

Detectarea de coliziuni între cercuri

- Quadrees -

- Facultatea de Matematică și Informatică -
Universitatea Transilvania din Brașov

June 19, 2022

- 1 Coliziuni - definiție și paradigmă
- 2 Coliziuni - probleme și soluții
- 3 Quadrees - date generale și implementare
- 4 Aplicație - implementare
- 5 Concluzii

- 1 Coliziuni - definiție și paradigmă
- 2 Coliziuni - probleme și soluții
- 3 Quadrees - date generale și implementare
- 4 Aplicație - implementare
- 5 Concluzii

- 1 Coliziuni - definiție și paradigmă
- 2 Coliziuni - probleme și soluții
- 3 Quadrees - date generale și implementare
- 4 Aplicație - implementare
- 5 Concluzii

- 1 Coliziuni - definiție și paradigmă
- 2 Coliziuni - probleme și soluții
- 3 Quadrees - date generale și implementare**
- 4 Aplicație - implementare
- 5 Concluzii

- 1 Coliziuni - definiție și paradigmă
- 2 Coliziuni - probleme și soluții
- 3 Quadrees - date generale și implementare
- 4 Aplicație - implementare**
- 5 Concluzii

- 1 Coliziuni - definiție și paradigmă
- 2 Coliziuni - probleme și soluții
- 3 Quadrees - date generale și implementare
- 4 Aplicație - implementare
- 5 Concluzii

Coliziuni - definiție și paradigmă

Definiție:

- Ciocnire de forțe

Coliziuni - definiție și paradigmă

Definiție:

- Ciocnire de forțe
- Tangența dintre obiecte care se deplasează unul spre altul

Coliziuni - definiție și paradigmă

Context:

În domeniu informatic, un sistem de detectare a coliziunilor poate ajuta la:

- Realizarea de simulări de sisteme fizice

Coliziuni - definiție și paradigmă

Context:

În domeniu informatic, un sistem de detectare a coliziunilor poate ajuta la:

- Realizarea de simulări de sisteme fizice
- Realizarea de jocuri video

Probleme:

- Complexitatea verificării coliziunilor dintre un obiect oarecare și oricare altul aflat în același spațiu

Probleme:

- Complexitatea verificării coliziunilor dintre un obiect oarecare și oricare altul aflat în același spațiu
- Această complexitate devine pătratică, iar pentru multe obiecte, devine ineficientă

Soluții:

- Partiționarea spațiului populat de obiecte în sub-spații

Soluții:

- Partiționarea spațiului populat de obiecte în sub-spații
- Oricare doua obiecte aflate în sub-spații diferite nu pot fi tangente, rezultând în faptul că verificarea coliziunii nu mai este necesară între acestea două

Soluții:

- Partiționarea spațiului populat de obiecte în sub-spații
- Oricare doua obiecte aflate în sub-spații diferite nu pot fi tangente, rezultând în faptul că verificarea coliziunii nu mai este necesară între acestea două
- Complexitatea se reduce la a verifica pentru fiecare careu coliziunea dintre obiectele care îl populează

Quadtrees - date generale și implementare

Date generale:

- Este o structură de date care codifică un spațiu bidimensional în celule adaptabile / ajustabile

Quadtrees - date generale și implementare

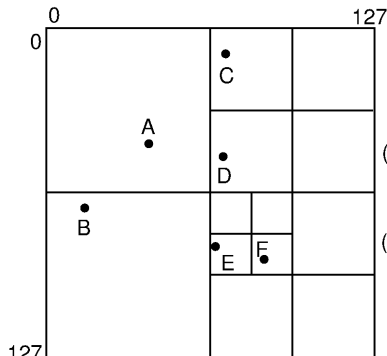
Date generale:

- Este o structură de date care codifică un spațiu bidimensional în celule adaptabile / ajustabile
- Se reprezintă grafic prin celule care partiționează un spațiu dat

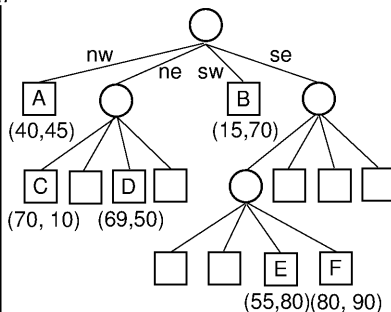
Quadtrees - date generale și implementare

Date generale:

