

KEMI-TORNION AMMATTIKORKEAKOULU

Kiinteistökauppapelin suunnittelu ja toteutus

Laukkanen Lassi

Tietotekniikan koulutusohjelman opinnäytetyö
Ohjelmointi

Kemi 2011

TIIVISTELMÄ

Kemi-Tornion ammattikorkeakoulu, Tekniikan ala	
Koulutusohjelma	Tietotekniikka
Opinnäytetyön tekijä	Lassi Laukkanen
Opinnäytetyön nimi	Kiinteistökauppapelin suunnittelu ja toteutus
Työn laji	Opinnäytetyö
päiväys	23.12.2010
sivumäärä	36 + 9 liitesivua
Opinnäytetyön ohjaaja	DI Marko Heikkilä

Opinnäytetyön tavoitteina oli kiinteistökauppapelin suunnittelu sekä sen toimintojen toteutus ja testaus. Lisäksi tavoitteena oli opiskella erilaisia ohjelmistosuunnittelumenetelmiä, C++-ohjelmointikieltä ja Qt-kehitysympäristöä.

Opinnäytetyöprojekti käynnistettiin kehitysympäristön ja ohjelmointikielen itsenäisellä opiskelulla. Opiskelun kautta edettiin pelin vaatimusmäärittelyyn, suunnitteluun, ohjelmointiin ja testaukseen. Ohjelmistosuunnittelussa käytettiin apuna käyttöliittymäkaavioita ja UML-mallinnusta.

Kiinteistökauppapelin päätoiminnot ovat pankkiasiointi, kiinteistömarkkinat ja henkilöstön palkkaaminen. Ohjelmointityön laajuus rajattiin aluksi näihin kolmeen edellä mainittuun osioon, mutta jatkokehitykselle jätettiin mahdollisuus. Uusille alkuperäisestä suunnitelmasta poikkeaville toiminnoille luotiin suunnitelmat ohjelmoinnin yhteydessä, jos siihen oli tarvetta. Yksi tämänkaltainen toiminto oli kiinteistöaiheiset tehtävät. Projektin laajuutta pidettiin tarkkailussa, jotta työn määrä ei kasva missään vaiheessa liian suureksi.

Päätavoitteet saavutettiin hyvissä ajoin ja projekti sai runsaasti uusia toimintoja, jotka olivat lopulta erittäin tärkeitä itse pelille. Uusien ominaisuuksien myötä pelin kehitys esittelyvaiheeseen vei runsaasti aikaa, mutta opintojen vuoksi sitä oli varattu projektille tarpeeksi. Peli oli esittelyvalmis suunnitellun aikataulun mukaisesti.

Peli pysyy valmistumisen jälkeen ainoastaan yksityisessä käytössä. Sen alkuperäinen tarkoitus oli opettaa kirjoittajalleen laajemman projektin käsittelyä. Paljon työtä sisältävä projekti vaatii erittäin paljon suunnittelua, ohjelmointia ja harkintakykyä, joita tässä työssä harjoitettiin.

Asiasanat: peli, ohjelmointi, qt-kehitysympäristö, c++.

ABSTRACT

Kemi-Tornio University of Applied Sciences, Technology	
Degree Programme	Information Technology
Name	Lassi Laukkanen
Title	Business game designing and execution
Type of Study	Bachelor's Thesis
Date	23 December 2010
Pages	36 + 9 appendices
Instructor	Marko Heikkilä (M.Sc)

Programming Computer Game with Qt-Creator

The main goal of the thesis was to create real estate business game through designing, programming and testing its features in practice. Secondary goals were to learn different programming methods, C++-programming and Qt-development environment.

Thesis project started by studying development environment and programming language. After studies project led to demand specifications, designing, programming and testing. User interface diagrams and UML-models were used for help with software designing.

Real estate business game main functions are banking, real estate market and staff management. Programming was first limited to these three main parts, but a chance for further development was kept possible. If needed, plans for new components were created during programming. One of these kind functions was real estate tasks. There was continuous observation on project wideness, in order to prevent overwork.

Main goals were early achieved and project got several new components and functions, which were important for the final game. Adding new components to the game took a lot of time, but fortunately there was enough time to get the game ready for presentation before deadline.

The game will be only for self use. Original purpose of the project was to teach to its maker how to handle larger projects. This project demanded lot of designing, programming skills and discretion which were practiced during this project.

Keywords: game, programming, qt-framework, c++.

SISÄLLYSLUETTELO

TIIVISTELMÄ.....	I
ABSTRACT	II
SISÄLLYSLUETTELO.....	III
KÄYTETYT MERKIT JA LYHENTEET	V
1. JOHDANTO.....	1
2. PELIN VAATIMUKSET	2
2.1. Ympäristö.....	2
2.2. Toiminnalliset vaatimukset.....	2
2.2.1. Käyttötapauskaavio.....	3
2.3. Rajoitukset	4
2.4. Jatkokehitys.....	4
3. PROJEKTIN JA PELIN SUUNNITTELU	5
3.1. Visio.....	5
3.2. Projektin aikataulusuunnittelu	6
3.3. Käyttöliittymän suunnittelu	6
3.3.1. Pankkiosion suunnittelu.....	8
3.3.2. Kiinteistöosion suunnittelu	8
3.3.3. Henkilöstöosion suunnittelu	10
3.3.4. Tehtäväosion suunnittelu.....	11
3.4. Pelin toiminnallisuuden ja rakenteen suunnittelu	12
3.4.1. Tilakaavio	12
3.4.2. Sekvenssikaavio	13
3.4.3. Luokkakaavio	15
4. PELIN TOTEUTUS.....	16
4.1. Qt.....	16
4.2. C++	17
4.3. Ohjelmointi.....	17
4.3.1. Pankki.....	17
4.3.2. Kauppa	19
4.3.3. Kiinteistöt	21
4.3.4. Henkilöstö	23
4.3.5. Tehtävät.....	25
4.3.6. Tilastot	27
4.3.7. Manuaali.....	27
4.3.8. Peli-ikkuna.....	28
4.4. Elementtien lisääminen.....	29
4.5. Muutokset	29
5. TESTAUS.....	31
5.1. Merkittävimmät ongelmat	31
5.1.1. Signaalit ja slotit.....	31
5.1.2. Muistitoiminto	32
5.1.3. Todennäköisyyslaskurit	33
5.2. Systemi- ja hyväksyntätestaus.....	33
6. YHTEENVETO.....	34

7. LÄHDELUETTELO	35
8. LIITELUETTELO	36

KÄYTETYT MERKIT JA LYHENTEET

Qt	Ohjelmistojen ja käyttöliittymien kehitysympäristö
widget	Qt:n nimike eri käyttöominaisuuksille
QList	Qt:n muistitoiminto
QMap	Qt:n muistitoiminto

1. JOHDANTO

Tämän opinnäytetyön aiheeksi valittiin suuren kiinnostuksen johdosta peliohjelmointi. Projekti sisälsi runsaasti itsenäistä opiskelua, ohjelmointia ja dokumentointia. Jokaiset osat alueet opettivat uusia asioita kirjoittajalle.

Opiskeluun kuuluivat ohjelmointikielen ja kehitysympäristön omaksuminen. Ohjelmointi tuntemattomalla kehitysympäristöllä oli suhteellisen haastavaa mutta sitäkin opettavaisempaa. Dokumentointiin käytettiin apuna erilaisia suunnitteluohjelmia ja -kaavioita, jotka vaativat myös opettelua.

Kiinteistöpelit oli tarkoitettu yksityiseen käyttöön. Projektissa käytettiin ainoastaan ilmaisohjelmia, jotka asennettiin henkilökohtaiselle tietokoneelle. Kaikki resurssit mahdollisia oppikirjoja lukuun ottamatta löytyivät projektiin käytettävältä tietokoneelta. Opiskeluun ja yleiseen tiedonhakuun käytettiin mm. keskustelupalstoja sekä E-kirjoja. Tietoa haettiin pääasiassa asiakohtaisesti, joten työ eteni opettellessa.

Peli ei ole aivan nykypelien tasoa laajuudeltaan, toiminnoiltaan eikä ulkoasultaan. Peli keskittyy liiketoimintaan ja siinä käsitellään runsaasti eri lukuarvoja ja muuttujia, joten loogiset toimenpiteet ovat pääosassa. Projektille asetettiin realistinen päämäärä laajuutensa vuoksi, mutta ajan salliessa lisäkehitystä suoritettiin.

2. PELIN VAATIMUKSET

Kiinteistökauppapeli ohjelmoitiin C++ -kielellä Qt-kehitysympäristössä. C++ oli osittain tuttu kirjoittajalle, mutta ohjelmointia kirjoittaja ei ollut sillä tehnyt. Qt oli myös täysin uusi kehitysympäristö kirjoittajalle, eli sitä kirjoittaja ei ole aikaisemmin opiskellut eikä käyttänyt. Peli suunniteltiin yksityiseen käyttöön Windows-pohjalle, joten käännökset muille alustoille eivät kuuluneet suunnitelmiin. Projektin suunnitelmat pidettiin kuitenkin avoimena ja aikaa projektille varattiin paljon, joten erilaiset ohjelmointiratkaisut olivat kokeiltavissa.

2.1. Ympäristö

Uuden kehitysympäristön vuoksi pelille asetettiin realistiset vaatimukset. Qt-kehitysympäristö eli Qt-creator oli uusi opiskelualue, eikä ollut varmaa, soveltuiko se peliohjelmointiin. Siksi peli-ideana syntyi graafisen käyttöliittymän omaava tietokonepeli, joka käsittelee yksinkertaisia toimintoja pelaajan valintojen mukaan.

Qt-creator sisältää ns. widget-kirjaston, jossa widget voi tarkoittaa esimerkiksi tekstiä, nappia, liukupalkkia tai jopa sovellusikkunaa. Widgettejä on useita ja jokainen on muokattavissa. Liikkuvaa grafiikkaa pelissä ei esiinny, mutta vuorovaikutusta Qt-creatorissa tunnettujen widgettien kanssa on mukana.

