

Detekcia kolízií

Úlohou detekcie kolízií je vyrátať kedy a kde sa objekty prekrývajú. Na zistenie, či sú telesá v rovine používame metódu Sweep And Prune:

- SAP (sweep and prune) $\left\{ \begin{array}{l} \text{založené na separating axis theorem (SAT)} \\ \text{založené na separating plane theorem (SPT)} \end{array} \right.$

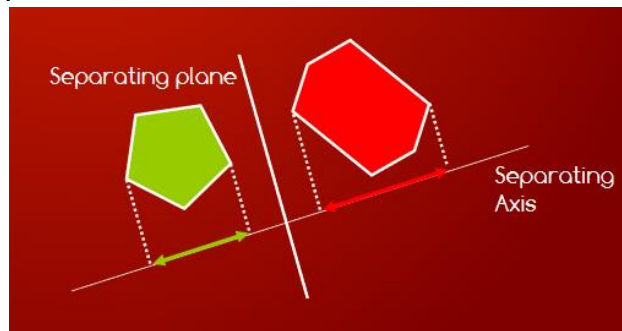
Nutná a postačujúca podmienka kedy nie sú dve telesá v rovine

SAT

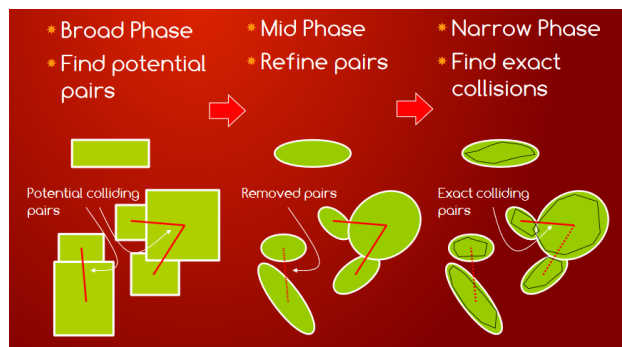
Dva konvexné objekty sa neprekrývajú vtedy a len vtedy, ak existuje (oddeľujúca) os, na ktorej sú projekcie objektov oddelené.

SPT

Dva konvexné objekty sa neprekrývajú vtedy a len vtedy, ak existuje (oddeľujúca) rovina, ktorá oddeľuje objekty.



Poznáme tri štádia detekcie kolízie. Broad Phase, Mid Phase, Narrow Phase.



Broad Phase

Ide o približnú detekciu. Myšlienkou tohto algoritmu je rýchlo nájsť páry objektov, ktoré môžu byť v kolízií a zamietnuť tie, o ktorých vieme s istotou povedať, že nie sú v kolízií.

Techniky:

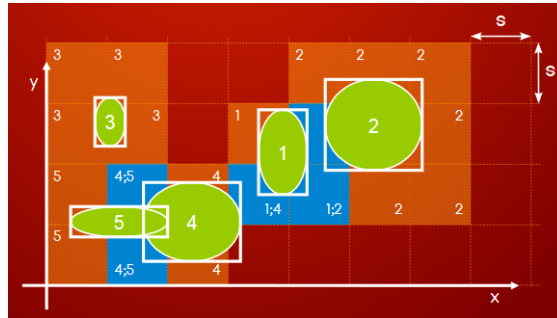
- Rovnomerné priestorové delenie - Hierarchická mriežka
- Komplexné priestorové delenie
- triedenie súradníc

Táto fáza je náročná na paralelizáciu pre GPU.

Uniformná (rovnomerná) mriežka

Majme rovnomernú (uniformnú) mriežku s veľkosťou bunky s . Pre každý bod $p = (x, y, z)$ vieme nájsť korešpondujúcu bunku $c = (i, j, k) = T(p)$. T je mriežkovacia (pokrývacia) funkcia, ktorá rozdelí priestor podľa veľkosti bunky nasledovne $T(p) = ([x/s], [y/s], [z/s])$.