- Este o structură de date care codifică un spațiu bidimensional în celule adaptabile / ajustabile
- Se reprezintă grafic prin celule care partiționează un spațiu dat



(a)



(b)

Inserarea:

- Se verifică dacă ceroul de inserat intră în careul curent, iar dacă da, aici se încheie procesul

Inserarea:

- Se verifică dacă cercul de inserat intră în careul curent, iar dacă da, aici se încheie procesul
- Dacă se poate realiza inserarea, cercul este adăugat în vectorul de cercuri

Inserarea:

- Se verifică dacă cercul de inserat intră în careul curent, iar dacă da, aici se încheie procesul
- Dacă se poate realiza inserarea, cercul este adăugat în vectorul de cercuri
- Pentru orice cerc din careul curent, se aplică o verificare a coliziunii cu cercul tocmai inserat

Împărțirea:

- Orice careu poate fi împărțit în alte 4

Împărțirea:

- Orice careu poate fi împărțit în alte 4
- Laturile careurilor noi sunt egale cu jumătate din cele ale careului inițial

Împărțirea:

- Orice careu poate fi împărțit în alte 4
- Laturile careurilor noi sunt egale cu jumătate din cele ale careului inițial
- După partiționare, trebuie realizată o operație de inserare pentru reaşezarea cercurilor în noile careuri

Ștergerea:

- Parcurgând fiecare nod al arborelui, verificăm dacă printre cercurile cuprinse de acesta există cercul căutat

Ștergerea:

- Parcurgând fiecare nod al arborelui, verificăm dacă printre cercurile cuprinse de acesta există cercul căutat
- Dacă da, este șters din vectorul de cercuri

Câmpurile unui obiect:

- Orice obiect este definit de o rază, o viteză, accelerație și de coordonatele centrului

Câmpurile unui obiect:

- `createObject` - creează un obiect, definindu-i unghiul cu care traversează ecranul și conturul

Câmpurile unui obiect:

- Orice obiect este definit de o rază, o viteză, accelerație și de coordonatele centrului

Câmpurile unui obiect:

- `createObject` - creează un obiect, definindu-i unghiul cu care traversează ecranul și conturul
- `updateObject` - actualizează poziția în spațiu a obiectului

Funcția de `onUserCreate()` setează datele inițiale pentru lansarea aplicației:

- Creează un nou arbore

Funcția de `onUserCreate()` setează datele inițiale pentru lansarea aplicației:

- Creează un nou arbore
- Șterge toate datele vechi

Funcția de `onUserCreate()` setează datele inițiale pentru lansarea aplicației:

- Creează un nou arbore
- Șterge toate datele vechi
- Realizează un random seed care va fi folosit la generarea cercurilor

Funcția de `onUserCreate()` setează datele inițiale pentru lansarea aplicației:

- Creează un nou arbore
- Șterge toate datele vechi
- Realizează un random seed care va fi folosit la generarea cercurilor
- Generează 100 de obiecte noi, pe care le inserează în arbore

Funcția de `onUserUpdate()` actualizează constant datele:

- Face fundalul negru

Funcția de `onUserUpdate()` actualizează constant datele:

- Face fundalul negru
- Recalculează coordonatele cerurilor

Funcția de `onUserUpdate()` actualizează constant datele:

- Face fundalul negru
- Recalculează coordonatele cerurilor
- Desenează cercurile

Funcția de `onUserUpdate()` actualizează constant datele:

- Face fundalul negru
- Recalculează coordonatele cerurilor
- Desenează cercurile
- Desenează liniile dintre cercurile tangente

- Numărul de careuri influențează numărul de operații de verificare

- Numărul de careuri influențează numărul de operații de verificare
- Se pot folosi și alte structuri de date pentru detectarea și rezolvarea coliziunilor

- Numărul de careuri influențează numărul de operații de verificare
- Se pot folosi și alte structuri de date pentru detectarea și rezolvarea coliziunilor
- Pot apărea erori în detectare în cazul în care există disproporție mare între viteza cu care circulă un obiect și numărul de cadre pe secunda (FPS)

- Numărul de careuri influențează numărul de operații de verificare
- Se pot folosi și alte structuri de date pentru detectarea și rezolvarea coliziunilor
- Pot apărea erori în detectare în cazul în care există disproporție mare între viteza cu care circulă un obiect și numărul de cadre pe secunda (FPS)
- Cercurile și poligoanele convexe au metode mai simple de detectare a coliziunilor, pe când poligoanele convexe sunt mai greu de tratat