

# PATH PLANNING IN 3D SPACE

**Radek Sasýn**

Master Degree Programme (2), FIT BUT

E-mail: xsasyn00@stud.fit.vutbr.cz

Supervised by: Jaroslav Rozman

E-mail: rozmanj@fit.vutbr.cz

**Abstract:** This work describes path finding among obstacles in 3D space using probabilistic algorithms. Users can create scene in application GUI – define start object, obstacles, goal position and run probabilistic algorithm. The finding path is visualized. The work describes probabilistic algorithm, collision detection and the basics of 3D graphics and shows the appropriate libraries to implement various parts.

**Keywords:** probabilistic algorithm, path planning, collision detection, 3D space, OpenGL, C++

## 1 ÚVOD

Hledání cesty objektu ve spojitém prostoru mezi překážkami je pro počítač velmi náročný problém. Ve spojitém prostředí nelze naivním přístupem zaručit nalezení výsledné cesty v konečném čase, jelikož prohledávaný prostor obsahuje nekonečné množství stavů. Jedním z možných přístupů je použití pravděpodobnostních algoritmů, které jsou schopny stavový prostor redukovat a tím zvýšit pravděpodobnost nalezení výsledku v konečném čase.

Cílem práce je vytvořit aplikaci s grafickým uživatelským rozhraním, která umožní uživateli vytvořit vlastní scénu a spustit implementovaný pravděpodobnostní algoritmus se zvolenými atributy. Tvorba scény musí umožnit objekt přidávat a odebírat, ale také s objekty manipulovat. Dále je potřeba, aby si uživatel mohl zvolit startovní objekt a cíl cesty. Po provedení algoritmu aplikace vizualizuje nalezenou cestu.

## 2 UŽIVATELSKÉ ROZHRAŇÍ A ZOBRAZENÍ

Výsledný vzhled aplikace lze rozdělit na dvě části – uživatelské rozhraní a zobrazení, obnášející práci s objekty.

### 2.1 UŽIVATELSKÉ ROZHRAŇÍ

Uživatelské rozhraní musí umožnit uživateli pohodlně pracovat s aplikací – vytvořit scénu podle vlastních představ, nastavovat parametry pravděpodobnostního algoritmu, zobrazit výsledek (nalezenou cestu) a další. K tomu je potřeba vytvořit ovládací prvky. Ovládání aplikace je zajištěno dvěma panely. Jeden panel obsahuje seznam použitelných objektů, které aplikace načítá z daného adresáře (*./objects*) a nastavitelné atributy pravděpodobnostního algoritmu. Druhý panel slouží jako seznam objektů ve scéně, obsahuje pro každý přidaný objekt možnost nastavení jako startovní nebo cílový objekt, popřípadě objekt aktivovat a upravit jeho pozici, rotace nebo velikost. Další možností je také smazání objektu ze scény.

Pro vytvoření panelů slouží knihovna *AntTweakBar*. Tato knihovna umožňuje vytvářet panely a do nich vkládat různé funkční prvky jako tlačítka, editační boxy, ovladače rotace a jiné. Další použitá knihovna se nazývá *SDL* a slouží k vytvoření aplikačního okna.

## 2.2 ZOBRAZENÍ

V rámci zobrazení byla nastudována tvorba 3D grafiky a modelování 3D objektů. Bylo potřeba nastudovat způsoby reprezentace 3D objektů (sít' trojúhelníků, hraniční reprezentace, CSG) a možnosti vytváření těchto objektů [3]. K tvorbě objektů byl použit nástroj *Blender*. Pro objekty použitelné v implementované aplikaci byl zvolen formát *.obj*, jiné formáty v rámci základní implementace aplikace nepodporuje.

Pro tvorbu grafiky byla použita knihovna *OpenGL*. *OpenGL* poskytuje širokou škálu možností pro tvorbu 3D grafiky aniž by byla potřeba znát grafický hardware počítače. Další nespornou výhodou je kompatibilita s ostatními používanými knihovnami (*AntTweakBar*, *SDL*, knihovny pro detekce kolizí).

## 3 DETEKCE KOLIZÍ

Detekce kolizí ve scéně úzce souvisí s výkonností celkové aplikace, jelikož je spouštěna velmi často (v rámci pravděpodobnostního algoritmu) a komplexně na celou scénu. Efektivní a přesná implementace detekce kolizí výrazně ovlivňuje rychlost a kvalitu výsledné implementace aplikace.

Detekce kolizí probíhá ve dvou fázích. V první fázi se zjišťují dvojice objektů, které spolu mohou kolidovat. V této fázi se scéna rozdělí na části za použití struktur k tomu určeným (pravidelná mřížka, segmentový nebo intervalový strom a podobně). Ke zjednodušení těles (mohou mít velmi složité tvary) se používají takzvané obálky (Osově zarovnaný box – AABB, Obalová koule – OBB a další). Pomocí obálek lze tvar tělesa zjednodušit a tím urychlit hledání kolizí. Ve druhé fázi se provádí hledání objektů (za použití dvojic objektů nalezených v první fázi), které spolu kolidují. Jelikož se v této fázi zjišťují už pouze kolize dvou objektů, provádí se test trojúhelník vůči trojúhelníku [2].

