

N-Body simulation

J. Franců

Fakulta jaderná a fyzikálně inženýrská, Břehová 7, 115 19 Praha 1

jan.francu@gmail.com

Abstrakt

Cílem mého projektu bylo vytvořit interaktivní simulaci problému n -těles. Finální projekt byl vytvořen za pomoci vývojového prostředí Unity3D, který je výborným nástrojem nejen pro vývoj her, ale jakéhokoli projektu s nutností 2D/3D vizualizace. Výsledkem je program, který dokáže v reálném čase simulovat pohyb hmotných bodů a zároveň umožňuje uživateli, na rozdíl od jiných simulací, přímo zasahovat do průběhu. Program tak tvoří velmi přístupný a zábavný učební nástroj.

1 Úvod

Na projekt mě přivedla přednáška ing. V. Svobody v zimním semestru, která se týkala numerických řešení diferenciálních rovnic. Tato matematicko-fyzikální úloha mě velice zaujala a rozhodl jsem se ji dále prozkoumat.

Stejně jako na přednášce, jsem na začátek použil pro výpočty tabulkový kalkulátor (v mém případě MS Excel). Začal jsem jednoduchou simulací lineárního harmonického oscilátoru a následně pohyby v centrálním gravitačním poli země.

Velký skok později způsobil nález fyzikálního appletu na internetových stránkách [1] kanadsko-polského čtrnáctideníku Nowy Kurier. Nebudu se zabývat tím, jak se na zmíněnou stránku dostal, ale stal se předobrazem mé budoucí aplikace. Jde o flashový applet s jednoduchým 2D rozhraním, který umožňuje interaktivně prozkoumat gravitační působení hmotných těles. Nejdůležitější je, že na rozdíl od ostatních appletů, kde zadáte pouze počáteční podmínky, je zde možnost již do rozběhnuté simulace vložit nový objekt, a tak okamžitě změnit situaci. Okamžitost je způsobena tím, že simulace probíhá v reálném čase. Na druhou stranu applet nedodržuje fyzikální velikost sil a jde více o fyzikální hříčku. Gravitační konstanta G je v tomto případě rovna jedné a hmotnosti jsou maximálně v řádech milionů kg. Jak jsem však posléze zjistil, jde o velice dobrou volbu z časového hlediska, protože přibližují roční děje na úroveň několika sekund.

2 Rozbor

Mým cílem bylo vytvořit aplikaci, která by splňovala následující: simulace problému n -těles v reálném čase, prvky pokročilé uživatelské interakce, schopnost simulovat i stovky objektů najednou, dostatečná fyzikální přesnost a rozumné uživatelské prostředí.

2.1 Problém n těles a jeho řešení

Jde o fyzikální problém zjištění polohy a rychlosti hmotných těles, které na sebe působí gravitační silou. Základní problém pohybu těles v centrálním gravitačním poli nehybného tělesa jsem již řešil a tak stačilo převést tento problém na více těles, z nichž žádné by nebylo "ukotvené". Na rozdíl od centrálního pole jednoho tělesa, kde bylo potřeba řešit pouze jednu diferenciální rovnici, zde se jich musí řešit n (počet těles). Diferenciální rovnice (1) vycházejí z Newtonova gravitačního a 2. pohybového zákona. Sumace na levé straně načítá silové příspěvky ostatních těles do celkové síly působící na dané těleso. Jedná se o metodu zvanou "direct approach" [2] a tedy celková složitost stoupá s $O(n^2)$.

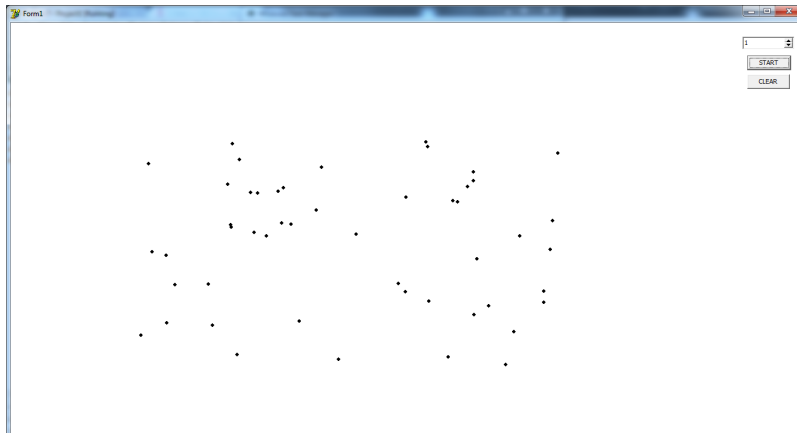
$$m_i \ddot{\mathbf{r}}_i = \sum_{j \neq i} G \frac{m_i m_j (\mathbf{r}_j - \mathbf{r}_i)}{|\mathbf{r}_j - \mathbf{r}_i|^3} \quad (1)$$

