            case ("Lobe"):
                CellTriangle = new CellTriangle(HepatocyteM, transform,
                    GameLiverPart.tag, meshTrianglesVertices[i]);
                LiverCluster = new ClusterOfLiverCells(CellTriangle);
                Liver.liverClusters.Add(LiverCluster);
                CellTriangle.AddTriangleToScene(
                    LiverCluster.GetCellTriangle().GetPoints());
            break;
        }
    }
}
```

```

        break;
    case ("PortalVein"):
        CellTriangle = new CellTriangle(VeinM, transform,
            GameLiverPart.tag, meshTrianglesVertices[i]);
        VeinCluster = new ClusterOfVeinCells(CellTriangle);
        Liver.veinClusters.Add(VeinCluster);
        CellTriangle.AddTriangleToScene(
            VeinCluster.GetCellTriangle().GetPoints());
        break;
        .
        .
        .
    }
}
GetComponent<MeshRenderer>().enabled = false;
}

```

Ukázka kódu 1 Kód vytvoření a přiřazení struktur v játrech



Obrázek 6 Zpracovaný model jater

9.4 Nedokonalosti modelu

I když je model upravený, je poměrně náročné v něm simulovat fyziologické procesy. Pro optimalizaci prostředí do funkčního stavu, je třeba přizpůsobit fungování aplikace.

9.4.1 Propletenost krevních cest

Krevní cesty zpracované v modelu často procházejí skrz sebe. Díky tomu může často vznikat problém s kolizemi, neboť se krevní buňky nemusí do určitých částí krevních cest vůbec dostat. Tento problém byl vyřešen pomocí rozdělení objektů do různých vrstev.

Vlastnost “layer” (vrstva) je v Unity užitečná při organizaci a optimalizaci scény, především pro správu mezi interagujícími objekty. Objektu lze přiřadit určitou vrstvu což

zjednodušuje kontrolu kolizí a detekci fyziky. Správné využití této vlastnosti dokáže zoptimalizovat výkon a zjednodušit logiku kódu.

Proto tento účel je při přidání clusterů buněk do scény nastavena vrstva, ve které se objekt nachází. Následně se krev generuje dle bodu přímo do vrstvy, ve kterém se má pohybovat.

	Default	TransparentFX	Ignore Raycast	Lobe	Water	UI	Blood	PortalVein	HepaticPortal	VenaCanva	GallBladder
Default	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TransparentFX	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Ignore Raycast	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Lobe	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Water	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
UI	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Blood	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PortalVein	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
HepaticPortal	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
VenaCanva	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
GallBladder	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Obrázek 7 Nastavení kolizí mezi vrstvami

9.4.1 Nespojené krevní cesty

Protože jsou krevní cesty v modelu tvořeny 3 různými modely, tak nejsou navzájem propojené. To tvoří především problém při simulaci krevního toku játry. Tento problém se v práci řeší metodou počítání kolizí a vytyčení předem pevně daných pozic ve VenaCava.

Při každém odrazu krevní buňky se iteruje soukromý atribut bounces. Při dosažení předem dané hodnoty, se buňka přesune na jednu z daných pozic a změní si vrstvu pro správné fungování kolizí.

9.5 Průběh simulace

Po zpracování modelu a vytvoření datových struktur, je třeba začít simulaci. V této práci, simulace probíhá vypouštěním krve do žil a následným odrážením. O vypouštění krve se starají 2 Body: “GutPoint” a “HeartPoint”.

Bod GutPoint je pouze 3D bod umístěný na začátek tepny VenaCanva. Simuluje bod, kterým vstupuje do jater krev, která obsahuje živiny, tuky, toxiny a jiné látky ze střev. Používá se zde vzor singleton, který zajišťuje pouze jednu instanci třídy pro celý program. To je zvláště výhodné u skriptovací logiky v Unity, kde není jedna třída, která řídí chování celé aplikace. Skrze metodu "GenerateNewCell" v pravidelných intervalech generuje nové krevní buňky, kterým přidává určité množství vlastností jako alkohol či tuk, které je pak předávají je dále do jater.

HeartPoint funguje velmi podobně, pouze generuje krev do tepny HepaticPortal, a primární účel je okysličení jaterních buněk. Proto hlavní látka, která se přidává je kyslík.

