
Programátorská dokumentace Raymagic

Marek Bečvář

Obsah

I	Popis	3
i	Naplnění funkčních požadavků	3
II	High-level	3
i	Herní loop	3
ii	Hráč	4
iii	Objekty	4
iv	Fyzikální objekty	5
v	Raymarching algoritmus	5
vi	Překladač - editor map	5
III	Popis dalších tříd	5
i	ConsoleMenu	5
ii	Graphics	5
iii	Informer	6
iv	Screen	6
v	Map / MapData / SaveContainer	6
vi	BVH / BVHNode	6
vii	Interactable	6
viii	PhysicsObject	7
ix	TransformHelper	7
x	Portal / PortalLogic	7
IV	Závěr	7

I Popis

Projekt měl za cíl vytvořit v jazyce C# a frameworku Monogame 3D vykreslovací raymarching engine, na kterém je postavená interaktivní ukázka hry napodobující hru Portal se všemi základními poznávacími vlastnostmi.

i. Naplnění funkčních požadavků

(z finální verze specifikace zápočtového projektu)

- úspěšná paralelizace algoritmu raymarching pro vykreslování
- implementace abstraktní třídy pro objekt, umožňující jednoduché rozšíření možných objektů
- fyzikální simulace jednotná pro hráče a všechny fyzikální objekty
- interaktivní objekty předávající signály o interakci pomocí delegátů
- funkční portály, přenos paprsků přes portály, přenos hráče a dalších přenosných objektů přes portály, zachování rychlostí v portálech
- načítání map v průběhu běhu úrovní, seamless automaticky generované načítací mapy
- implementace logiky hry Portal

II High-level

i. Herní loop

Entry point programu je samozřejmě třída metoda `Main` třídy `Program`. Ta ale hned spouští třídu frameworku Monogame `MainGame`, kterou budu zde brát jako za skutečný entry point. Zde proběhne všechna dlouhodobá inicializace (třídy zpracovávající grafiku, hráče, provně, ...), načtení využitého fontu a dalších a následuje přechod do `update` a `draw` smyčky, kde se program do ukončení točí.

Update loop Z `MainGame` třídy se z `update` cyklu program dělí do řady míst. Posílají se do třídy hráče pro kontrolu stisku kláves, tak pro `update` fyziky hráče. Řeší se zde zároveň také `update` úrovní (výpočet fyzikálních objektů v úrovních a interaktivních objektech) a test portálů, jestli jimi neprošly nějaké objekty.

Draw loop Draw loop poté už pouze volá hlavní třídu starající se o vykreslování na obrazovku **Screen** a dále každý frame povoluje informačnímu systému **Informer** vypisovat skrz minulý cyklus nastřádané informace.

Upozornění: Objekty nezávislé na interních cyklech Ve hře existuje řada objektů, které se neřídí těmito cyklickými updaty, ale fungují ve vlastním načasování (obvykle skrz nekonečné cykly s delay přes Task třídu). Příklady těchto jsou například lasery a interaktivní objekty. Pro správné fungování těchto typů objektů musíme kontrolovat a čekat, aby nejprve doběhl vlastní cyklus programu a až poté je možno dělat nějaké změny (zabránit tak kolizím a úpravám listů, se kterými právě pracujeme).

ii. Hráč

Všechna logika hráče je situovaná ve třídě hráče - **Player**. Od inicializace a vytváření modelu hráče po update fyziky a kontrolu vstupů uživatele pro hráče je vše řešené uvnitř této třídy.

iii. Objekty

Šablona Hlavní třídou objektů je abstraktní třída **Object**, od které dědí všechny 3D objekty aplikace. Stará se o všechnu práci s vykreslováním, spravuje potřebné transformace objektů a tvoří solidní základ pro rozšíření seznamu objektů.

Signed distance function Třída blízko spojená s třídou pro objekty je třída **SDFs**, která pouze obsahuje výpis všech statiských metod využívaných pro jednotlivé objekty. Pokud je potřeba přidat do programu nový objekt, nejprve je potřeba přidat do této třídy odpovídající metodu, která bude využívána ve výpočtech **SDF** pro objekt.

Odvozené třídy objektů Abstraktní třída **Objekt** je navržena dostatečně solidně, že při rozšiřování seznamu objektů stačí vytvořit novou třídu pojmenovanou jak chceme, dodat do třídy **SDFs** novou statickou metodu pro výpočet **SDF** a vytvořit třídě vlastní konstruktor, odpovídající potřebám daného objektu.

iv. Fyzikální objekty

Fyzikální prostor Fyzikální prostor drží všechny aktuální fyzikální objekty v úrovni je zajišťování třídou `PhysicsSpace`, jehož `update` metoda je volána z `update` funkce mapy každý frame.