2.2. Toiminnalliset vaatimukset

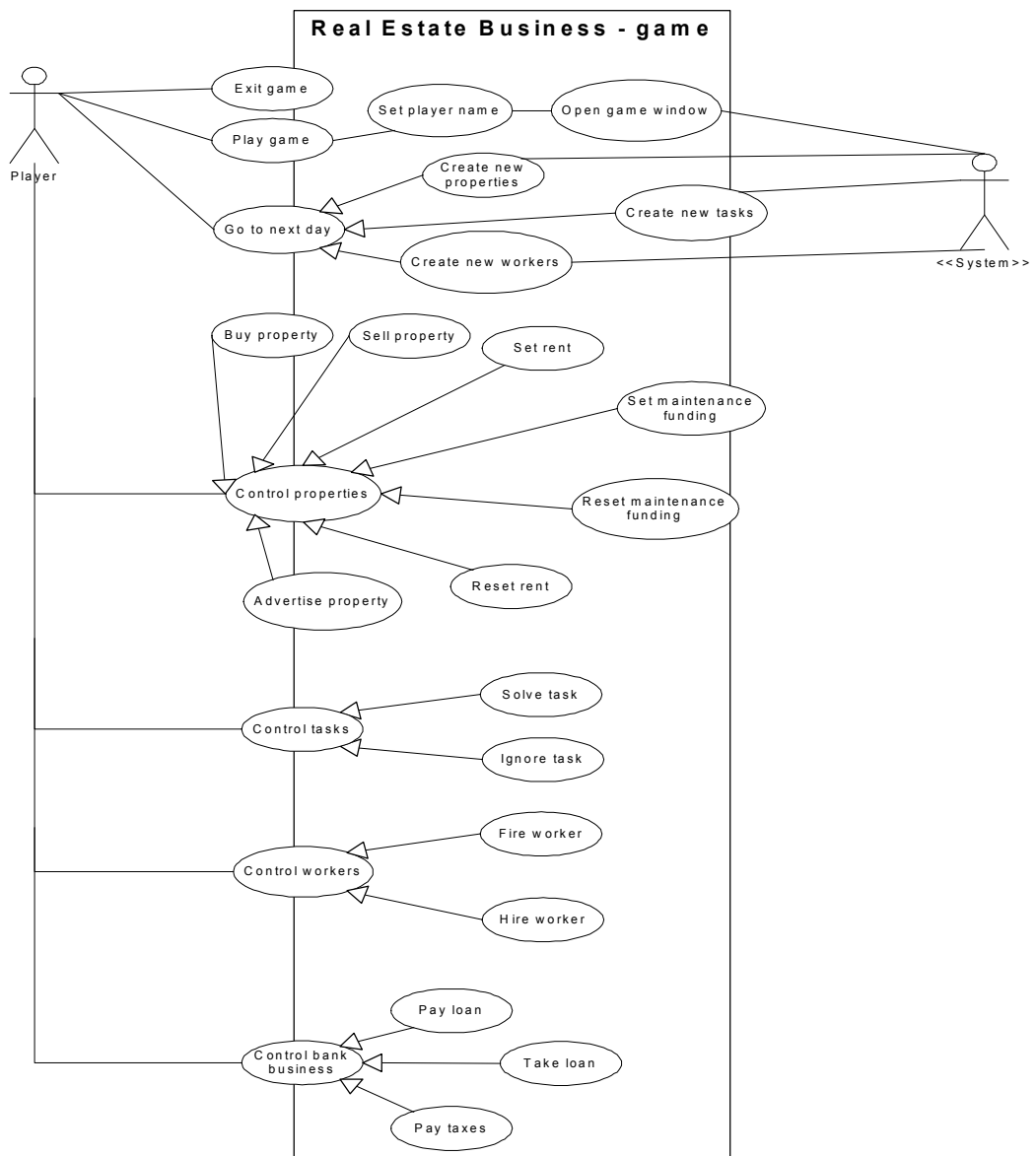
Peli keskittyy kiinteistökauppoihin ja se sisältää pankkiasioinnin, kiinteistömarkkinat ja henkilöstön ylläpidon. Pelin tavoitteena on ansaita enemmän mitä menettää, jotta kasvu on mahdollista. Pankkiasiointiin pelaajalle annetaan mahdollisuus nostaa ja maksaa lainaa. Kiinteistöille pelaaja voi ostamisen ja myymisen lisäksi asettaa vuokran ja huoltorahan sekä mainostuksen. Henkilöstön palkkaaminen ja erottaminen taas vaikuttaa positiivisesti ja negatiivisesti kiinteistöihin.

Mielenkiintoisuutta peliin lisättiin kiinteistöaiheisilla tehtävillä, jotka olivat ensisijaisena päämääränä päätoimintojen luomisen jälkeen. Kiinteistötehtävien ratkaiseminen on pelissä vapaaehtoista ja lopputulos riippuu pelaajan palkkaamien työntekijöiden ominaisuuksista. Työtehtävän voi ratkaista tai jättää tekemättä. Kummallakin valinnalla on seurauksensa.

Neljä suunniteltua osiota antoivat hyvän yleiskuvan koko pelille. Pelaaja aloittaa liiketoimintansa ottamalla rahaa pankista. Näillä rahoilla ostetaan kiinteistö, jolle asetetaan vuokra ja huoltoraha. Vuokran suuruuden ja huollon määrän perusteella asukkaita muuttaa kiinteistöön ja pelaaja alkaa periä vuokraa. Asetetun huoltorahan perusteella kiinteistössä ilmenee ajoittain ongelmia, joiden hoitamiseen pelaajan tulee palkata henkilöstöä. Näiden eri osioiden hallinnasta pelaajan toimia ylläpitää info-osio, joka auttaa pelaajaa seuraamaan pelin kulkua.

Pelin toiminnot ovat täysin riippuvaisia pelaajan valinnoista. Toiminnallisesti pelissä ei ole jatkuvaa prosessia ajossa, vaan pelaaja saa itse päättää, miten ja milloin peli etenee. Pelaajan käynnistäessä toiminnon ohjelma käy sille ohjelmoidut funktiot läpi ja näyttää pelaajalle toimintojen seuraukset. Pelin liiketoimintaan keskittyvä suunta on paljon lukujen ja erilaisten muuttujien käsittelyä, joten peli ei vaadi suurta suorituskykyä grafiikan puutteen vuoksi.

2.2.1. Käyttötapauskaavio



Kuva 1. Käyttötapauskaavio

Pelin käyttäytymistä kuvataan käyttötapauskaaviolla (Kuva 1). Kaaviosta näkee pelaajan (Player) käyttöön annetut toiminnot. Kiinteistöjen, tehtävien, henkilöstön ja pankin eri toiminnot ovat pelaajasta täysin riippuvaisia, joten lähes kaikki toiminnot aktivoituvat vain pelaajan käskystä. Pelin sisäiselle järjestelmälle (System) jää ainoastaan uusien tehtävien, kiinteistöjen ja työnhakijoiden luonti. Nämäkin luodaan vain, kun pelaaja päättää siirtyä pelipäivän eteenpäin. Käyttötapauskaavio on liitteenä 9.

2.3. Rajoitukset

Peli valmistui suunnitelmien mukaan ainoastaan yksityiseen käyttöön ja vain Windows-pohjalle. Projektille varatun ajan oli tarkoitus riittää ohjelmoimaan pelille määritetyt päätoiminnot ja ajan salliessa luomaan lisätoimintoja. Tavoitteita ei asetettu kerralla liikaa, jotta aikataulussa pysyminen oli mahdollista. Liian suuri lisäys jätettiin ohjelmoimatta, jos se uhkasi ylittää aikataulun, mutta jokainen idea kirjoitettiin muistiin myöhempää käyttöä varten.

Pelin suorittamiseen vaaditaan Qt-creator, sillä peliä ei luotu suoritettavaksi sovellukseksi. Pelille käytetyt ohjelmointikirjastot ovat kaikki Qt-creatorin sisäisiä, joten Qt-creatorin uusien versio mahdollistaa pelin ajamisen. Peli kuitenkin sisältää hieman grafiikkaa, joka ei näy ilman pelille tarkoitettuja kuvatiedostoja. Pelin jakamiseen olisi suositeltavaa kopioida koko projektikansio toiminnan takaamiseksi. Esittelyvaiheessa peli esitetään projektitietokoneelta. Tulevaisuudessa helpommin suoritettavan sovellusversion kehittäminen voi olla mahdollista.

2.4. Jatkokehitys

Esittelyvalmiin pelin arvostelun jälkeen pelin jatkokehitystä voidaan jatkaa. Pelin toiminnalliset rajoitukset ovat tarkoitettu välttämään opinnäytetyön aikarajan ylitys. Jatkokehitykseen kuuluu mahdollisesti uusien ominaisuuksien lisäyksiä, animointia sekä helposti suoritettavan sovellusversion luominen. Jatkokehitys on vain Qt:n harjoittelua varten, sillä peli jää jatkossakin yksityiseen käyttöön.

3. PROJEKTIN JA PELIN SUUNNITTELU

Suunnittelussa käytettiin paperin lisäksi Microsoft Visiota, jonka avulla luotiin käyttöliittymä- ja UML-kaaviot. Käyttöliittymäkaavioilla suunniteltiin pelin pääikkunat ja UML-kaavioilla suunniteltiin pelin käyttäytyminen ja luokat.

Grafiikka jätettiin suunnitelmista kokonaan pois, mutta suurin osa toiminnoista ja niiden sijainnista ikkunoissa sisältyi suunnitelmiin. MS Visiolla luotuja pohjustuksia tuki samalla ohjelmistolla tehdyt UML-kaaviot. Projektille valittiin sopivat UML-kaaviot kuvaamaan käyttäytymistä (käyttötapauskaavio ja tilakaavio), vuorovaikutusta (sekvenssikaavio) ja rakennetta (luokkakaavio).