```
void GenerateNewCell()
{
    Bloodcell = Instantiate<BloodCell>(prefab, ThisObject.transform.position,
    Quaternion.identity);
    Liver.bloodCells.Add(Bloodcell);
    Bloodcell.gameObject.layer = 7;
    Bloodcell.SetPosition(this.transform.position);
    int outAlcohol = 100;
    int outFat = 100;
    if (alcohol >= 100)
    {
        alcohol -= outAlcohol;
    }
    else
    {
        outAlcohol = alcohol;
        alcohol = 0;
    }
    if (fat >= 100)
    {
        fat -= outFat;
    }
    else
    {
        outFat = fat;
        fat = 0;
    }
    Bloodcell.SetAlcohol(outAlcohol);
    Bloodcell.SetFat(outFat);
}
```

Ukázka kódu 3 Generování krevních buněk ze střev [3]

9.5.1 Pohyb krve

Pohyb krve je realizován díky vlastnosti gameObjectu: Rigidbody.

Komponent "Rigidbody" se používá pro manipulování s objekty, na které je zapotřebí aplikovat fyzikální síly. Mezi tyto fyzikální síly patří především kolize, gravitace a aplikování síly pohyb těles.

Každý objekt v simulaci má přiřazen RigidBody. Při vytvoření objektu se vygeneruje náhodný vektor pro pohyb tělesa směrem nahoru, což vypadá vizuálně lépe než předem dané pohybové vektory. V Intervalu každých 1.5 sekund je provolávána metoda “CorrectFlow”, která koriguje směr krve.

CorrectFlow projde všechny krevní buňky, a nastaví jim pohyb dle jejich vrstvy a pozici v prostoru. Pohyb se nastavuje v rámci předem nastavených bodů. Zatím co krev v jaterních tepnách se pohybuje směrem od prvního bodu a následně druhého bodu pro lepší distribuci krve, krev z jaterní žíly se pohybuje směrem k bodu v hlavní jaterní žíle (Vena Canva Superior).

9.5.2 Předání vlastností

Kdykoliv se krevní buňka dotkne žilního clusteru buněk, tak se zavolá metoda “Transfer properties”. Tato metoda odečte sledované atributy z krevní buňky a přičte je do třídy Liver. Tento přenos závisí na atributu Třídy Liver “TransmissionRatio”, která se odvozuje od atributu zdraví (Lepší kondice jater = Efektivnější zpracování a získávání látek).

9.5.3 Sledování změn

Pro sledování změn v práci je implementován “displayText”, který zajišťuje vykreslení textu v aplikaci. Je pravidelně aktualizován, čímž umožňuje sledovat vnitřní stav jater, což pomáhá při pochopení jejich vnitřních pochodů

Dále je možné pozorovat změny pomocí změny barvy a typu jaterních clusterů, v závislosti stavu stavu jater (poměr buněk hepatocytů, poškozených hepatocytů, fibrózy a množství uloženého tuku).

