

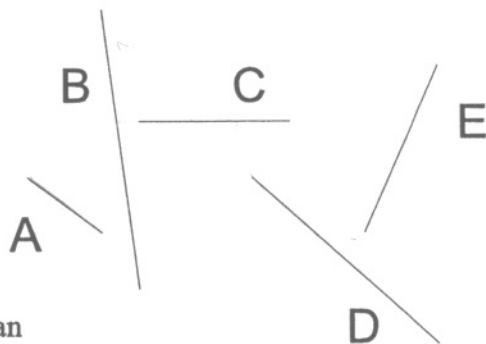
Tentissä ei saa käyttää kirjallisuutta, muistiinpanoja eikä laskimia.

1. Selitä lyhyesti seuraavat käsitteet ja perustele niiden hyödyllisyys peliohjelmoinnissa.

- Ohjaimen vastekäyrä (1,5 p)
- Octree (1,5 p)
- Käänteinen kinematiikka (1,5 p)
- Pelattavuustesti (1,5 p)

2. Kaksiulotteisessa tapauksessa BSP-puuhun tallennetaan kolmioiden sijaan janoja.

- Muodosta kuvan 1 janoista BSP-puu. Jos tarvitset janojen normaaleja, voit valita niiden suunnan kuten haluat. Esitä myös prosessi välivaiheineen, ei vain



Kuva 1

- lopputulosta (millintarkka sijainti ei ole tärkeää vaan muodostusperiaate). (2 p)

- Mitä hyötyä BSP-puusta on peleissä? (1,5 p)
 - Miten leafy-BSP eroaa tavallisesta BSP-puusta ja mihin sitä käytetään? (1,5 p)
 - Miksi BSP-puun käyttö vaikeuttaa dynaamisten peliympäristöjen luomista? (1 p)
3. a) Törmäystarkistus on luonnostaan kertaluokkaa $O(n^2)$. Mistä tämä johtuu ja millaisia ongelmia se voi aiheuttaa? Anna esimerkki. (2 p)
- b) Miten pelin toteuttaja voi lieventää edellä mainittua ongelmaa? Jatka edellisen kohdan esimerkkiä ja esitä suuntaa-antavia laskelmia parannusten suuruusluokasta. (2 p)
- c) Mitä tarkoitetaan integrointimenetelmillä pelifysiikassa? Mitä merkitystä niillä on? (2 p)
4. Pelintekijä oli toteuttanut pelinsä laajat ulkotilat käyttäen maailman tasalevyistä jakoa kolmioihin. Peli osoittautui hitaaksi, vaikka hänen käyttämänsä shaderit olivat hyvin yksinkertaisia ja tekoälykin oli vielä toteuttamatta.
- Mitä tarkoitetaan järjestelmä-, sovellus- ja mikrotason optimoinneilla? Mikä näistä olisi todennäköisesti oikea taso aloittaa pelin optimointi ja miksi? (2 p)
 - Tutkittuaan asiaa pelintekijä totesi tarpeelliseksi käyttää maaston piirrosta tarkkuustasoja (level of detail). Kuvaa jonkin tällaisen menetelmän pääperiaatteet. (3 p)
 - Tämän jälkeen hän huomasi, että järjestelmän pullonkaulana oli CPU:n ja 3d-kortin välinen väylä. Miten hän saattoi havaita tämän? Mitä ongelmalle voisi tehdä? (1 p)