Käyttöliittymäsuunnitelmat oli helppo luoda. MS Visio sisälsi drag-and-drop-toiminnon, jolla eri käyttöliittymien osat oli vaivatonta sijoittaa oikeille paikoilleen. UML-kaaviot tosin vaativat enemmän tutustumista, sillä niiden rakenne oli vieras. Internetistä löytyvien esimerkkien ja ohjeistuksien avulla kaaviot lopulta saatiin valmiiksi.

3.1. Visio

Idea pelille saatiin kirjoittajan omista kokemuksista kiinteistöongelmien kanssa. Pelin liiketoimintapuoli antoi mahdollisuuden luoda käyttöliittymäystävällinen peli, joten grafiikasta huolehtiminen jätettiin heti alussa pois. Graafinen peli olisi erittäin todennäköisesti vienyt runsaasti enemmän aikaa ja liiallisen grafiikan piirtäminen olisi saattanut kallistua liikaa pois ohjelmointipuolesta.

Alkuperäinen visio oli kehittää peli, joka omaa yksinkertaisen käyttöliittymän sisältäen navigointijärjestelmän kolmen eri päätoiminnon hallintaan. Visio pysyi koko projektin ajan samana, eikä suurempia muutoksia siihen tehty; ainoastaan myöhempiä lisäyksiä. Pankkitoiminto sisältää lainan noston sekä maksamisen. Henkilöstön palkkaaminen ja kiinteistöjen ostaminen on periaatteeltaan sama: henkilöstöä voidaan palkata sekä erottaa ja kiinteistöjä voidaan ostaa sekä myydä. Henkilöstöä käytetään kiinteistöissä ilmeneviin ongelmiin, jotka tulivat myöhempänä lisäyksenä peliin. Nämä peruselementit olivat suhteellisen laajoja työmäärältään, mutta lisätoimintoja kehitettiin mahdollisuuksien mukaan.

Visio oli tarkoitus pitää yksinkertaisena, sillä yksi toiminto voi käsittää pelissä useampia asioita. Esimerkiksi palkan maksaminen työntekijöille voi olla pelaajalle näkymätön prosessi, mutta ohjelmoinnissa se vaati runsaasti työtä. Työtä oli paljon ja vision tuli pysyä järkevissä mittasuhteissa projektin alussa, koska projektin laajeneminen oli todennäköistä.

3.2. Projektin aikataulusuunnittelu

Projekti ei päässyt alkamaan välittömästi vuoden alkupuolella, koska opiskelu vei vielä oman aikansa. Kurseja oli vielä kesken, mutta aina ajan salliessa käytettiin aikaa C++ ja Qt:n opiskeluun. Kurssien määrä väheni lukuvuoden viimeisellä jaksolla, joten aikaa jäi enemmän projektille ja se saatiin käytännössä alkuunsa. Projektin valmistumistavoite asetettiin vuoden 2010 lopulle, koska viimeisiä kurseja oli vielä suunniteltuna syksylle ja niiden päättymisen oli vasta joulukuussa.

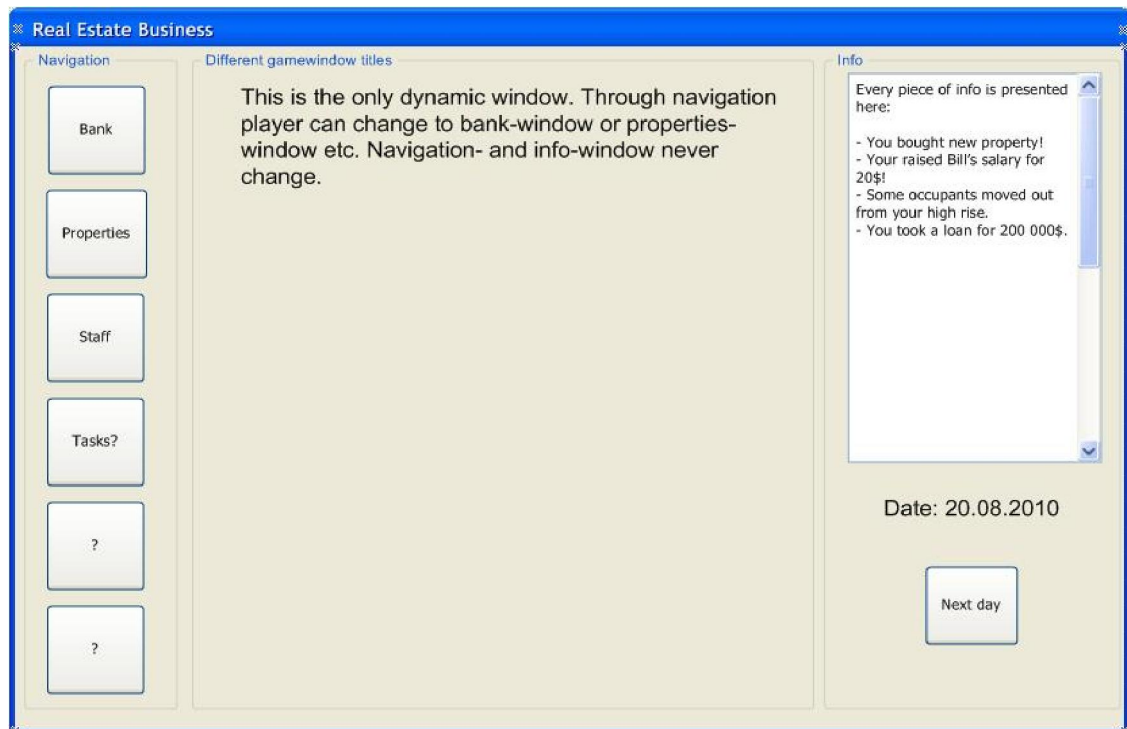
Suurin osa projektin ajasta kului ohjelmoinnissa, sillä työ oli erittäin laaja yhdelle henkilölle. Huomattava aika varattiin silti aiheen opiskelulle, sillä se loi perustan projektille. Uutta asiaa oli erittäin paljon, joten oli järkevää varata niihin tutustumiseen useampi viikko.

Suunnitteluun ja teorian opiskeluun käytettiin aikaa n. 2 kuukautta. Ohjelmointityö voitiin aloittaa jo teorian opiskelun yhteydessä ja sille varattiin aikaa vähintään neljä kuukautta. Ohjelmointiin käytettävät kuukaudet sisälsivät myös kesätyössä käyntiä, joten ohjelmointi ei ollut täyspäiväistä ja pitkittyminen oli mahdollista. Testaus- ja hallintavaiheeseen päästiin, kun peli oli lähes esittelykelpoinen. Aikaa näille varattiin vajaa 2 kuukautta, jonka jälkeen voitiin aloittaa dokumentointi.

Ajallisesti projekti siis vei n. 8 kuukautta, huhtikuusta ja joulukuuhun. Tämän jälkeen peli luovutettiin arvosteltavaksi. Projektin kesto oli pitkä, mutta tulee ottaa huomioon koulu- ja työtehtävät vuoden aikana, sillä nekin veivät oman aikansa. Viimeiset koulukurssit lopuivat vuodenvaihteessa, joten deadline oli järkevää asettaa vuoden lopulle.

3.3. Käyttöliittymän suunnittelu

Peli sisältää yksinkertaisen ja selkeän graafisen käyttöliittymän. Grafiikkaa ei pelissä paljon esiinny, koska tämänkaltaisessa pelissä se ei ole pääosassa. Peli sisältää luonnollisesti tyylilleen paljon numeroiden käsittelyä, joten liiallinen grafiikan käyttäminen olisi turhaa. Peliä hallitaan lähes täysin hiirellä. Ainoastaan lukuarvoja kirjoittaessa näppäimistö tulee tarpeeseen, mutta ei pakolliseksi.



Kuva 2. Käyttöliittymäsuunnitelma, peli-ikkuna

Navigointi pelissä tapahtuu peli-ikkunan (Kuva 2) vasemmassa laidassa olevilla graafisilla painikkeilla. Näillä siirrytään osioista toiseen. Alkuperäiseen suunnitelmaan sisältyivät pankki, kiinteistön osto, henkilöstön palkkaaminen. Näiden lisäksi varattiin tilaa muille vielä tuntemattomille osioille. Jonkinlainen osio oli tarkoitus kehittää myös erilaisille tehtäville (Tasks), joita kiinteistöissä hoidetaan sekä pelaajan tietoja ylläpitävälle tilastoille. Tilastoissa näkyisivät rahatilanne sekä tietoa pelin kulusta.

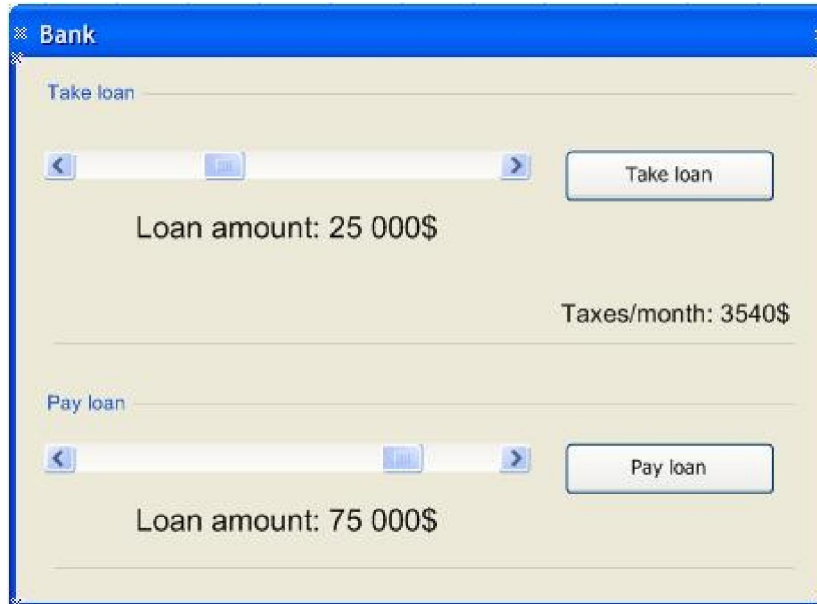
Peli-ikkunan oikeassa laidassa näkyvät pelaajalle tarvittavat tiedot ja tapahtumat. Näiden esittämiseen käytetään suunnitelmassa näkyvää informuutua. Info-osioon oli alun perin tarkoitus yhdistää Suomen kartta, jonka avulla pelaaja pääsisi klikkaamalla haluamaansa paikkakuntaan tutkimaan ostettuja sekä vapaana olevia kiinteistöjä ja niiden ominaisuuksia. Idea kuitenkin hylättiin, sillä se oli enemmänkin ulkoasuun liittyvä asia eikä lopulta niin oleellinen pelille.