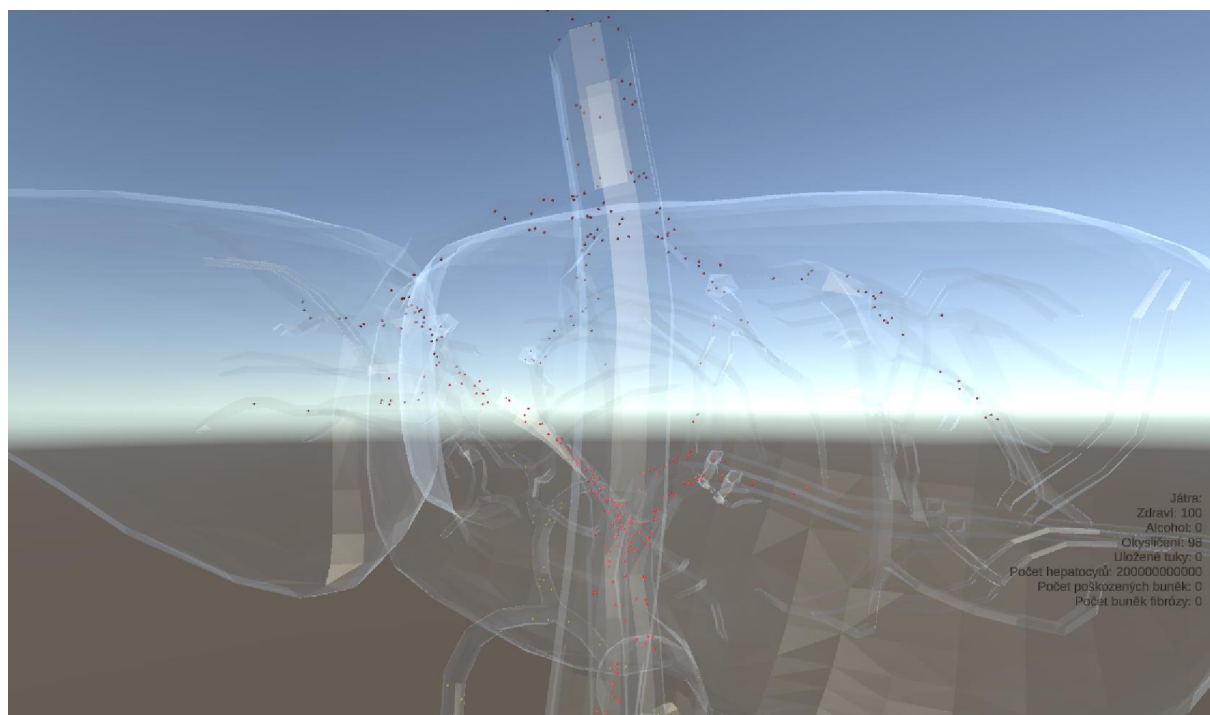
9.6 UI a Ovládání

Pohyb v aplikaci je zajištěn skriptem, který je přiřazen kameře. Zajišťuje podobné ovládání jako při návrhovém vytváření simulace, což zajišťuje ještě lepší přiblížení jaterních funkcí a volnost pohybu.

O ovládání simulace se stará třída “Controller”. Tato třída zpracovává uživatelské vstupy (NUM 1, NUM2, NUM9) a zajišťuje jejich vykonání. Funkce kláves:

- NUM1 – Přidání alkoholu pro vyslání do jater
- NUM2 – Přidání tuků pro vyslání do jater
- NUM9 – Zprůhlednění modelu jater

Tato třída také provádí korekce modelu, jako je tok krve a mazání nepotřebných částí krve



Obrázek 8 Průhledný model jater

10 Shrnutí výsledků

V prvním úseku teoretické části, práce pojednává o důležitosti jater, jejich funkcích a stavbě.

Druhá část práce se zabývá typy simulací a jejich zobrazení a možnostmi optimalizace.

Následně pojednává o používaných metodách pro simulace orgánů a organických struktur.

V praktické části je nejdříve provedeno seznámení s populárními nástroji pro 3D vizualizaci a následně vybírá nejvhodnější nástroj pro zpracování simulace. Poslední část práce poskytuje vhled do tvorby projektu prezentováním struktury a logiky vytvořené aplikace.

Praktický projekt představuje spíše vizualizaci než generování dat. Detailní simulace jaterních funkcí je komplexní na zpracování, ale je zpracován jednoduchý zástupný deterministický algoritmus. Na základě pořízeného modelu vytvoří vlastní model jater vhodný pro práci, generuje části krve, které následně simulují chování krve předáváním vlastností do jater. Na játrech je vidět změna stavu dle zaslaných vlastností. Vytvořený model je vhodně zpracovaný pro budoucí doplnění simulačních prvků a poskytuje zjednodušený pohled na vnitřní fungování orgánu, který je ve 3D a vizuálně příjemně zpracovaný.

10.1 Budoucí rozšíření

Aplikace poskytuje pouze povrchní představy o fungování jater a má vizuální účel. Nicméně u simulace orgánu tak složitého jako játra není snadné nemít další funkce k zobrazení.

Vizuálně se dá zlepšit samotné zobrazení a reprezentace Clusterů a krve. Aplikací Metakoulí místo trojúhelníků, by se daly vytvořit dynamické modely, které dokážou vytvořit detailnější modely jater, které více připomínají strukturu orgánu. Zároveň pomocí techniky cube(sphere) marching dokážou spojit plochy krve, díky čemuž bude více připomínat tekutinu. To však za cenu větší výpočetní náročnosti.