Princíp spočíva v postupnom vkladanií objektov do mriežky. Najprv vložíme prvý objekt ($id=1$) do mriežky. Do každej bunky, ktorou objekt prechádza zapíšeme jeho id na základe jeho AABB obálky. Potom vložíme druhý objekt ($id=2$) do mriežky... Po vložení všetkých objektov budú niektoré bunky obsahovať id len jedného objektu, ale niektoré bunky aj viacerých objektov. Tieto bunky definujú kolidujúce páry.

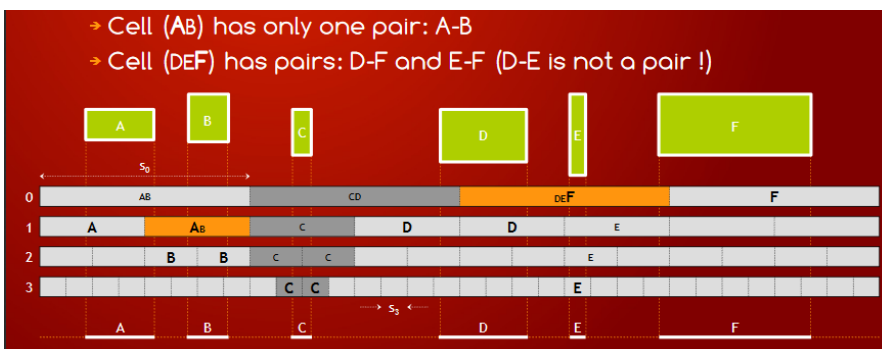


Do mriežky dokážeme objekty pridávať, odoberať z nej a updatovať ju, keď sa nejaký objekt pohne.

Takáto uniformná mriežka je jednoduchá na implementáciu a je rýchla pre particles (malé dynamické objekty) a statické prostredie. Nevieme však ako nájsť optimálnu veľkosť mriežky, kvôli rôznej veľkosti objektov. Ak dáme príliš hustú mriežku, výpočtová náročnosť práce v danej mriežke bude veľká. Veľké mriežky zaberú veľké množstvo pamäte a update je pomalý pri veľkých objektoch.

Hierarchická mriežka

Majme štvorúrovňovú rovnomernú mriežku s rozlíšeniami 2^k (každá nižšia úroveň je dvakrát hustejšia). Keď chceme vložiť objekt do mriežky, nájdeme jeho AABB obálku a vložíme ho čo najnižšie, kam sa podľa obálky zmestí (či sa zmestí sa rozhoduje podľa šírky obálky a šírky mriežky) a zapíšeme id tohto objektu do všetkých úrovní, nad ktorými sa nachádza. Najnižšia úroveň znamená pre objekt jeho rozlíšenie. Ak je v bunke zapísaných viac ako jedno id, tak sa zostavujú možné páry (kolidujúcich) dvojíc. Pár koliduje ak rozlíšenie mriežky je rovnaké ako rozlíšenie jedného z objektov.



Veľkosť bunky $s = 1/2^k$

$s_0: 1, s_1: 1/2, s_2: 1/4, \dots$

Rozlíšenia:

A: 2^1 , B: 2^2 , C: 2^3 ,

D: 2^1 , E: 2^3 , F: 2^0

Objekty dokážeme do hierarchickej mriežky pridávať, odoberať a updatovať. Za lineárny čas zvláda malé aj veľké dynamické objekty. Potrebuje ale viac pamäte a musí updatovať viac mriežok.

Mid Phase

Ide o spresnenie správnych párov, ktoré dostaneme z broad phase a zjednodušenie práce pre nasledujúcu narrow phase.

Techniky:

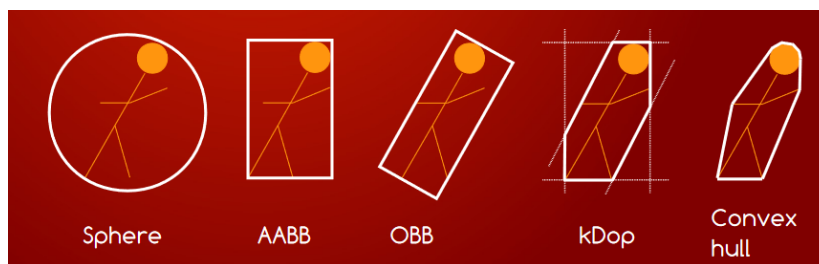
- zjednodušenie komplexnej geometrie do hierarchie obálok (Bounding Volume Hierarchies)
- dekompozícia nekonvexných objektov na konvexné časti

Táto fáza je dobrá pre paralelizáciu .

