

Ohjelman rakenne

App

Ohjelma ajetaan mainista, luomalla App - olio ja kutsumalla sen execute-metodia. Execute-metodi sisältää luopin, jossa kutsutaan päivitys- ja piirto metodeja. Nämä operaatiot on ulkoistettu Appin omistamalle EntityManager - oliolle. EntityManager kutsuu kaikkien Entity-olioiden update- ja draw-metodeja. Entity-luokan voi peria esimerkiksi luokka, joka toteuttaa Miinaharavan ruudukon logiikan. Entityt luodaan ja lisätään heti luomisen jälkeen EntityManagerille. EntityManager huolehtii niiden poistamisesta. App omistaa myös muita manager-olioita. Ne näkyvät alla olevassa kuvassa.

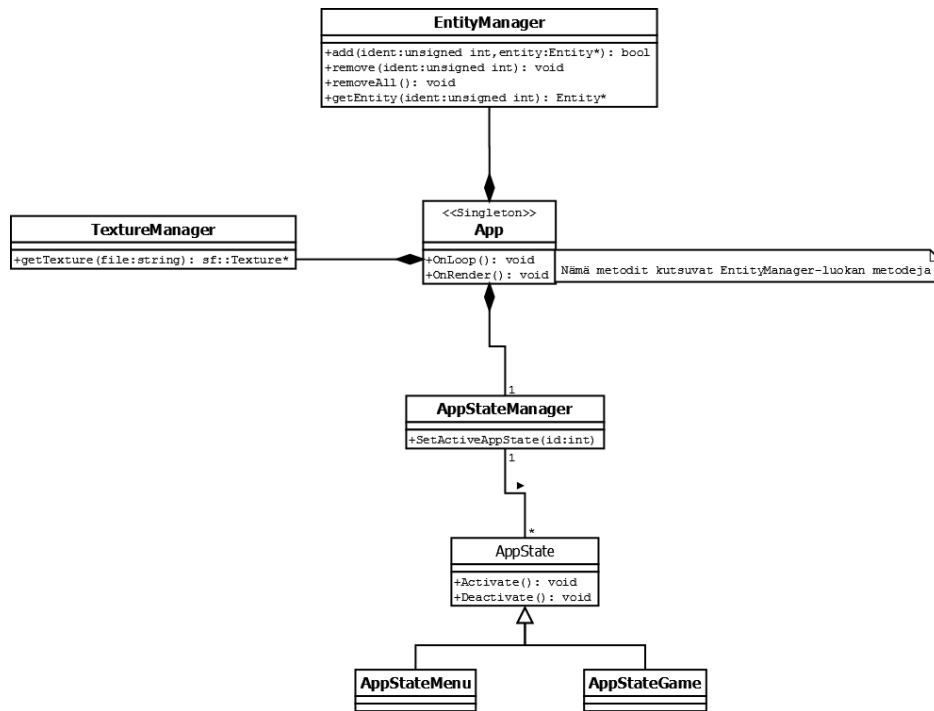


Figure 1: App-luokan rakenne

AppStateManager vastaa ohjelman tämänhetkisestä tilasta. Tila voi olla vaikkapa aloitusruutu, päävalikko tai pelitila. Kaikki ohjelman tilat periytyvät AppState-luokasta, joka toimii kuten rajapinta. TextureManager luokka huolehtii tekstuurien lataamisesta muistiin. Tekstuuria pyydetään oliolta. Jos se on jo muistissa, palauttaa olio osoittimen siihen. Jos sitä ei ole muistissa, olio lataa sen muistiin, ja sitten palauttaa osoittimen siihen.

Pelin rakenne

Pelin rakenne näkyy alla olevassa kuvassa. Pelin logiikka on jaettu GridBehavior-luokkaan ja TileDrawableSprite - luokkaan. GridBehavior luokka sisältää pelkästään logiikka, se ei piirrä mitään. Luokka siis sisältää pelkästään update-metodin. TileDrawableSprite on luokka, joka voi piirtää monta erilaista ruutua, riippuen mistä tilassa se on. Luokalla on siis pelkästään draw-metodi. Ruudun eri tilojen toteutukseen on kehitetty hyvin saman tyyppinen järjestelmä kuin Appille. Luokalla omistaa TileStateManager - olion, jolle klikkauslogiikka on ulkoistettu. Riippuen ruudun tilasta, samalla klikkauksilla voi olla erilainen lopputulos, joka on kätevästi toteutettu vastaavaan luokkaansa.

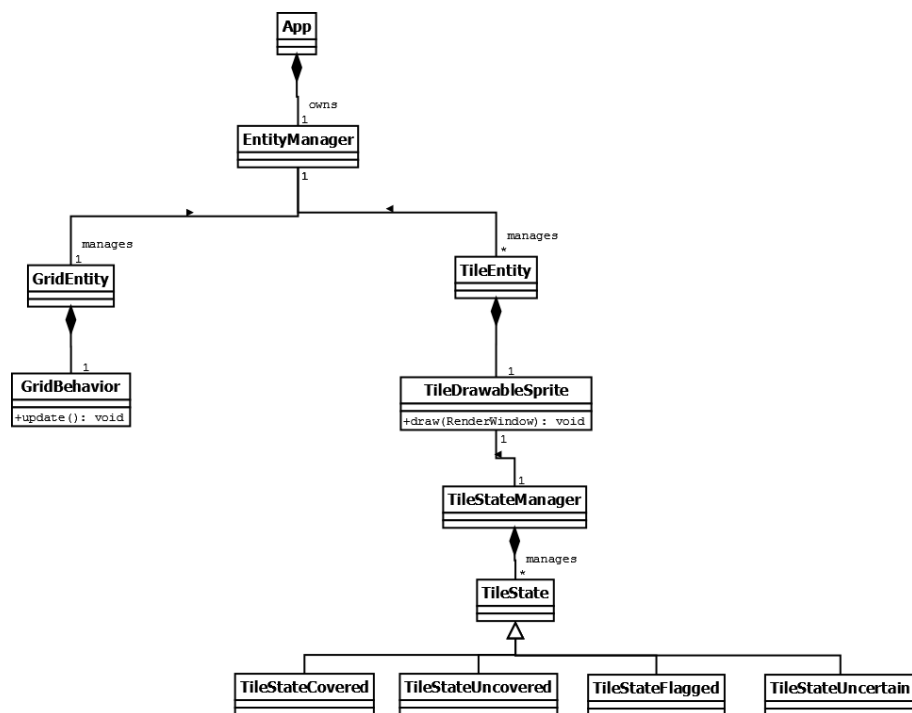


Figure 2: Pelin rakenne