Pelin kulun hallitsemiseen päätettiin käyttää pelille asetettua päivämäärää. Pelaaja saa rauhassa tehdä haluamansa muutokset liiketoimiinsa, jonka jälkeen siirrytään klikkaamalla seuraavaan päivään. Tämä mahdollisti myös tapahtumisen helppoluontoisen ohjelmoinnin halutuille päivämäärille tuomaan vaihtelua pelaamiseen.

Käyttöliittymän arkkitehtuurin voi jakaa käytännössä kolmeen osioon, vasemmalta oikealle; navigointi, hallinta ja info. Navigointi- ja info-osio ovat staattisia, eli ne pysyvät aina paikallaan. Hallintaosio on ainoa, joka muuttuu navigoinnin myötä. Hallintaosio siis vaihtuu aina esimerkiksi pankkiasioinnista kiinteistöjen hallintaan jos navigoinnissa niin päätetään tehdä. Käyttöliittymän suunnitelma on liitteenä 1.

3.3.1. Pankkiosion suunnittelu

Pankkiasiointi on yksinkertainen osio pelissä. Pankin toimintoihin kuuluvat lainan ottaminen ja maksaminen. Pelaajan ottamasta lainasta maksetaan kuukausittain 7 % takaisin pankille, joten pelaajan tulisi nopeasti hankkia tuloja kiinteistöjen avulla, jotta pankille saa maksettua lainamäärän takaisin. Pankkiosion suunnitelma on liitteenä 2.

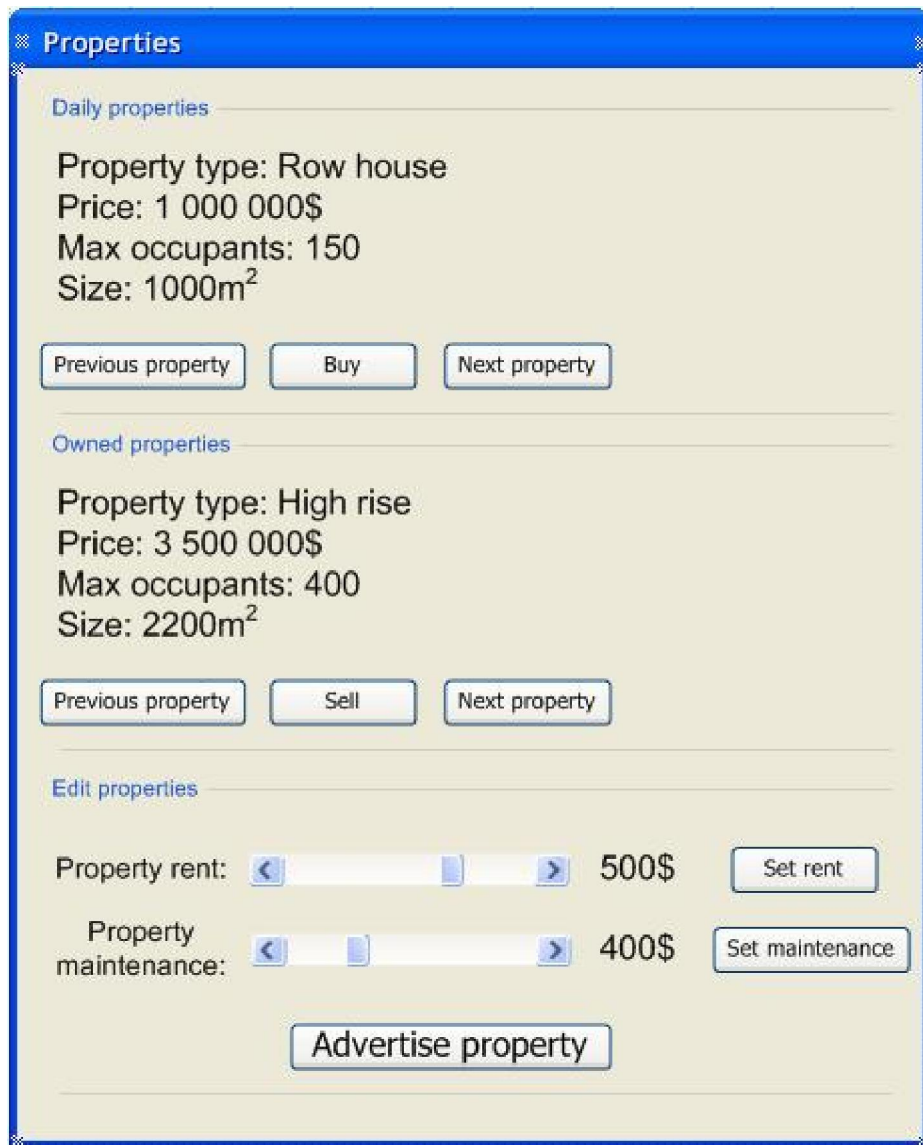


Kuva 3. Pankin yleiskuva

Kuvassa 3 on pankkiosion ainoat toiminnot. Lainan ottaminen ja maksaminen tehdään neljää eri widgettiä käyttäen. Liukupalkkien avulla säädetään haluttu lainan määrä käsiteltäväksi. Lainan otto tai maksu vahvistetaan normaalilla painikkeella. Lainasta lasketaan vero, joka pelaajalle päivittyy aina toimintoa kutsuessa. Lainalle asetettiin enimmäismäärä, mutta sen lopullinen määrittäminen jäi viimeiselle testivaiheelle vaikeusasteen sopivuuden löytämiseksi.

3.3.2. Kiinteistöosion suunnittelu

Pelin tärkein ja todennäköisesti monipuolisin toiminto sisälsi kiinteistöjen hallinnan. Suunnitelmana oli näyttää pelaajalle samassa ikkunassa sekä vapaana olevat ja ostetut kiinteistöt. Näiden lisäksi ostetuille kiinteistöille annettiin mahdollisuus säätää vuokran sekä huoltorahan määrää. Nämä kaksi arvoa vaikuttavat kiinteistön tilaan. Liian korkea vuokra karkottaa asukkaita, kun taas alhaisella vuokralla asukkaita muuttaa normaalia enemmän kerralla kiinteistöön. Huoltoraha vaikuttaa lähes samalla tavalla kiinteistössä tapahtuviin eri ongelmiin. Suuri asetettu huoltoraha tarkoittaa vähäistä ongelmamäärää ja päinvastoin pienellä rahoituksella. Kiinteistöosion suunnitelma on liitteenä 3.



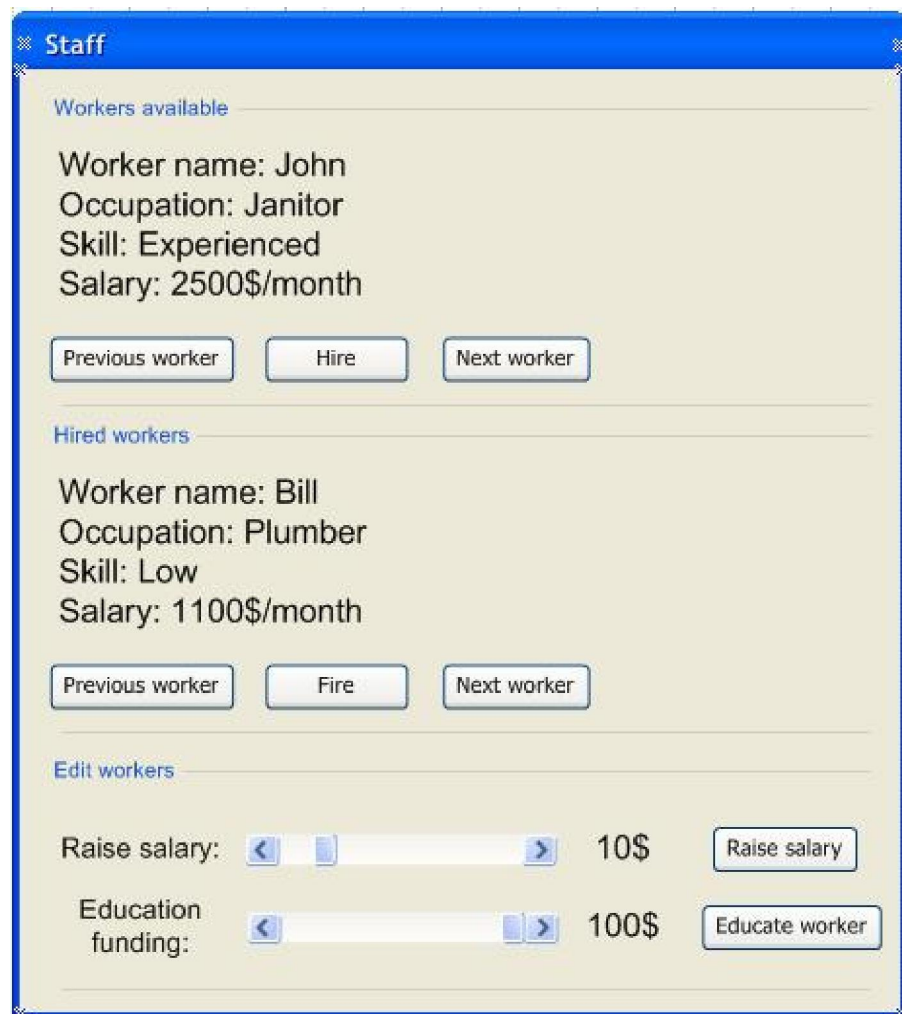
Kuva 4. Kiinteistöosion yleisnäkymä

Kiinteistöosion ylin osa on tarkoitettu kiinteistön ostoa varten ja sen alapuolella on ostettujen kiinteistöjen tiedot (Kuva 4). Alimmat toiminnot ovat tarkoitettu tarkasteltavan kiinteistön hallitsemiseen. Kiinteistöosiossa pelaajalle annetaan tarkasteltavaksi uusia kiinteistöjä. Alun perin idea oli ohjelmoida jokainen pelissä esiintyvä kiinteistö käsin. Ideaa ei kuitenkaan lopulta hyväksytty, koska se vaati erittäin paljon työtä. Vaihtoehtoisesti keksittiin kiinteistöjen luomiseen tarkoitettu satunnaisgeneraattori.