K řešení těchto rovnic jsem použil nejjednodušší metodu a to Eulerovu, protože analytické řešení pro n těles existuje jen v podobě pomalu konvergujících mocninných řad. Hlavním parametrem pro mnou zvolenou metodu je časový element, na kterém se řešení aproximuje pomocí přímků. Čím kratší interval, tím je řešení přesnější ale pomalejší a naopak. To platí však jen do určité meze, protože žádná numerická metoda není dokonalá.

3 Realizace

3.1 Delphi

Při prvním pokusu jsem sáhl po ověřeném vývojovém prostředí Delphi pracující se stejnojmenným programovacím jazykem. Vytvořil jsem tak program s jednoduchým uživatelským prostředím viz Obr. 1.



Obr. 1: Program vytvořený v Delphi, jednotlivé tečky reprezentují tělesa vykreslená na formulář, v pravém horním rohu lze volit počet generovaných těles a startovat simulaci nebo udělat reset počátečních podmínek.

Na začátku se vygenerují do pole náhodná tělesa (rychlost, hmotnost, poloha). Simulace probíhá pomocí třech jednoduchých vložených for cyklů. První krokuje po časových elementech, dokud nedosáhne předem daného času, druhý a třetí se stará o výpočet sil (tedy i zrychlení) působící na každé těleso. Z těch se poté Eulerovou metodou vypočítává celková změna polohy a rychlosti těles. Následně proběhne část programu, která se stará o aktualizaci proměnných jednotlivých těles a zobrazení (vykreslení) změn na formulář.

Velice záhy jsem narazil na několik problémů, které mě přesvědčily, že Delphi nebude cestou k cíli. Prvním problémem bylo, že Delphi neumožňuje více vláknové aplikace. Znamenalo to velikou překážku, protože, jakmile jsem spustil simulaci, již nebylo možné jí žádným příkazem pozastavit. Další problém byl s vykreslováním, kdy po každém kroku jsem musel celý formulář překreslit. Při vyšší obnovovací frekvenci to způsobovalo časté blikání, které se mi nepodařilo odstranit a zamezilo tak dalšímu vývoji.

3.2 Unity3D

Pro další vývoj jsem použil vývojové prostředí Unity3D, které mi bylo doporučeno mým bratrem. Jedná se především o herní vývojové prostředí, ale jde použít i pro mnoho jiných aplikací, které vyžadují 3D vizualizaci. Programuje se zde pomocí skriptů, které lze psát v jazyce C# nebo Java. Velkou výhodou je schopnost převést vytvořenou aplikaci z Windows na MacOS, Linux či herní konzole a mobilní telefony.

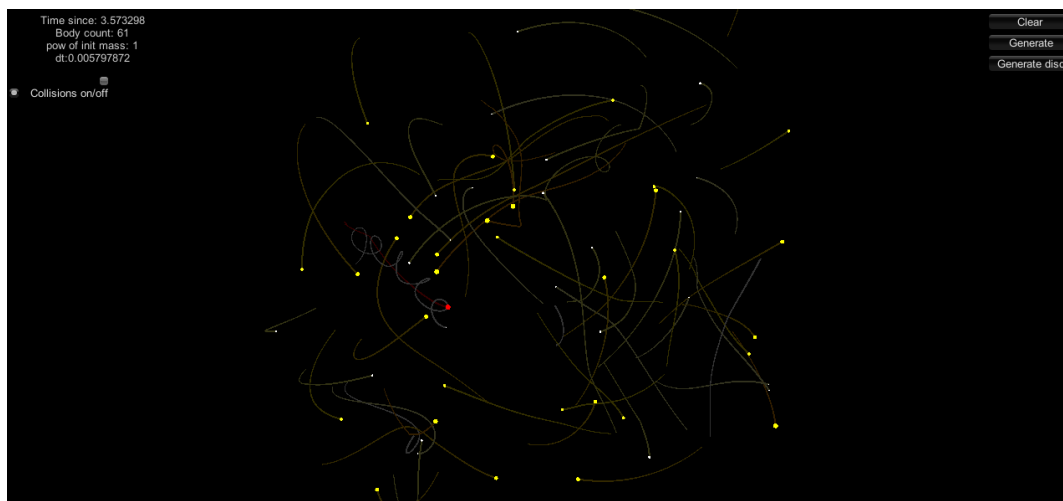
Byl to pro mě veliký skok, ale nakonec se mi podařilo převést kód do C#. Nové prostředí vyřešilo všechny předchozí problémy. Vykreslování probíhalo ve 3D pomocí DirectX. Základní multitasking mi poskytla změna algoritmu simulace, která nyní neprobíhala pouze po určité době ve for cyklu, ale nacházela se v části skriptu volaná každý vykreslený snímek obrazovky. Do stejné části jsem umístil i detekční podmínky pro interakci (přidávání a odebrání těles, zoom, posun obrazovky,...).