Simulace aktuálně pracuje pouze s pár látkami a spíše pro vizuální změny modelu. Implementací víceúrovňového modelu by bylo možné simulovat detailnější systém, který lépe odpovídal komplexnosti jater, například implementovat zpracování živin buňkami či zahrnutí času pro přesnější simulace biologických pochodů.

11 Závěr

V teoretické části, práce pojednává o důležitosti, fungování a funkcích jater a o možnostech simulací orgánů, čímž odpovídá na otázky položené v metodice. Praktická aplikace splnila většinu stanovených cílů a představuje vizualizaci jater, která přináší přehledný 3D pohled na simulované funkce a strukturu tohoto orgánu. Práce zdůrazňuje význam jater v lidském těle a představuje moderní přístupy v oblasti počítačových simulací orgánů, včetně metod zpracování a optimalizace výkonu. Po výběru vhodného softwarového prostředí se podařilo vytvořit přehlednou simulaci, která zahrnuje chování krevních složek a jejich interakce s játry. Praktická část práce poskytuje detailní pohled na architekturu aplikace, která zprostředkovává změny stavu jater na základě vlastností, které jim krev předává. Celkově práce přispívá k lepšímu pochopení principů simulace orgánů a nabízí inspiraci pro další rozvoj v této oblasti.

Seznam Použité literatury

- [1] Why It Matters: Overview of Body Systems <https://courses.lumenlearning.com/suny-wmopen-biology2/chapter/why-it-matters-overview-of-body-systems/>
- [2] Weinhart M. 3D organ models-Revolution in pharmacological research doi: 10.1016/j.phrs.2018.11.002 . Epub 2018 Nov 3. PMID: 30395949; PMCID: PMC7129286.
- [3] Health and Its Significance <https://www.toppr.com/guides/biology/why-do-we-fall-ill/health-and-its-significance/>
- [4] Luděk Šídlo, Pracovní vytížení praktických lékařů pro děti a dorost v Česku pohledem samotných lékařů [https://www.researchgate.net/publication/332407512 Pracovni vytizeni praktickych lekaru pro deti a dorost v Cesku pohledem samotnych lekaru](https://www.researchgate.net/publication/332407512_Pracovni_vytizeni_praktickych_lekaru_pro_deti_a_dorost_v_Cesku_pohledem_samotnych_lekaru) The workload of general paediatricians in Czechia as viewed by the physicians themselves
- [5] Britannica, The Editors of Encyclopaedia. "organ". Encyclopedia Britannica, 21 Sep. 2024, <https://www.britannica.com/science/organ-biology>
- [6] Body Tissues https://training.seer.cancer.gov/anatomy/cells_tissues_membranes/tissues/
- [7] What is a Cell? <https://www.nature.com/scitable/topicpage/what-is-a-cell-14023083/>
- [8] Britannica, The Editors of Encyclopaedia. "liver". Encyclopedia Britannica, 12 Nov. 2024, <https://www.britannica.com/science/liver> .
- [9] Liver: Anatomy and Functions <https://www.hopkinsmedicine.org/health/conditions-and-diseases/liver-anatomy-and-functions>
- [10] Hannah May, The Liver <https://teachmeanatomy.info/abdomen/viscera/liver/>
- [11] Ryan J. Schulze; The cell biology of the hepatocyte: A membrane trafficking machine. *J Cell Biol* 1 July 2019; 218 (7): 2096–2112. doi: <https://doi.org/10.1083/jcb.201903090>
- [12] Banales, J. Cholangiocyte pathobiology. (2019). <https://doi.org/10.1038/s41575-019-0125-y>
- [13] Nguyen-Lefebvre AT. 2015;1(1):101. Epub 2015 Aug 14. PMID: 26937490; PMCID: PMC4771376. <https://pmc.ncbi.nlm.nih.gov/articles/PMC4771376/>
- [14] Michel Félétou , Multiple Functions of the Endothelial Cells <https://www.ncbi.nlm.nih.gov/books/NBK57148/>
- [15] V. Tsutsumi Chapter 2 - Structure and Ultrastructure of the Normal and Diseased Liver, ISBN 9780128042748, <https://doi.org/10.1016/B978-0-12-804274-8.00002-3>. <https://www.sciencedirect.com/science/article/pii/B9780128042748000023>