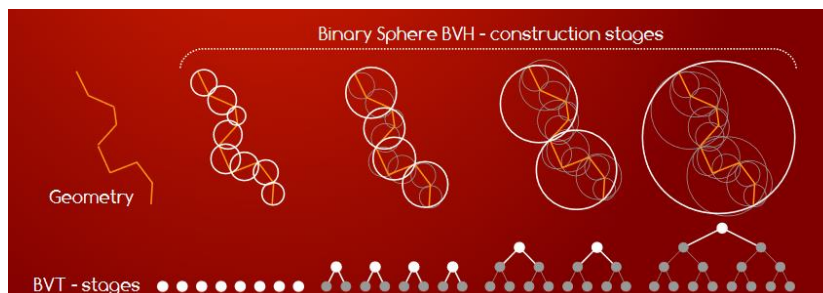
Hierarchia obálok (Bounding Volume Hierarchies)

Hierarchia obálok, tiež známa ako strom obálok, je vo všeobecnosti m-árny strom $T = \{T_1, \dots, T_m, BV, G\}$, ktorého vrcholy (T_i) obsahujú určitú obálku (Bounding Volume), ktorá pokrýva určitú časť geometrie objektu (G).

Každá nižšia úroveň stromu teda reprezentuje lepší odhad geometrie objektu. Deti vrcholu dokopy popisujú rovnakú časť geometrie ako ich rodič. Obaly by mali byť čo najmenšie (konvexné) časti také, aby pokryli celú geometriu objektu.



Stratégie na zostavenie hierarchie obálok sú Top-down BVT a Bottom-Up BVT. Top-down sa zostavuje postupne od koreňa, ktorý reprezentuje celý objekt až po listy, čo sú jednotlivé najmenšie časti. Bottom-up, naopak, začína v zostavovaní najprv listami a postupne buduje strom až ku koreňu.

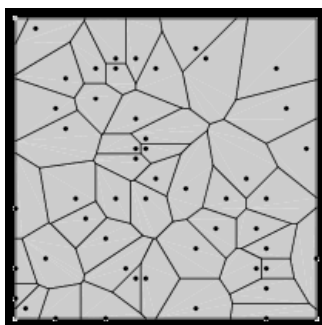


Voronoiove oblasti v kolízii

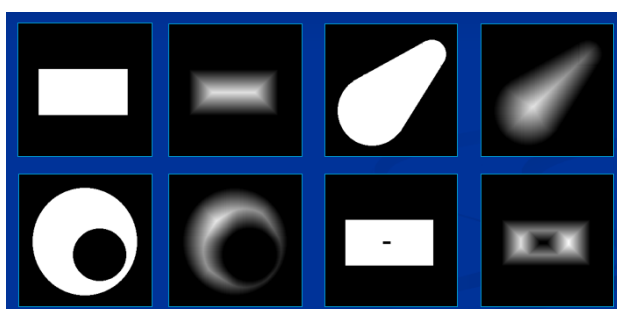
Kolízia guľa-kapsula

Na zistenie bodu, kde sa tieto dva objekty stretli potrebujeme spraviť projekciu stredu gule „C“ na smerovú os kapsule „a“.

Bod C a os a vieme zistiť pomocou Voronoiových oblastí (diagramu). Voronoiov diagram je rozdelenie roviny do n oblastí. Pre n rôznych bodov v rovine sa priradí oblasť, ktorá je najbližšia k danému bodu.



Ak si predstavíme, že z bodu C začneme radiálny rast do určitej šírky dostaneme guľu a ak na osi kapsule začneme radiálny rast do určitej šírky, dostaneme danú kapsulu. Táto os a bod sú kostrou pre dané objekty. Kostra sa dá získať pomocou distance transform, kedy postupujeme z kraja objektu do stredu a kde sa oblasti postupne stretnú vzniká kostra objektu (distance map).