Přidávání nových objektů probíhá pomocí systému drag&drop, kdy se při držení Ctrl volí kolečkem myši hmotnost vloženého objektu, následně táhnutím přes obrazovku se určí vektor polohy a rychlosti. Hmotnost se udává v desítkové logaritmické stupnici, kde zároveň logaritmus hmotnosti odpovídá poloměru jednotlivých těles na obrazovce. Oproti Delphi verzi jsem navíc přidal nepružné srážky těles a možnost generace protoplanetárního disku. Dalším vylepšení přineslo ukládání dat o tělesech do spojovaného seznamu, který mi umožnil lépe přistupovat k jednotlivým datům.

4 Hodnocení a budoucnost

Navenek vše vypadá velice pěkně a myslím, že jsem v mnoha ohledech i předčil původní applet. Největším problémem zůstává celková pomalost programu, která je způsobena špatnou optimalizací algoritmu. Narazil jsem také na své protichůdné požadavky. Program nemůže z hardwarového hlediska zvládat simulace v reálném čase stovek objektů a zároveň dodržovat dostatečnou fyzikální přesnost. Jsem si vědom, že existují i jiné metody řešení problému n těles, jako např. "metody stromů" nebo Particle Mesh, ale mnou zvolená metoda se mi zdá nejprůběžnější, a proto bych ji zatím neměnil.

Východiskem při dodržení stejné metody a požadavků by mohla být paralelizace. Ta lze



Obr. 2: Uživatelské rozhraní finálního programu. Vlevo se nachází údaje o uběhlém čase, počtu těles, stavu srážek (vyp/zap), zvolené hmotnosti, elementu času, který se dá upravovat pomocí posuvníku. Vpravo je tlačítko pro vyčištění simulace, generace náhodného počtu objektů a generace protoplanetárního disku. Barvy těles se odvozují podle hmotnosti.

velice dobře aplikovat v části, kdy se vyhodnocují silové příspěvky ostatních těles. Mohly by tak jednotlivé procesory obdržet vždy jen pár těles na vyhodnocení. Nejpřístupnější hardware na paralelní výpočty jsou grafické karty od NVIDIA, které již pár let nabízejí výkon stovek nízkofrekvenčních CUDA procesorů pro vlastní využití. Jelikož jsem vlastníkem takovéto karty, vyzkoušel jsem veřejně dostupnou demonstraci programu [3] pro řešení problému n těles a byl jsem schopen simulovat naráz i 1000 těles. Do budoucnosti bych chtěl tedy zavést tuto metodu a optimalizovat celý program.

Poděkování

Rád bych poděkoval ing. V. Svobodovi za přednášku v zimním semestru, která mě přivedla na cestu modelování. Zároveň bych poděkoval svému bratru A. Franců za pomoc s prostředím Unity3D a přechodem na jazyk C#.

Reference

- [1] anonym, applet na Nowykurier.com,
<http://nowykurier.com/toys/gravity/gravity.html>
- [2] P. Hut, M. Trenti, *N-body simulations*,
[http://www.scholarpedia.org/article/N-body_simulations_\(gravitational\)](http://www.scholarpedia.org/article/N-body_simulations_(gravitational))
- [3] M. Harris, L. Nyland, J. Prins, *Fast N-Body Simulation with CUDA*,
http://http.developer.nvidia.com/GPUGems3/gpugems3_ch31.html