- [16] Conley, C. Lockard and Schwartz, Robert S.. "blood". Encyclopedia Britannica, 31 Oct. 2024, <https://www.britannica.com/science/blood-biochemistry>

- [17] Liver: Anatomy and Functions <https://www.hopkinsmedicine.org/health/conditions-and-diseases/liver-anatomy-and-functions>
- [18] Nutrition And The Liver <https://www.liver.ca/wp-content/uploads/2020/06/Nutrition-And-The-Liver.pdf>
- [19] Alcohol Metabolism <https://www.bgsu.edu/recwell/wellness-connection/alcohol-education/alcohol-metabolism.html>
- [20] Stefania Manetti, Protein in diet <https://medlineplus.gov/ency/article/002467.htm>
- [21] Amino Acids <https://my.clevelandclinic.org/health/articles/22243-amino-acids>
- [22] Hundt M, Bile Secretion. [Updated 2022 Sep 26]. In: StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing; 2024 Jan-. Available from: <https://www.ncbi.nlm.nih.gov/books/NBK470209/>
- [23] Louis N. Gray, Computer Simulation <https://www.sciencedirect.com/topics/computer-science/computer-simulation>
- [24] Goldstine, H. H. (1972). The Computer from Pascal to von Neumann. https://monoskop.org/images/f/fc/Goldstine_Herman_H_The_Computer_from_Pascal_to_von_Neumann.pdf
- [25] Introduction to Modeling and Simulation Systems <https://uh.edu/~lcr3600/simulation/historical.html>
- [26] Eric Winsberg Computer Simulations in Science <https://plato.stanford.edu/entries/simulations-science/#EquBasSim>
- [27] Will Kenton, Stochastic Modeling: Definition, Advantage, and Who Uses It <https://www.investopedia.com/terms/s/stochastic-modeling.asp>
- [28] Agent-Based modeling <https://www.publichealth.columbia.edu/research/population-health-methods/agent-based-modeling>
- [29] Roberto A. Abbiati, Multiscale Modeling <https://www.sciencedirect.com/topics/engineering/multiscale-modeling>
- [30] What is Monte Carlo Simulation? <https://www.ibm.com/topics/monte-carlo-simulation>
- [31] Simulation vs. Visualization – what’s the difference? <https://www.visualcomponents.com/blog/simulation-vs-visualization-difference/>
- [32] Ikpe Justice Akpan, A comparative evaluation of the effectiveness of virtual reality, 3D visualization and 2D visual interactive simulation: an exploratory meta-analysis <https://journals.sagepub.com/doi/full/10.1177/0037549718757039>

- [33] Particles and Simulations
https://learn.foundry.com/modo/content/help/pages/simulation/particles/part_sims_overview.html
- [34] Virtual Reality (VR) and Augmented Reality (AR) <https://www.princetonreview.com/ai-education/vr-and-ar>
- [35] Optimization <https://www.techcareer.net/en/dictionary/optimization>
- [36] Yolanda Carson, Anu Maria Simulation optimization: methods and applications
<https://dl.acm.org/doi/pdf/10.1145/268437.268460>
- [37] Why is it important to optimize data for real-time 3D?
https://create.unity.com/optimizing-cad-data-for-real-time-3d-visualization?utm_source=linkedin&utm_medium=cpc&utm_campaign=nxt_abm_ind_emea_emea_en_co_soc-li_acq_pr_2023-07_dt-ind-skl-xr-snr_cc3022_dt-co1_id:201087456&utm_content=g-wp_s-sgl_nm:pixwp1-01-sz:1080x1080_dt-ind-skl-xr-snr_a_all_emea_co_id:287566096
- [38] Andreas Edesberg ,November 13, 2024, Mastering Level of Detail
<https://www.sloyd.ai/blog/mastering-level-of-detail-lod-balancing-graphics-and-performance-in-game-development>
- [39] 11 Jan. 2024 What is GPU Acceleration? <https://www.geeksforgeeks.org/what-is-gpu-acceleration/>
- [40] Preetam Kumar, What is Caching & Why we need it in our applications ?