(kostra je len biela časť)

Pre zistenie kontaktného bodu / normály potrebujem najprv zistiť danú projekciu guľi na kapsulu:

$$a = C_2 - C_1; u = \text{norm}(a); v = C - C_1; q = u \text{ dot } v$$

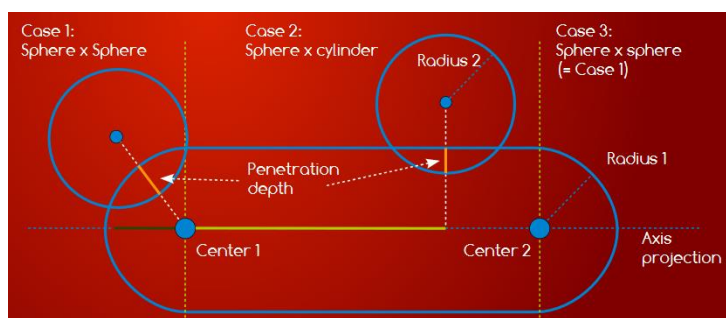
Kapsulu si rozdelíme na tri časti – dve guľe a valec. Prípady kedy $q < 0$ alebo $q > |a|$ budeme riešiť ako kolíziu guľa x guľa. Inak budeme riešiť kolíziu guľa x nekonečný valec:

$$\begin{aligned} Q &= c_1 + qu \\ m &= C - Q \\ n &= \text{norm}(m) \\ P_1 &= Q + r_1 n \\ P_2 &= C - r_2 n \end{aligned}$$

Kontaktný bod / normála teda bude:

$$\begin{aligned} p &= 0.5(P_1 + P_2) \\ n &= \text{norm}(P_2 - P_1) \end{aligned}$$

Hĺbka prieniku bude: $d = -|P_2 - P_1|$



Dekompozícia telesa na konvexné časti

Potrebujeme pre dané nekonvexné teleso nájsť také podčasti, ktoré sú skoro konvexné. Geometria je skoro konvexná, ak rozdiel medzi jej objemom a objemom konvexného obalu je pod určitým prahom. Takéto rozdelenie sa dá dosiahnuť aj pomocou Simple relaxation strategy (ACD):

- Zvolíme si top-down deliacu stratégiu (napríklad OBB)
- Rekurzívne delíme geometriu až po listy
- Použijeme objemový prah a zastavovacie kritériá
- Dostaneme (veľkú) množinu malých skoro konvexných podčastí
- Dáme podčasti do priority queue usporiadaných na základe ich objemu
- Popneme prvý prvok a snažíme sa ho zmergovať s ďalšou malou časťou. Dve malé časti sa spoja, ak spĺňajú objemový prah

Narrow Phase

Je to najužší výber párov dvojíc – fáza presnej detekcie kolízií.

V tejto fáze sa z daného zoznamu potencionálnych dvojíc kolidujúcich objektov vyberie presný čas a geometrické príznaky (body, hrany, steny), kde objekty navzájom kolidovali. Tiež zamietne tie páry objektov, ktoré nekolidujú.

Techniky:

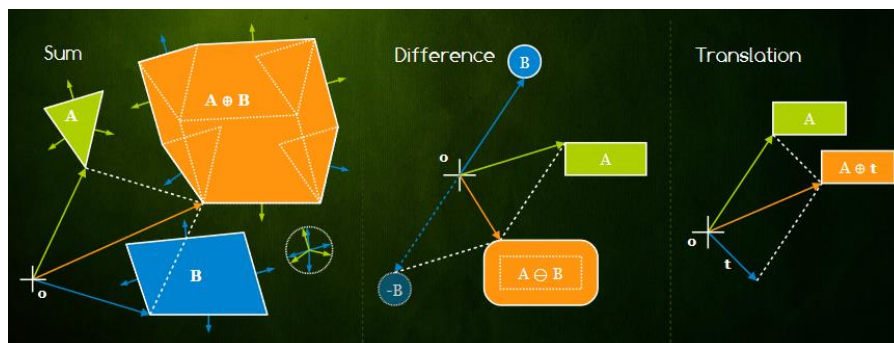
- Bounding volume hierarchy (AABB, OBB,...)
- Coherent feature tracking
- Signed distance map queries

Táto fáza je vhodná pre paralelizáciu.