Pelaaja voi ostaa vapaana olevan kiinteistön itselleen jos rahaa on tarpeeksi. Kiinteistöoston jälkeen voidaan asettaa vuokra ja huoltoraha, jonka jälkeen kiinteistöä on mahdollisuus mainostaa. Kiinteistö on aluksi tyhjä asukkaista, joten sen mainostaminen tuo asukkaita. Asukkaita ei kuitenkaan muuta heti mainostuksen jälkeen, vaan se vie muutaman pelipäivän.

3.3.3. Henkilöstöosion suunnittelu

Kiinteistöistä huolehtimiseen pelaajan tulee palkata henkilöstöä. Ammatteja peliin luotiin kolme: talonmies, rakentaja ja putkimies. Ammatteja voidaan keksiä jatkokehityksessä lisää. Työntekijöiden merkitys johti osaltansa myös tehtäväosion syntymiseen. Ensimmäisissä suunnitelmissa ei ollut ideaa mihin henkilöstöä käytettäisiin, joten tehtävien ratkaiseminen antoi tarkoituksensa niille. Henkilöstöosion suunnitelma on liitteenä 4.

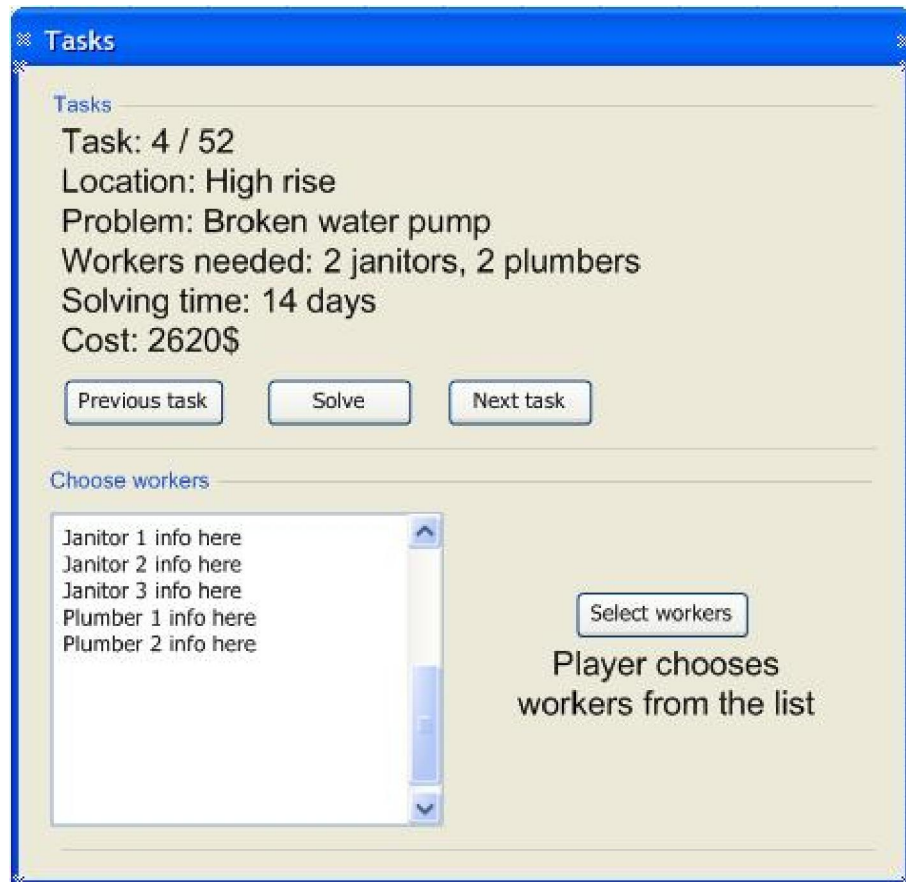


Kuva 5. Henkilöstöosion yleisnäkymä

Henkilöstöosio (Kuva 5) on melko identtinen kiinteistöosion kanssa. Päivittäin esitetään pelaajalle työtä vailla olevia henkilöitä, joita tarpeen mukaan voidaan palkata. Työntekijän palkka lisätään pelaajan kuukausittaisiin menoihin ja tarpeen tulleen työntekijälle voidaan antaa potkut. Työntekijän taitotasoon pelaaja voi vaikuttaa koulutuksella tai palkan korotuksella. Tarkkaa määritystä niiden vaikutuksesta ei suunnitteluvaiheessa vielä ollut, mutta jonkinlainen henkilöstöhallinta nähtiin hyväksi ominaisuudeksi peliin. Henkilöstöhallintaa jätettiin kuitenkin vielä esittelypelistä pois.

3.3.4. Tehtäväosion suunnittelu

Kun vuokran ja huoltorahan asetus lisättiin kiinteistöosion suunnitelmaan mukaan, tuli tarpeen alustaa suunnitelma kiinteistöjen tehtäväosiosta. Huoltorahan suuruudesta riippuen tehtäviä tulee enemmän tai vähemmän jokaisesta omistetusta kiinteistöstä ja niiden ratkaisemiseen käytetään palkattua henkilökuntaa. Tehtäväosion suunnitelma on liitteenä 5.



Kuva 6. Tehtäväosion yleisnäkymä

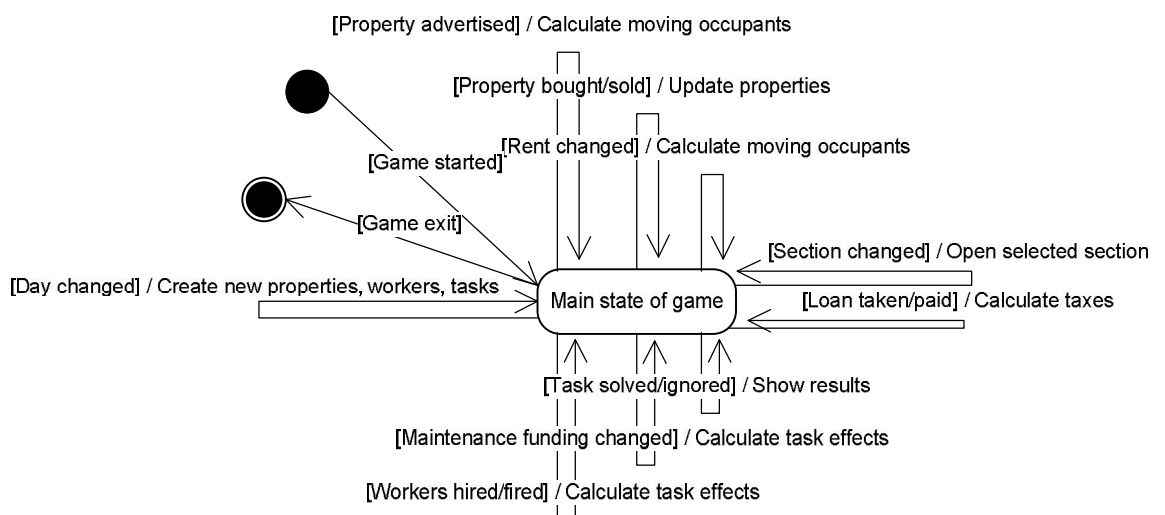
Tehtäväosion suunnitelma (Kuva 6) sisältää kaksi osaa: ylemmällä tasolla on kiinteistötehtävän tiedot ja niiden alapuolella on sopivien työntekijöiden valitseminen tehtävälle. Tehtävän tunnistamiseen esitetään tarvittavat tiedot: ongelman nimi, missä ongelma esiintyy, paljonko työntekijöitä tarvitaan sen ratkaisemiseen, kuinka kauan tehtävän suorittaminen vie ja paljonko se tulee maksamaan.

Pelaajan tulee valita sopivat työntekijät alemmassa osiossa olevasta laatikosta. Tehtävälle valittujen työntekijöiden taitotasot vaikuttavat lopputulokseen. Puutteellisella työtaidolla tehty työ voisi tuottaa lisäkuluja tai muita menetyksiä pelaajalle. Työntekijöiden valintalaatikko hylättiin ideana jatkuvan valintatoiston vuoksi.

3.4. Pelin toiminnallisuuden ja rakenteen suunnittelu

Pelin toimintojen ja sisäisen rakenteen selventämiseksi käytetään UML-kaavioita. Pelistä tehtiin käyttötapauskaavion lisäksi kolme muuta kaaviota: Tilakaavio, sekvenssikaavio ja luokkakaavio. Kaaviot sisältävät kaikki pelin esittelyversion toiminnot, jotka eivät välttämättä näkyneet alkuperäisissä suunnitelmissa. Kaavioiden tekoon käytettiin Microsoft Visio -ohjelmistoa.