<https://medium.com/@pritam73/what-is-caching-why-we-need-it-in-our-applications-f43d7e9a0dde>
- [41] Fernando & Fredy Martinez, Organic-Shaped Structures Design Using Genetic Algorithms and Metaballs <https://www.m-hikari.com/ces/ces2017/ces21-24-2017/p/hernandezCES21-24-2017.pdf>
- [42] Michael Levit, 2021 Computer Simulations in Service of Biology
https://www.researchgate.net/publication/354410862_Computer_Simulations_in_Service_of_Biology
- [43] Socrates Dokos, Modelling Organs, Tissues, Cells and Devices
<https://link.springer.com/book/10.1007/978-3-642-54801-7>
- [44] Stevens S. Andrews, Simulating cell biology [https://www.cell.com/current-biology/fulltext/S0960-9822\(06\)01775-1](https://www.cell.com/current-biology/fulltext/S0960-9822(06)01775-1)
- [45] Jonas Pleyer Agent-based models in cellular systems ,
<https://doi.org/10.3389/fphy.2022.968409>
<https://www.frontiersin.org/journals/physics/articles/10.3389/fphy.2022.968409/full>

- [46] Maciej Swat, Virtual-tissue computer simulations
<https://www.imagwiki.nibib.nih.gov/content/virtual-tissue-computer-simulations>
- [47] Nikola Milenkovski, The Role of 3D in Medical Training and Simulations
<https://iwconnect.com/3d-in-medical-training/>
- [48] Swat MH. Multi-scale modeling of tissues using CompuCell3D. Methods Cell Biol.doi: 10.1016/B978-0-12-388403-9.00013-8. <https://pmc.ncbi.nlm.nih.gov/articles/PMC3612985/>
- [49] Ahmadreza Ghaffarizadeh, PhysiCell: An open source physics-based cell simulator for 3-D multicellular systems <https://doi.org/10.1371/journal.pcbi.1005991>
<https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1005991>
- [50] Morpheus <https://morpheus.gitlab.io>
- [51] Cooper FR. Chaste: Cancer, Heart and Soft Tissue Environment. 2020 Mar 13;5(47):1848. doi: 10.21105/joss.01848. <https://pmc.ncbi.nlm.nih.gov/articles/PMC7614534/>
- [52] Yueyang Qu, Organ mimicking technologies and their applications in drug discovery,Intelligent Pharmacy<https://doi.org/10.1016/j.ipha.2023.05.003>.
<https://www.sciencedirect.com/science/article/pii/S2949866X23000345>
- [53] Pattanayak. Microfluidic chips: recent advances, critical strategies in design, applications and future perspectives. Microfluid Nanofluidics. 2021;25(12):99. doi: 10.1007/s10404-021-02502-2 .<https://pmc.ncbi.nlm.nih.gov/articles/PMC8547131/>
- [54] Zhao, Z., Chen, X., Dowbaj, A.M. et al. Organoids. Nat Rev Methods Primers 2, 94 (2022). <https://doi.org/10.1038/s43586-022-00174-y>
- [55] What is 3D bioprinting? <https://www.upmbiomedicals.com/solutions/life-science/what-is-3d-bioprinting/>
- [56] Dominik Kolouch, Jakub Beneš 2023 Vývoj počítačových her v enginu Unity
<https://theses.cz/id/6d0gdi/STAG98475.pdf>
- [57] What is Unity? 2024 Zenva <https://gamedevacademy.org/what-is-unity/>
- [58] Unreal engine 5 <https://www.unrealengine.com/en-US/unreal-engine-5>
- [59] Divanshu Singh, Computer Graphics <https://noobtomaster.com/computer-graphics/introduction-to-graphics-apis-opengl-directx/>

Seznam obrázků

- [1] <https://www.wikiskripta.eu/w/Játra>
- [2] <https://www.mypathologyreport.ca/pathology-dictionary/hepatocytes/>
- [3] Příklad rozložení úrovní víceúrovňové simulace
- [4] Ukázka využití částicového systému v simulaci, sand:box, SmellyMoo , GooglePlay
- [4] Deng J , Microfluidic liver model, <https://doi.org/10.1016/j.ipha.2023.05.003>
<https://www.sciencedirect.com/science/article/pii/S2949866X23000345>
- [5] 3D model jater
- [6] Zpracovaný 3D model jater
- [7] Nastavení kolizí v projektu Unity
- [8] Průhledný model jater

Seznam ukázek kódu

- [1] Ukázka kódu 1 Kód vytvoření a přiřazení struktur v játrech
- [2] Ukázka kódu 2 Kód přidání objektu Celltriangle do scény
- [3] Ukázka kódu 3 Kód generování krevních buněk

Použité modely

- [1] 3D exports, <https://3dexport.com/3dmodel-liver-anatomy-479483.htm>