

# DETEKCIA KOLÍZIÍ

---

doc. Ing. Branislav Sobota, PhD.  
Katedra počítačov a informatiky  
FEI TU Košice





# Kolízie vo virtuálnych scénach

- Detekcia kolízie (*collision detection*)
  - vznik kolízie – indikácia
  - forma kolízie
  - objekty kolízie
  - čas kolízie
- Reakcia na kolíziu (*collision response*)
  - správanie sa objektov podľa fyzikálnych zákonov
  - deformácia objektov
  - animačné efekty



# Modely objektov pre kolízie

- **Polygonálne**
  - *Štrukturované* (konvexné alebo nekonvexné (konkávne) polygóny )
  - *Neštrukturované* (polygóny, ktoré nie sú pospájané a nemajú informáciu o typológii modelu)
- **Nepolygonálne** (napr. výsledok určitej funkcie, implicitné resp. parametrické plochy)

Pre každý z týchto modelov existuje iná forma detekcie kolízie



# Algoritmy detekcie kolízie

## Podľa reprezentácie detekcie kolízie:

- *Detekčné*: reprezentácia iba pomocou logických hodnôt - true, false.
- *Určujúce*: reprezentácia pomocou priesečníkov, kolíznych čiar a podobne.

## Podľa počtu objektov, ktoré sa vyskytujú pri výpočte detekcie kolízie:

- *“N- Body system“* – Výpočet detekcie kolízie sa robí naraz so všetkými pohybujúcimi sa objektmi a zisťuje sa čas, kedy ku kolízie dôjde pri známej rýchlosti objektov.
- *“2-Body system“* – Výpočet sa robí len s dvoma objektmi bez ohľadu na to, či sa objekty pohybujú alebo nie.



# Fázy detekcie kolízie

1. Výber objektov na detekciu kolízie
2. Utriedenie objektov na detekciu kolízie. Vybratie vhodného algoritmu
3. Použitie vhodného algoritmu detekcie kolízie. Test vylúčenia
4. Výpočet kolízie medzi objektami
5. Indikácia detekcie kolízie



# Základné používané matematické vzťahy

- Veľkosť vektora

$$\|\mathbf{v}\| = \sqrt{v_x^2 + v_y^2 + v_z^2}$$

- Vzdialenosť dvoch vektorov

$$d = d(\mathbf{v}, \mathbf{u}) = \sqrt{(u_x - v_x)^2 + (u_y - v_y)^2 + (u_z - v_z)^2}$$

- Skalárny súčin vektorov

$$\mathbf{v} \cdot \mathbf{u} = \|\mathbf{v}\| \cdot \|\mathbf{u}\| \cos(\mathbf{v}, \mathbf{u}) = v_x u_x + v_y u_y + v_z u_z$$

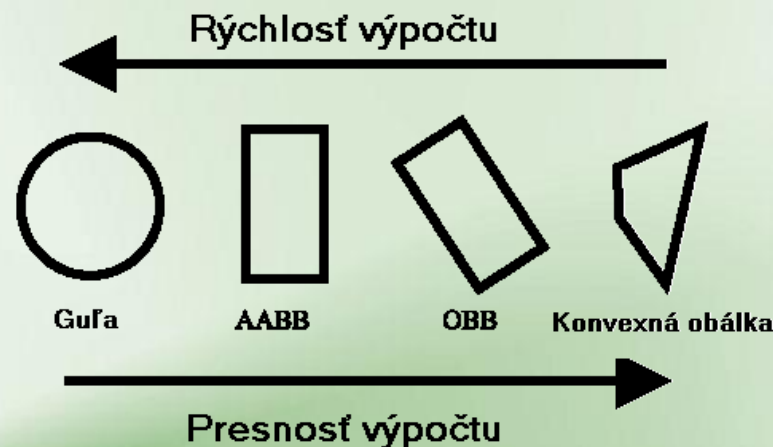
- Vektorový súčin vektorov

$$\mathbf{u} \times \mathbf{v} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ v_x & v_y & v_z \\ u_x & u_y & u_z \end{vmatrix} = [a_y b_z - a_z b_y, a_z b_x - a_x b_z, a_x b_y - a_y b_x]$$