3.4.1. Tilakaavio



Kuva 7. Tilakaavio

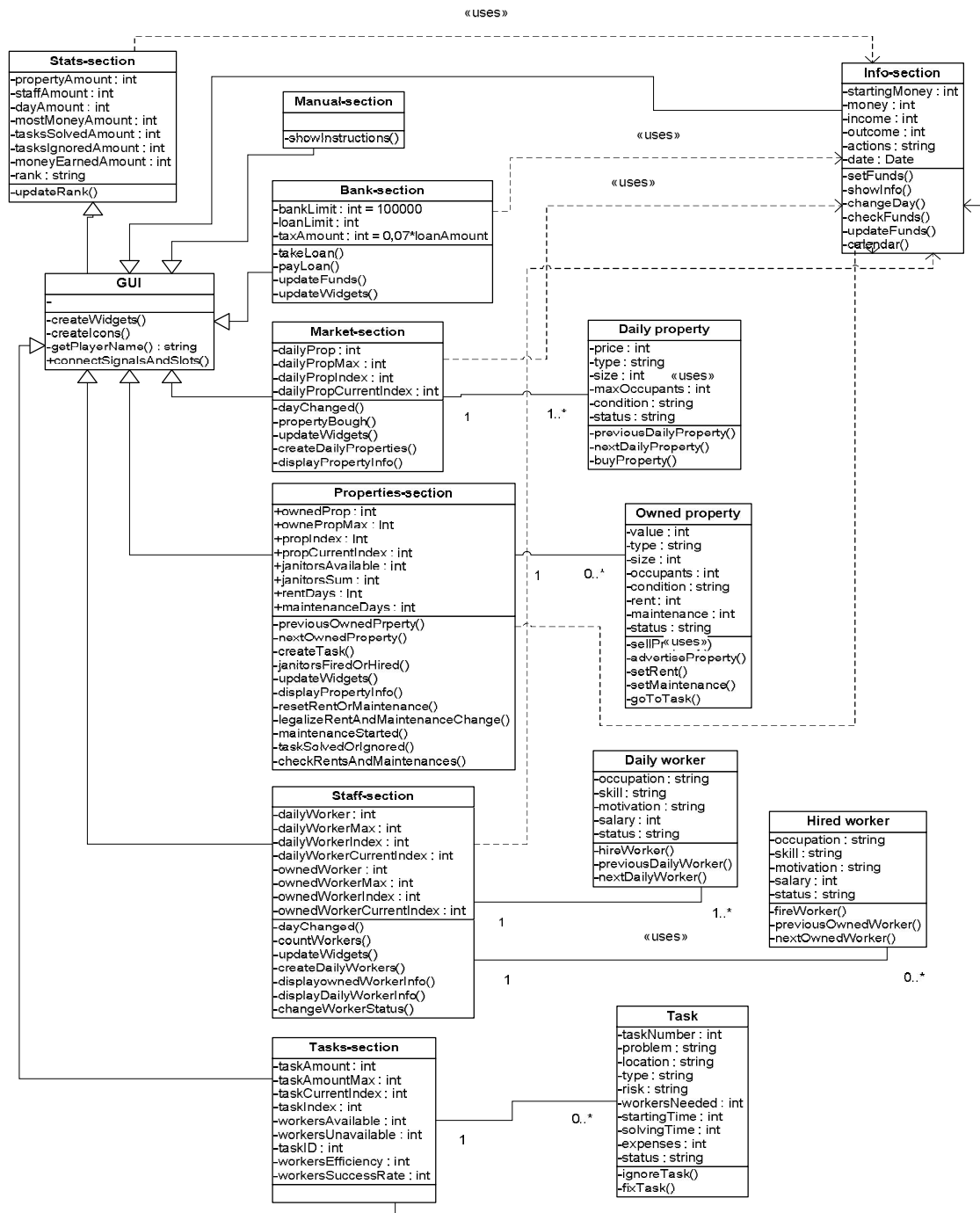
Pelin eri tilasiirtymistä voidaan kuvata tilakaaviolla (Kuva 7). Peli on ohjelmoitu käytännössä yhden päätilan ympärille. Päätila on peli-ikkuna kokonaisuudessaan, jossa pelaaja voi suorittaa toimenpiteitä. Pelille määrätyt toimenpiteet ovat hetkittäisiä tapahtumia, jotka eivät yleensä näy pelaajalle uutena tilana lainkaan. Eli pelaaja valitsee toimenpiteen ja peli suorittaa annetun tehtävän mutta tila ei vaihdu. Tilakaavio on liitteenä 6.

3.4.2. Sekvenssikaavio

Pelin elementtien vuorovaikutusta toisiinsa kuvaa sekvenssikaavio. Kaaviota luetaan vasemmalta oikealle ja ylhäältä alas. Kaavio näyttää, miten peli etenee normin mukaisessa aikajärjestyksessä. Normin mukaisella pelillä tarkoitetaan pelin pelaamista sille suunnitellun järjestyksen mukaisesti. Järjestys voidaan selittää yksinkertaisesti numerojärjestyksessä:

1. Pelaaja ottaa lainan.
2. Maksettava verosumma päivittyy ja tieto siitä menee info-osioon
3. Pelin järjestelmä on luonut kiinteistöt ja pelaaja ostaa kiinteistön lainarahoilla.
4. Ostetun kiinteistön tiedot lähetetään kiinteistöosioon ja rahatilanne päivittyy info-osiossa.
5. Pelin järjestelmä on luonut työnhakijat ja pelaaja palkkaa työntekijän.
6. Järjestelmä laskee pelaajan palkkaamat työntekijät ja ilmoittaa ne tehtäväosiossa.
7. Pelaaja asettaa ostetulle kiinteistölle vuokran ja huoltorahan sekä aloittaa kiinteistön mainostuksen.
8. Muutokset lähetetään info-osioon näkyviin ja järjestelmä luo kiinteistölle mahdolliset tehtävät.
9. Pelaaja ratkaisee tai välttää kiinteistön ongelman, joka päivittää ongelman nykyisen tilanteen päätöksen mukaan.
10. Pelaajan päätöksen seuraukset näkyvät info-osiossa.

3.4.3. Luokkakaavio



Kuva 9. Luokkakaavio

Pelin rakennetta kuvaamaan valittiin luokkakaavio (Kuva 9). Luokkakaavio kertoo jokaisen luokan eri attribuutit ja funktiot. Pelissä jokainen osio on oma luokkansa ja useimmat toiminnot ovat oman luokkansa sisäisiä. Luokat luodaan pelin alussa, kun käyttöliittymä on avattu. Toiminnot luokissa ovat riippuvaisia pelaajasta. Luokkakaavio on liitteenä 7.

4. PELIN TOTEUTUS

Projektin kehitystä suoritettiin henkilökohtaisella tietokoneella, jolle asennettiin Qt-creator. Qt-creator sai ajoittain ohjelmistopäivityksiä, mutta varmuuden vuoksi ne jätettiin asentamatta yhteensopivuusongelmien välttämiseksi. Projektikansiosta ja projektidokumentista otettiin varmuuskopiot ulkoiselle kovalevylle aina edistymisen myötä.

4.1. Qt

Qt on alustariippumaton ohjelmistojen ja graafisten käyttöliittymien kehitysympäristö. Qt sisältää C++ -luokkakirjaston ja alustariippumattoman ohjelmointiympäristön. Qt:n alkuperäinen kehittäjä on norjalainen Trolltech, mutta Nokia osti sen vuonna 2008.

Koska Qt oli täysin uusi kehitysympäristö kirjoittajalle, sen tutustumiseen varattiin aikaa. Internetistä löytyi erinomaisesti esimerkkejä ja vinkkejä, joita pystyi hyödyntämään tämänkaltaisessa projektissa. Ennen varsinaisen projektin aloitusta testattiin useita erilaisia esimerkkejä, jotta saatiin jonkinlainen käsitys Qt-ohjelmoinnista. Opiskelu vei aikansa, mutta se oli välttämätöntä. Ohjelmoinnin yhteydessä apua etsittiin tarvittaessa Qt-ongelmiin keskittyvältä keskustelupalstalta. Keskustelupalstalla kirjoitetaan englanniksi ja apua saa kysymällä harjoittelijoilta ja alan ammattilaisilta. /2/

Qt-creatorilla on kaksi mahdollisuutta kehittää graafista käyttöliittymää hyödyntävä sovellus. Helpompi ja yksinkertaisempi tapa on käyttää ohjelmistoon sisältyvää widget-valikkoa, josta voi hiirellä siirtää erilaisia widgettejä suoraan sovellusikkunaan. Yhtään koodiriviä ei siis tarvitse kirjoittaa, vaan kaikki luominen tapahtuu drag-and-drop-menetelmällä. Widgettien yhdistäminen käy lähes yhtä helposti: yhdistämällä kaksi widgettiä hiirellä toisiinsa voi määritellä miten niiden käyttö vaikuttaa toisiinsa. Ohjelmoiminen jäisi siis melko vähälle.

Enemmän aikaa ja vaivaa vievä tapa on tehdä kaikki edellä mainittu ohjelmointityönä. Se tarkoittaa sitä, että jokainen widget luodaan koodirivillä ja sen sijainti sovellusikkunassa määritellään tietyillä lukuarvoilla. Tämä tietysti lisää satoja koodirivejä verrattuna aikaisemmin mainittuun helpompaan ohjelmoimistapaan. Kuitenkin opinnäytetyön aiheena oli peliohjelmointi, joten ohjelmointityylin valinta oli selkeä: tehdään kaikki tekstipohjaisesti. Tällä tavalla myös oppi paljon enemmän Qt:n käyttöä.

4.2. C++

Koulussa panostettiin pääosin C-kielen opettamiseen, joka antoi vain jonkinlaista pohjaa C++-kielen ymmärtämiseen. Tämän vuoksi ohjelmointikieli oli lähes täysin vieras. Kielen opiskeluun käytettiin Qt-kehitysympäristön ja C++ -ohjelmoinnin oppikirjaa, jossa selitetään asiat Qt:n ja kielen perusteista vaativampiin ohjelmointimenetelmiin. Suurin osa opitusta taidoista tuli kuitenkin jättämällä teoria käytännön työn ohelle. Lukumateriaalin oli vain tarkoitus tukea ohjelmointia. Koko kirjaa ei siis luettu läpi. /1/

4.3. Ohjelmointi

Tämän kaltaisessa projektissa suurin työ on itse ohjelmointi. Varsinainen työ alkoi erään Qt-esimerkin pohjalta. Tämän esimerkin käyttöliittymä oli suhteellisen nopea muokata sellaiseksi kuin pelille oli suunniteltu. Tämän jälkeen käyttöliittymään lisättiin uusia painikkeita, tekstiä, lukuarvoja, joista ulkoasun kokonaisuus alkoi syntyä. Millään napilla tai muullakaan widgetillä ei ollut alussa minkäänlaista yhteyttä toisiinsa, vaan käyttöliittymän yleisnäköyksen luonti nähtiin hyväksi aloituskohdaksi.