Minkowského priestor

Teleso sa nazýva konvexné, keď je spojité a spojnice ľubovoľných dvoch bodov sa celá nachádza v danom telese. Pre dané dva konvexné telesá A a B vieme definovať Minkowského operácie a to súčet, rozdiel a posunutie ako:

- Minkowského súčet: $A \oplus B = \{a + b \mid a \in A \wedge b \in B\}$
- Minkowského rozdiel: $A \ominus B = A \oplus (-B) = \{a - b \mid a \in A \wedge b \in B\}$
- Minkowského posunutie: $A \oplus t = A \oplus \{t\} = \{a + t \mid a \in A\}$



Blízkosť konvexných telies

Dve konvexné telesá sa dotýkajú, ak ich prienik je podmnožina kontaktnej roviny β .
 $A \cap B \subset \beta$

Dotykový vektor t_{AB} medzi dvomi konvexnými telesami je najkratší vektor posunutia t , ktorý posúva telesá do vzájomného kontaktu.

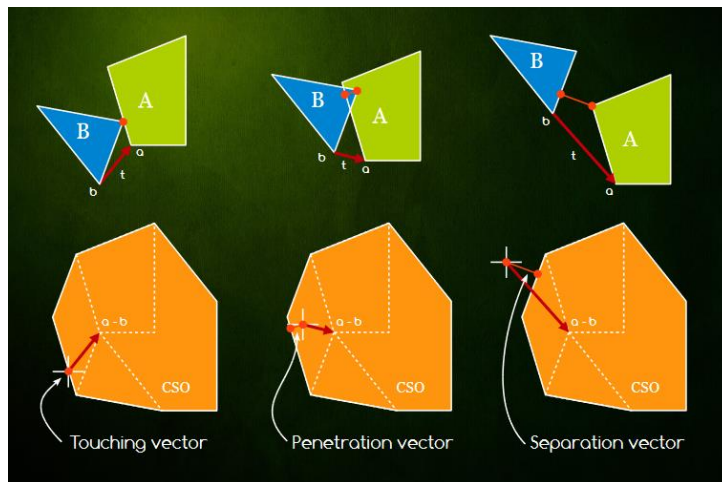
$$t_{AB} \in \{t \mid A \cap (B \oplus t) \subset \beta \wedge t \in \mathbb{R}^3 \wedge |t| = d_{AB}\}$$

Dotyková vzdialenosť d_{AB} je dĺžka dotykového vektora t_{AB} .

$$d_{AB} = \min\{|t| \mid A \cap (B \oplus t) \subset \beta \wedge t \in \mathbb{R}^3\}$$

Telesá sú v tesnej blízkosti, ak je ich dotyková vzdialenosť menšia ako zvolený prah.

Disjunktné (disjoint) telesá (ktoré majú disjunktný dotykový vektor / vzdialenosť) hovoríme, že majú rozdeľujúci vektor / vzdialenosť (separating vector). Ak sa telesá prekrývajú hovoríme, že majú prienikový vektor / hĺbku (penetration vector).



Ak sa telesá dotýkajú (t_{AB} je nula), ich prienik vytvára kontaktný región. Ak sa telesá prekrývajú alebo su disjunktné (t_{AB} je nenulové) kontaktný región sa vytvára nasledovne:

- Vypočítame dve pomocné množiny (množina bodov v konvexnom objekte, ktorá má minimálnu projekciu na smerovú os) pre teleso A a B
- Spravíme projekciu pomocných množín na dotykový vektor t_{AB} a zoberieme medián.
- Vytvoríme kontaktnú rovinu, kde medián bude stred a normála bude t_{AB}
- Spravme projekciu oboch pomocných množín na kontaktnú rovinu a zoberieme ich prienik ako kontaktný región