Fyzikální solver Řešič fyzikálních situací je implementován v třídě `PhysicsSolver`. Tomu je z prostoru předáván seznam objektů a solver následně nad seznamem provádí operace založené na **verlet integration** metodě.

v. Raymarching algoritmus

Velkou důležitou částí aplikace je samotný raymarching algoritmus. Ten se nachází ve třech statických metodách třídy `RayMarchingHelper`. První metoda řeší raymarching vracející barvu výsledného objektu, na který proces narazil. Druhá je metoda využívající raymarching pro propagaci paprsků světla. Třetí metoda je fyzikální raymarching, vracející výsledný nalezený objekt, vzdálenost, kam paprsek urazil a další. Tento typ raymarchingu je využíváný pro kontroly kolizí objektů a míření hráče.

vi. Překladač - editor map

Velkou částí projektu je překladač vlastního programovacího jazyka pro zadávání úrovní a modelování objektů - `TxtMapCompiler`. Tato třída obsahuje celý proces překládání a kontrolu chyb v **.map** textových souborech využívaných pro zadávání map.

III Popis dalších tříd

i. ConsoleMenu

Třída vytvářející celé úvodní konzolové menu, ve kterém hráč volí mezi dostupnými herními módy a dalšími možnostmi. Kvůli jednoduchosti problému je ve třídě zároveň vykreslování i logika menu.

ii. Graphics

Třída zajišťující veškeré vykreslování, jak barevných polí, tak veškerého textu po spuštění grafické aplikace (po konzolovém menu).

iii. Informer

Informační statická třída sbírající debug informace z libovolných tříd a z vlastního překladače za doby překladu. Výpis nashromážděných dat je pak proveden v draw loopu třídy `MainGame`.

iv. Screen

Třída starající se o vykreslování výsledků raymarching algoritmu na obrazovku (pohled hráče) a vykreslení aktuálních kurzorů pro hráče.

v. Map / MapData / SaveContainer

`Map` je třída spravující všechny informace ohledně aktuální a načítání následující úrovně. Pracuje zde všechna logika okolo načítání a před-načítání dalších map. Šablona pro všechna data o úrovních je třída `MapData`. Do této třídy jsou data ukládána ať už při vytváření map skrz kód nebo skrz vlastní editor a v této formě jsou dál po aplikaci data předávána. `SaveContainer` je kontejner obalující předpočítaná data map (předpočítané vzdálenostní funkce statických objektů, včetně odkazů na objekty), umožňující jejich serializaci.

vi. BVH / BVHNode

Projekt při optimalizaci vykreslování dynamických objektů využívá BVH - *Bounding Volume Hierarchy*. To spočívá v ohraničení objektů pomocí bounding boxů a následné rozdělení prostoru bounding boxů do stromové struktury. Díky tomu následně musíme pouze ty objekty, které nás opravdu zajímají (projdou testem bounding boxů). Vytváření této struktury a hledání objektů pomocí této struktury zajišťují třídy `BVH` a `BVHNode`.

vii. Interactable

Třída `Interactable` je podobně jako třída `Object` abstraktní třídou, šablonou pro objekty, se kterými může hráč nějak interagovat (obvykle objekty centrální pro logické problémy v jednotlivých úrovních). Tyto třídy obsahují metody na přípravu modelu objektu, jelikož modely těchto objektů jsou často skládané z více typů objektů (dynamické i statické). Dále upravuje styl jejich vykreslování. Oproti klasickým objektům se zde často pracuje s `Tasky` pro různé interaktivní funkce a s delegáty pro předávání signálů mezi objekty.

viii. **PhysicsObject**

Šablona dalších a zároveň aktivně využívaný fyzikální objekt, představující dvoubarevnou kuličku, použitelná pro řešení fyzikálních hádanek.

ix. **TransformHelper**

Třída obsahující řadu statických funkcí pracujících s transformačními maticemi, které jsou využívány při rotaci a translaci objektů ve hře (důležité pro ray-marching algoritmus).

x. **Portal / PortalLogic**

Samostatná třída (parciální třídy) pro portál a jeho logiku. **Portal** obsahuje funkce vykreslování portálu, nebo vykreslování přenosu paprsků přes portál, pokud jsou oba aktivní. **PortalLogic** se stará o kontrolu a přenos portálovatelných objektů přes portály.

IV Závěr

Projekt byl vytvořen jako závěrečná semestrální práce pro předměty *NPRG035 Programování C#* a *NPRG038 Pokročilé programování C#* - zimní a letní semestr 2021/22 - Matfyz UK.