Käyttöliittymä sai yleisilmeensä alkuperäisen käyttöliittymäsuunnitelman mukaan, eli kaikki grafiikka jätettiin tässä vaiheessa vielä pois. Ulkoasua parantavaa grafiikkaa lisättiin kyllä, mutta se jäi myöhempään vaiheeseen, kunnes suurin osa toiminnoista oli ohjelmoitu.

4.3.1. Pankki

Ohjelmointi aloitettiin pankkiosioista. Se nähtiin sopivana lähtöaskeleena, koska se sisälsi vain lainan noston ja sen maksun. Koko osio sisältää vain 2 liukupalkkia, 2 numerolaatikkoa ja 2 nappia. Liukupalkeilla ja numerolaatikoilla hallitaan haluttua lainan määrää ja vahvistus suoritetaan napeilla. Numerolaatikot eivät olleet suunnitelmassa, mutta ne antavat tarkemman mahdollisuuden säätää haluttua lainan määrää.

Suunnitelmien mukaan pelaajan on pakko turvautua pankkilainaan heti pelin alussa, sillä ideana on aloittaa liiketoimet tyhjästä. Pankkilaina on rajoitettu, ja siitä maksetaan kuukausittain 7% takaisin pankille. Laina vähentyy kuukausittain automaattisesti. Jatkokehityksessä voidaan mahdollisesti ottaa huomioon koron merkitys.

The image shows a software interface for managing loans, divided into two main sections: 'Take a loan' and 'Pay loan'.

Take a loan section:

- Label: 'Take a loan'
- Control: A slider for 'Loan amount:' with a value of 14543 and a numeric input field.
- Button: A blue button labeled 'Take loan'.
- Text: 'Bank will you give you max. 100 000\$. You can still withdraw: \$25647'
- Text: 'Bank interest rate is 7%. You have to pay it monthly: \$5204'

Pay loan section:

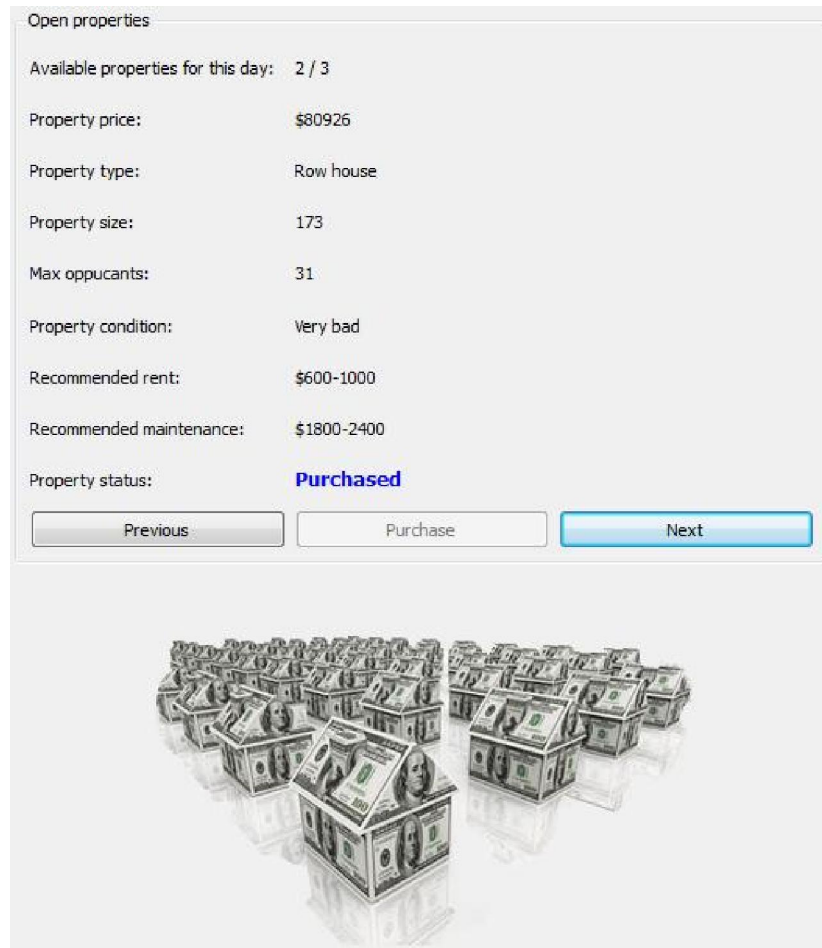
- Label: 'Pay loan'
- Control: A slider for 'Loan amount:' with a value of 29391 and a numeric input field.
- Button: A grey button labeled 'Pay loan'.
- Text: 'Taken loan: \$74353'

Kuva 10. Pankkitoiminnot

Lopullinen pankki-osio (Kuva 10) valmistui lähes täysin alkuperäisen suunnitelman mukaan. Lisänä ovat vain numerolaatikot ja aina näkyvillä olevat lainan määrät. Jotta pankkiosioon saatiin hieman väriä, toimintojen alle lisättiin kuva rahasäkistä. Erilaisilla kuvilla piristettiin myös muiden osioiden yleiskuvaa.

Yksinkertaisuudestaan huolimatta pankkiosio ei valmistunut vauhdilla, koska siinä ilmeni ensimmäinen haastava ongelma. Tarkoitus oli lähettää pelaajan valitsema lainasumma info-osioon näkyville, mutta lukuisten yrityksiä ja epäonnistumisten jälkeen sitä ei saatu toteutettua. Ongelmasta kerrotaan enemmän myöhemmin tekstissä. Pankkiosio ohjelmoitiin muilta osin valmiiksi, eli lainan maksaminen ja ottaminen päivitti widgetit ja luvut, mutta alussa sillä ei ollut lainkaan yhteyttä toisiin osioihin.

4.3.2. Kauppa



Kuva 11. Kiinteistökauppa

Kauppaosiolle (Kuva 11) ei ollut lainkaan suunnitelmaa. Se sai alkunsa kiinteistöosiosta, jossa suunnitelmien mukaan olisi tapahtunut sekä kiinteistön osto että hallinta. Kuitenkin pian huomattiin, että tämä ratkaisu toi liikaa informaatiota liian pieneen tilaan. Nähtiin järkeväksi luoda täysin uusi osio ainoastaan kiinteistön ostoa varten. Kauppaosion rakenne on kuitenkin samanlainen kuin suunnitelmissa. Kiinteistöt vain saivat uusia ominaisuuksia.

Kauppaosio tuo pelaajalle uusia kiinteistöjä ostettavaksi. Alun perin idea oli tehdä jokainen pelissä esiintyvä kiinteistö käsin, mutta se tuntui vaativan liikaa työtä. Tämän myötä kehitettiin satunnaisgeneraattori, joka arpoo aina tarvittaessa muutaman uuden kiinteistökohteen täysin satunnaisilla arvoilla. Näin kiinteistökohteet ovat kaikki erilaisia ja niitä on loputtomasti.

Kiinteistöjen luominen aloitetaan arpomalla sen tyyppi ja kunto. Näiden kahden kriteerin perusteella arvotaan kiinteistön arvo ja koko. Kiinteistöjä on rajattu määrä kerralla ja uudet kiinteistöt arvotaan, kun pelaaja siirtyy pelissä seuraavaan päivään.

Suunnitelmasta poiketen kiinteistöt saivat enemmän ominaisuuksia. Nyt jokaisella kiinteistöllä ovat seuraavat ominaisuudet: tyyppi, koko, asukasmäärä, kunto ja hinta. Tyyppejä on yhteensä neljä; kaksio, mökki ja rivi- sekä kerrostalo. Kiinteistöjen myyntihinnat sekä vuokrat pääteltiin aitojen kiinteistökohteiden perusteella. Hintojen perusteella kiinteistöille luotiin kuntoluokitukset, joita on yhteensä seitsemän, surkeasta loistavaan. Suositeltava vuokra ja huoltoraha esitetään muiden ominaisuuksien yhteydessä. Suositellut huoltorahat ovat päätelty vaikeusasteen mukaan.

Satunnaisgeneraattorin myötä kiinteistöjen luominen oli pääosin helppoa. Ongelmilta ei kuitenkaan vältytty. Ensin oli vain ”Next”-nappi, joka oli tarkoitettu kohteiden selaamiseen. Nappia käytettiin ensiksi testaamaan, miten satunnaisien kiinteistöjen luominen toimii. Ongelmana oli, että jos edellinen tarkasteltu kiinteistö olikin nykyistä kiinnostavampi, sitä ei enää päässyt tarkastelemaan eikä ostamaan. Jonkinlainen muistitoiminto oli siis keksittävä, jotta aiemmin tarkastellut kiinteistöt pysyivät tallessa. Osio jätettiin pankin tavoin kesken myöhempää suunnittelua varten ja keskityttiin muiden osioiden valmisteluun. Kului useampi viikko ennen kuin ongelmalle löytyi järkevä ratkaisu.

Ratkaisu ongelmaan oli QList-toiminto. Tämä toiminto mahdollisti tiedon tallentamisen listalle, josta oli helppoa hakea mikä tahansa sille tallennettu tieto. Tieto tallennettiin yksinkertaisella ohjelmointisilmukalla. Peli arpoo ensin kiinteistöjen määrän, jonka jälkeen jokaiselle kiinteistölle arvotaan tyyppi ja ominaisuudet. Tämän jälkeen toistosilmukka suorittaa QListalle tallentamista, kunnes kaikki luodut kiinteistöt ovat muistissa. Tätä prosessia suoritetaan aina pelipäivän vaihtuessa. Tallennettuja tietoja pystyi tämän jälkeen helposti selaamaan niille tarkoitetuilla napeilla.