Pro implementaci detekce kolizí existuje množství knihoven. Jedná se například o knihovny *ColDet*, *OZCollide* a *FreeSOLID*. Tyto knihovny umožňují použít efektivní detekci kolizí mezi dvěma a více složitými objekty. Implementují možnosti použití množství technik vhodných zejména pro interaktivní aplikace.

## 4 PRAVDĚPODOBNOSTNÍ ALGORITMY

Pravděpodobnostní algoritmy pracují se spojitým prostředím, které nejprve navzorkují a poté hledají cestu nad grafem diskrétního prostředí. Vzorkování probíhá pomocí generování náhodných bodů v prostoru. Nagenované body jsou pospojovány do výsledného grafu, ve kterém následně probíhá hledání cesty. Aby bylo možné hledat cestu ve spojitém prostředí pomocí pravděpodobnostních algoritmů, musí být toto prostředí předem známé, tedy existuje nějaký popis tohoto prostředí (model prostředí). Pokud tato podmínka není splněna, nelze pravděpodobnostní algoritmy pro hledání cesty použít. Pomocí pravděpodobnostních algoritmů je možné hledat cesty objektů i ve vícedimenzionálním prostoru [1].

Pravděpodobnostní algoritmy lze rozdělit na (podle možnosti použití vytvořeného grafu prostředí opakovaně na různé body):

- *Vícetázové* – vytvořený graf prostředí lze použít opakovaně. Lze tedy hledat cesty mezi libovolnými dvěma body bez opětovného generování grafu prostředí.
- *Jednotázové* – nelze použít vygenerovaný graf vícekrát, jelikož grafem je strom vytvořený mezi počátečním a cílovým bodem.
- *Kombinované* – Kombinuje předchozí přístupy.

#### 4.1 PRM ALGORITMUS

PRM algoritmus je jedním ze základních pravděpodobnostních algoritmů pro hledání cesty ve více-dimenzionálním prostoru. Jedná se o vícedotazový pravděpodobnostní algoritmus, který pracuje ve dvou krocích. V prvním kroku se buduje graf prostředí (roadmapa). Ve druhém kroku lze poté vyhledávat cesty mezi libovolnými uzly grafu konfiguračního prostoru, který byl vytvořen v prvním kroku [1].

#### 4.2 EST ALGORITMUS

EST algoritmus je jednodotazový pravděpodobnostní algoritmus. Pracuje podobně jako PRM algoritmus s tím rozdílem, že graf volného konfiguračního prostředí je strom, který má počáteční uzel jako kořen. Snahou algoritmu je spojit cílový uzel s budovaným stromem [1].

#### 4.3 RRT ALGORITMUS

RRT algoritmus je dalším jednodotazovým pravděpodobnostním algoritmem, který k popisu volného konfiguračního prostoru používá dva stromy – první s kořenem v počáteční konfiguraci, druhý s kořenem v cílové konfiguraci (lze použít i jeden strom podobně jako u EST algoritmu). Podobně jako EST algoritmus tvoří RRT algoritmus postupně oba stromy, které se poté snaží spojit, čímž zaručí nalezení cesty mezi počátečním a koncovým uzlem [1].

#### 4.4 SRT ALGORITMUS

SRT algoritmus kombinuje jednodotazový a vícedotazový přístup pravděpodobnostních algoritmů. Vícedotazový je z hlediska tvorby grafu volného konfiguračního prostoru (roadmapy), který tvoří pomocí některého z jednodotazových pravděpodobnostních algoritmů. Průběh SRT algoritmu lze rozdělit na dvě fáze, v první fázi se vytváří graf volného konfiguračního prostoru a v druhé fázi se vyhledávají cesty mezi libovolnými body ve volném konfiguračním prostoru [1].

SRT algoritmus byl zvolen jako vhodný pro implementaci, jelikož je nejefektivnější a vhodnou úpravou parametrů lze SRT algoritmus degradovat na kterýkoliv jiný ze zmíněných pravděpodobnostních algoritmů.

### 5 ZÁVĚR

V rámci projektu byla nastudována problematika tvorby 3D grafiky a 3D modelů, dále detekce kolizí a pravděpodobnostní algoritmy. Byl proveden návrh aplikace – volba dostupných knihoven, které jsou vhodné pro implementaci jednotlivých částí, návrh uživatelského rozhraní, volba pravděpodobnostního algoritmu a vytvoření vzorku 3D objektů použitelných pro testování. Navržená aplikace bude implementována a otestována v rámci diplomové práce.

### REFERENCE

- [1] Choset, H.M.: Principles of Robot Motion: Theory, Algorithms, and Implementation, Mit Press, 2005, ISBN 9780262033275
- [2] Ericson, C.: Real-Time Collision Detection, Elsevier, 2005, ISBN 9781558607323
- [3] Žára, J., Beneš, B., Sochor, J., Felkel, P.: Moderní počítačová grafika, Computer Press, 2004, ISBN 9788025104545