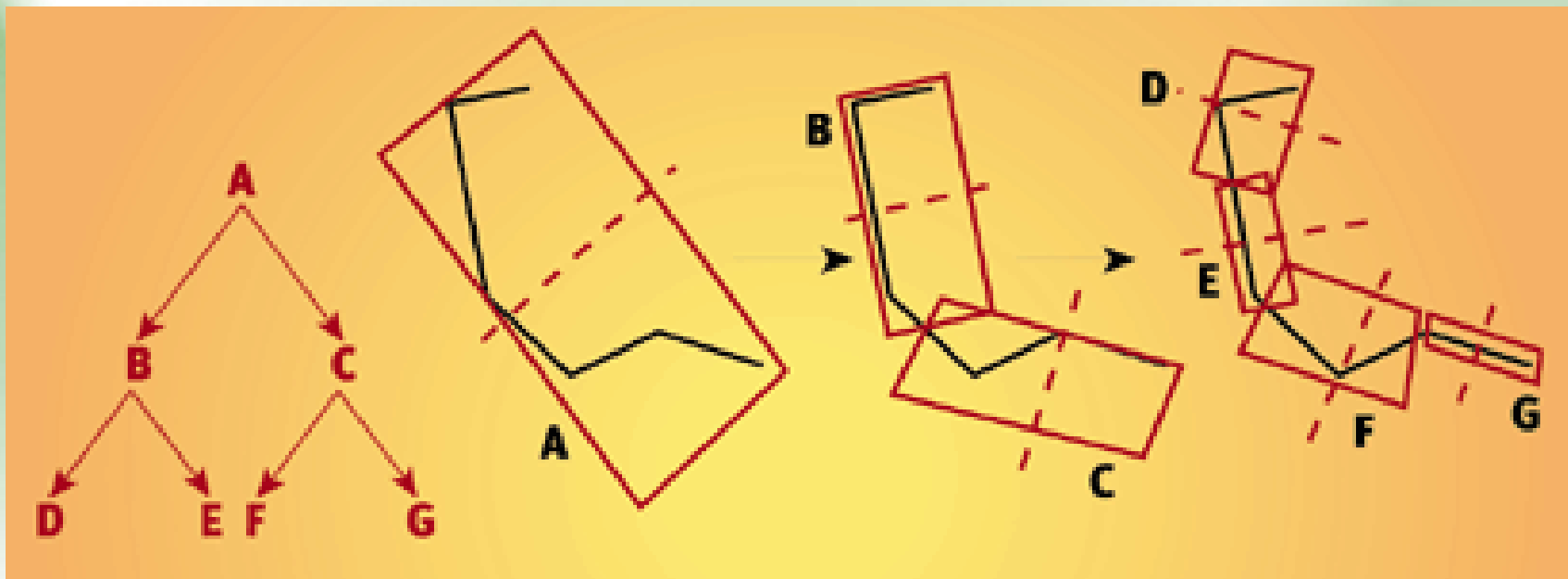
# Typy algoritmov riešenia kolízie

- **Hierarchia guľí** - Je tvorená iba stredovým bodom a polomerom.
- **Osovo - orientovaný kváder (AABB)** – Je tvorený minmaxovými hodnotami na osiach X,Y,Z alebo stredovým bodom a troma vektormi určujúce vzdialenosť stien od stredového bodu (extend).
- **Objektovo - orientovaný kváder (OBB)** – Tvorený šiestimi polygónmi pevne zviazaný s objektom, ktorý ohraničuje. Je reprezentovaný maticou orientácie, stredovým bodom a troma vektormi určujúce vzdialenosť stien od stredového bodu (extend).
- **Konvexná obálka (Convex Hull)** – Najzložitejšia obálka. Tvoria ho konvexné polygóny, ktoré čo najvierohodnejšie obklopujú zložitý objekt.





