

Rakennekuvaus

Ohjelma on rakenteeltaan entiteetti-komponentti-pohjainen (http://en.wikipedia.org/wiki/Entity_component_system). Mailat, pallot ja esteet ovat kaikki entiteettejä joiden toiminta määrittyy sopivien komponenttien avulla. Komponentteja ovat esimerkiksi PaikkaKomponentti, PaloPiirtoKomponentti tai TormaysAlueKomponentti jotka vastaavat paikan ylläpidosta, pallografiikan piirtämisestä ja törmäysalueen tiedon ylläpitämisestä. Komponentit ovat itsenäisiä eivätkä oleta tai tiedä muista komponenteista mitään.

Komponentit käyttävät entiteetissä majoilevaa viestijonoa keskinäisessä kommunikoinnissa. Jokin tapahtuma saattaa generoida viestin johonka toiset komponentit voivat reagoida jos viesti on niiden toimintaan liittyvä. Esimerkiksi InputKomponentti joka kuuntelee näppäinsyötettä generoi LiikeViestin joka kertoo mihin suuntaan halutaan liikkua. MailaNopeusKomponentti puolestaan reagoi tähän viestiin ja päivittää liikkeen suunnan oikeaksi; muut komponentit puolestaan eivät reagoi tähän viestiin.

Viestien käsittelyssä käytetään hyväksi vierailijamallia (http://en.wikipedia.org/wiki/Visitor_pattern); viesti ottaa vastaan ViestiVierailija-rajapinnan toteuttavan olion jonka vieraile-metodia viesti kutsuu käyttäen parametrinaan itseään. Abstrakti Komponentti-luokka toteuttaa ViestiVierailija-rajapinnan ja tarjoaa kaikille eri viestityypeille oletusvierailijatoteutuksen joka jättää viestin huomioimatta (esimerkiksi public void vieraile(MuutaNopeusViesti viesti) {}). Varsinaiset komponentti-implementaatiot voivat korvata haluamansa metodit joka mahdollistaa viesteihin reagoinnin.

Entiteetit luodaan tehdasmallilla (http://en.wikipedia.org/wiki/Factory_pattern) koska entiteetin luominen on kohtuullisen monimutkainen operaatio. Itse entiteettiobjektin lisäksi on luotava ja alustettava kaikki entiteettiin liittyvät komponentit. Tehdasmallilla tämän prosessin pystyy piilottamaan varsinaiselta ohjelmalta. Tehdas ottaa parametrinaan haluttavan entiteetin tyypin ja paikan ja palauttaa entiteetin. Heikkoutena ohjelmassa on tällä hetkellä se että tehdas ei voi alustaa esimerkiksi AI-mailaa täysin loppuun; AI-maila tarvitsee pallon paikkatiedon jotta se voi liikkua oikeaan suuntaan, mutta tehdas luo vain yksittäisiä objekteja. Tämä pakottaa manipuloimaan entiteettiä ja sen komponentteja tehtaan ulkopuolella mikä rikkoo kapsulointia (asetukset-luokka joka alustaa AI-mailat loppuun on tiedettävä mitä komponentteja entiteetti sisältää ja osattava käyttää näitä).

Asetukset-luokka alustaa pelin tilan luomalla ja alustamalla tarvittavat objektit. Lisäksi luokka toteuttaa NappulaKuuntelija-rajapinnan jonka avulla se kuuntelee käyttöliittymän nappuloiden tilamuutoksia (luo staattinen este, vaihda pelaajan kontrollia jne).

TormaysManagerin vastaa törmäysten valvomisesta. Jos entiteettien törmäysalueet ovat päällekkäin tai jos entiteetti on ulos ruudulta, generoi manageri sopivan viestin jonka se lähettää entiteetille. Entiteettien välinen törmäystarkistus on tehoton, $O(n^2)$, koska kaikki entiteetit tarkistavat törmäykset kaikkiin entiteetteihin (myös itseensä mutta tämä tapaus jätetään huomioimatta), mutta pienillä entiteettien määrillä tämä ei ole ongelma. Entiteettien törmäyskäsittelijäkomponentit jättävät törmäyksen huomioimatta jos on äskettäin törmätty jotta entiteetit eivät jäisi jumiin syvien törmäysten takia.

Käyttöliittymä on toteutettu javan omilla kirjastoilla ohjelmoinnin jatkokurssin oppeihin nojautuen.