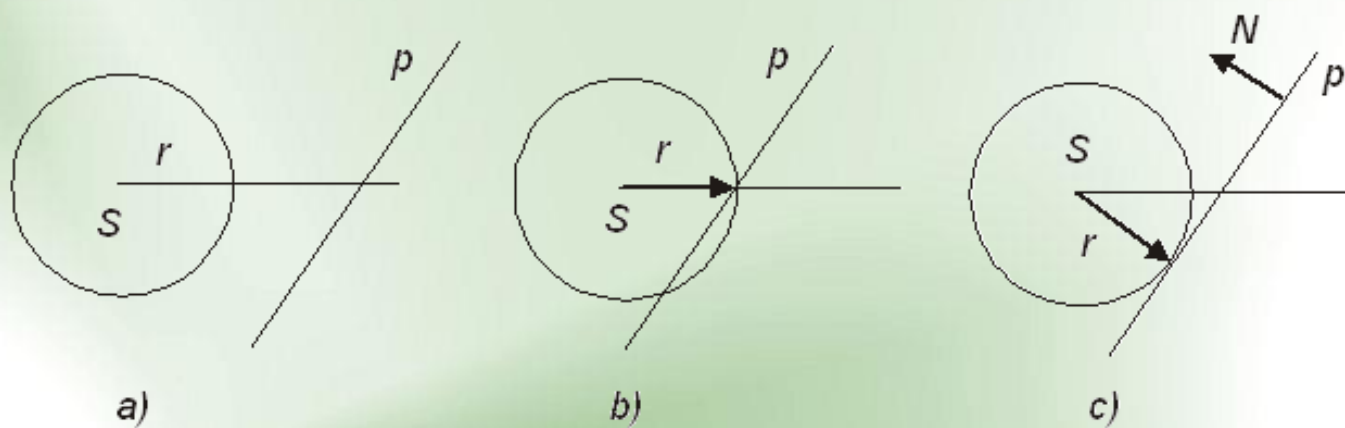
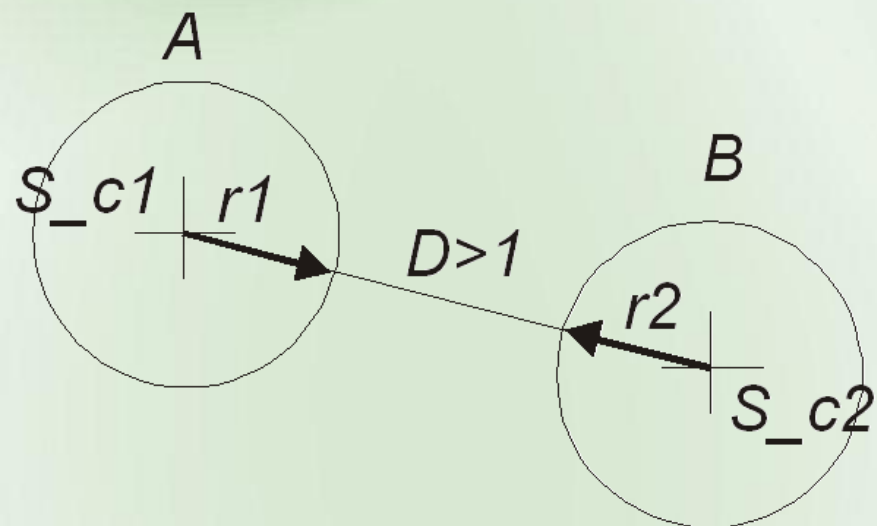
# System hraničných obálok riešenia kolízie





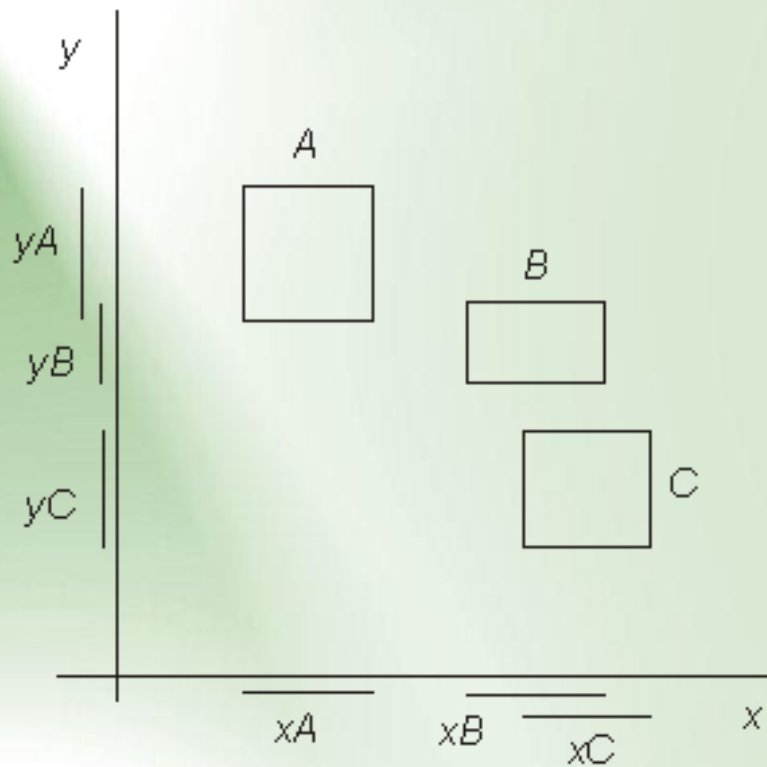


# Hierarchia gulí

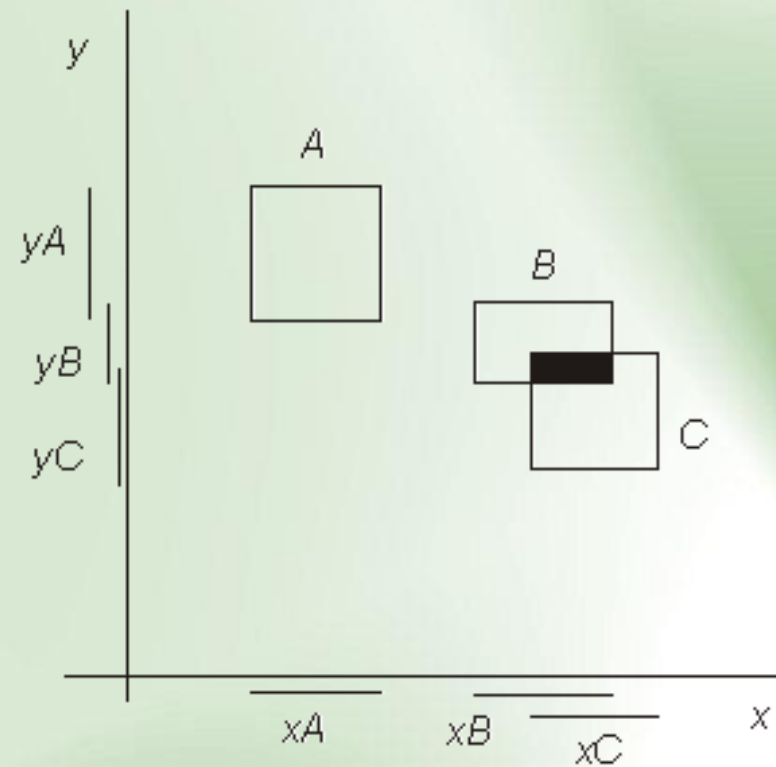




# AABB - Axis Aligned Bounding Box



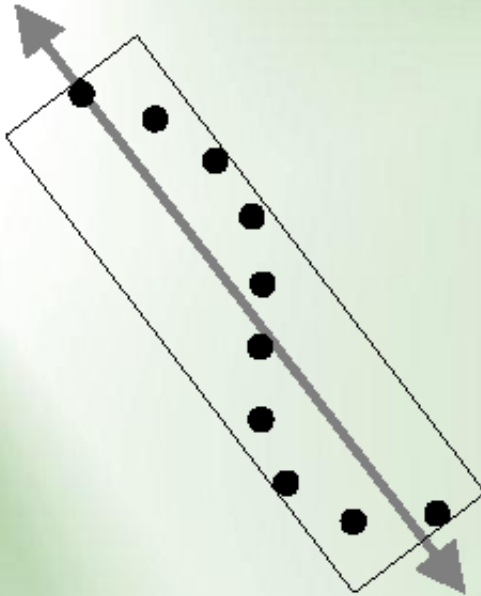
a)



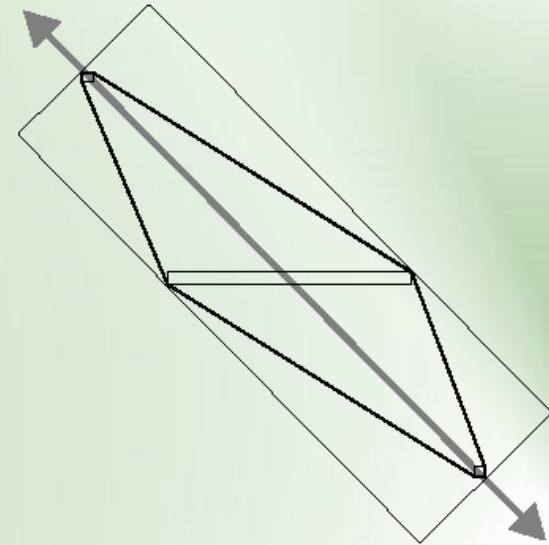
b)



# OBB - *Object Oriented Bounding Box*



Konvariančná metóda. Model bol aproximovaný na sled vrcholov, sivo je označený stredový vrchol. Zistením min. a max. hodnôt osí modelu dostaneme OBB obálku.



V ďalšom kroku degenerujeme konvexnú obálku do formy kvádra. Rozšírime tie hrany, ktoré sa nedotýkajú modelu. Použitím konvexnej obálky môžeme dostať najtesnejšie spojenie. Neskôr degradujeme túto obálku na OBB strom



## Dôvod riešenia kolízií

- Detekcia kolízie a následné zastavenie pohybu
- Detekcia pohybu a následná zmena trajektórie pohybu
- Aby sa neprechádzalo kamerou cez objekty scény
- Využíva sa pri senzorickej systéme na vykreslenie textu na obrazovku pri vstupe do senzorickej oblasti



## Typy senzorov (trigerov)

- **vertikálny senzorový polygón** – určený na detekciu z pohľadu zobrazovača s napojením na definovaný typ ukazovacieho zariadenia, ktoré používa pozorovateľ (napr. myš alebo dátová rukavica). Po zistení polohy a stlačení definovaného aktivačného prvku (napr. tlačidla myši sa napr. zobrazia príslušné informácie)
- **horizontálny senzorový polygón** – určený na detekciu polohy kamery a aktivuje sa automaticky pri zistení kolízie. Jedná sa v princípe o polohový senzor.
- **pohľadový senzor** – určený na detekciu ihlana pohľadu kamery a aktivuje sa v prípade ak sa objekt dostane do zorného poľa kamery. V tomto prípade je nutné ešte detekovať najbližší objekt k osi pohľadu, pretože v ihlane pohľadu sa môže nachádzať viacero objektov.

# OTÁZKY ?

---