QList osoittautui tarpeelliseksi muissakin tapauksissa, kuten myös kiinteistökaupoissa. Ostamisen yhteydessä Kiinteistö-osioon lähetettiin kyseessä olevan kiinteistön jokainen ominaisuus, jotka tallennettiin QListin muistiin. Helpompi tapa olisi ollut yhdistää Kauppa- ja Kiinteistö-osio, koska eri luokkien välistä yhteyttä ei olisi tarvinnut luoda. Selkeämmän käyttöliittymän vuoksi nähtiin kuitenkin järkevämmäksi kiinteistöjen ostaminen ja hallitseminen erikseen.

4.3.3. Kiinteistöt

Jokaisen ostetun kiinteistön tiedot lähetetään tähän osioon, jossa pelaaja pystyy selaamaan niitä samalla tekniikalla kuin Kauppa-osiossa. Ostamisen sijasta kiinteistö voidaan myydä, jos se ei tuota tarpeeksi. Myynnin tapahtuessa kiinteistö poistetaan QList:n muistista ja luonnollisesti pelaaja saa myyntirahat. Kuva 12 näyttää, kuinka paljon kiinteistöjen hallinta kasvoi ominaisuuksiltaan alkuperäisestä suunnitelmasta. Kiinteistöosion laajuuden vuoksi kiinteistöjen osto ja hallinta päätettiin toteuttaa erikseen.

The screenshot displays a property management interface with three main sections:

- Owned properties:** Shows details for a property, including its value, type, size, occupants, condition, rent, and maintenance costs. The status is 'Under maintenance' with 1 day remaining. Navigation buttons for 'Previous', 'Sell', and 'Next' are provided.
- Edit property:** Offers an 'Advertise' button and an 'Auto advertise' option. It includes sliders and input fields to adjust 'Set rent' (currently 1897) and 'Set maintenance' (currently 1914). A 'Janitors available' count of 2 is also shown.
- Current tasks:** A table listing tasks with columns for Type, Task ID, and Status. Each task has a 'Go to task' button.

Tasks!	Type	Task ID	Status	
Task #1:	Flood	187	Unsolved	Go to task
Task #2:	Broken pipe(s)	183	Under work	Go to task
Task #3:	No task	0	-	Go to task

Kuva 12. Kiinteistöjen hallinta

Jokaiselle kiinteistölle ohjelmoitiin mahdollisuus asettaa vuokra ja huoltoraha. Vuokran säätörajat on laskettu aitojen kiinteistökohteiden perusteella. Huoltoraha on päätelty vaikeusasteen sekä kiinteistön kunnan ja koon mukaan. Vuokran ja huoltorahan suuruudet saattavat jatkokehityksessä muuttua oikean vaikeusasteen etsimisessä. Vuokran ja huoltorahan voi nollata tai muuttaa ainoastaan jokaisen pelikuukauden ensimmäisenä viitenä päivänä. Tämä rajoitus tehtiin huijauksen ehkäisemiseksi, jossa pelaaja korottaa juuri ennen vuokran- ja palkanperintäpäivää arvot huippuunsa omien etujensa mukaisesti.

Kiinteistö on ostamisen jälkeen tyhjä eikä se tuota pelaajalle rahaa. Kun vuokra on asetettu, kiinteistöä voidaan mainostaa. Mainostaminen maksaa ja menee muutama pelipäivä, kunnes asukkaita alkaa muuttamaan kiinteistöön. Asetettu vuokran määrä vaikuttaa asukkaiden halukkuuteen muuttaa kiinteistöön. Riippumatta vuokran määrästä, kiinteistö ei kuitenkaan täyty yhdellä mainostuskerralla, vaan asukkaita tulee satunnainen määrä. Kuitenkin tämä määrä on suurempi, kun vuokra on alhaisempi.

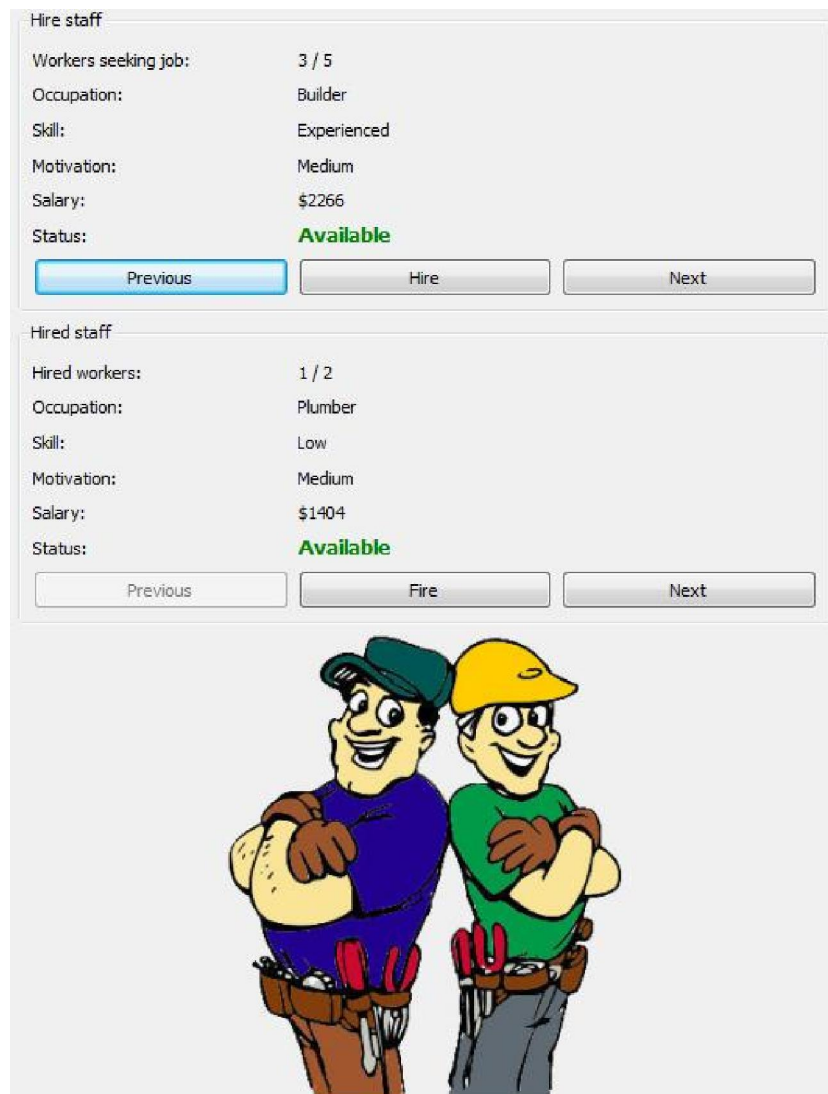
Huoltoraha määrittelee kiinteistölle aiheutuvien sattumuksien määrän. Huoltoraha voidaan asettaa, kun talonmies on palkattu. Silloin vapaana oleva talonmies alkaa työskennellä kyseisessä kiinteistössä. Mitä suurempi huoltoraha, sitä pienemmällä todennäköisyydellä kiinteistössä sattuu ongelmia. Sekä vuokran ja huoltorahan merkitys vaatii kuitenkin lopullista hiomista. Tämä tehdään jatkokehityksessä, joten ei ole varmaa, onko peli esittelyvaiheessa helppo, sopiva vai vaikea.

Testausvaiheessa kävi ilmi, että pelaajan kiinteistöjen kasvaessa ongelmien määrä voi olla suuri. Tietyn kiinteistön tietty ongelma oli vaikeaa etsiä pitkästä ongelmalistasta, joten kiinteistöosioon kehitettiin pieni navigointitoiminto. Kiinteistöllä voi olla kolme ongelmaa kerralla ja ne ilmestyvät tarkkailun alla olevan kiinteistön pieneen ilmoitusosioon. Tästä voidaan nappia painamalla siirtyä suoraan halutun tehtävän kohdalle tehtäväosioon. Toiminnon on tarkoitus nopeuttaa pelin kulkua.

Toisena lisäominaisuutena kehitettiin automaattinen mainostus. Testivaiheessa oli turhauttavaa klikkailla useita kiinteistöjä eteenpäin ja mainostaa jokaista yksi kerrallaan, joten ongelmaan keksittiin toiminto nimeltä ”auto advertise”. Asettamalla toiminto päälle, jokaista kiinteistöä mainostetaan automaattisesti sen ollessa mahdollista.

4.3.4. Henkilöstö

Henkilöstö-osio (Kuva 13) on toiminnoiltaan lähes identtinen Kauppa-osion kanssa. Työnhakijoita on siis pelipäivälle rajattu määrä ja pelaaja voi selata niitä samalla tavalla kuin kiinteistöjäkin. Pelipäivän vaihtuessa sekä palkkaamisessa tiedot työntekijöistä siirtyvät QList:n muistiin. Uusia ja palkattuja työntekijöitä pystyy tarkkailemaan samassa ikkunassa, toisin kuin kiinteistöjen kanssa.



Kuva 13. Työntekijöiden hallinta

Jokaisella työntekijällä on seuraavat ominaisuudet: ammatti, taitotaso, motivaatio ja palkka. Nimiominaisuus kuului suunnitelmaan, mutta se jätettiin lopulta pois, koska sillä ei ollut lopullisessa pelissä lainkaan merkitystä. Ammatteja on talonmies, putkimies ja rakentaja. Talonmies sallii kiinteistön huoltorahan asettamisen, mutta muut työntekijät tulevat vasta tarpeen erilaisissa tehtävissä, joita kiinteistöissä ilmenee.

Motivaatio oli uusi lisäys työntekijän ominaisuuksiin, koska työntekijöille haluttiin enemmän vaikutusta pelin kuluun. Motivaation merkitys näkyy työtehtävän suoritusajassa. Huono motivaatio voi johtaa työtehtävän pitenemiseen ja päinvastoin. Alun perin työntekijän suoritukseen piti vaikuttaa suunnitelmaan kuuluva palkankorotus ja koulutus, mutta niiden ohjelmoiminen jätettiin toistaiseksi kokonaan pois. Nämä toiminnot voidaan ohjelmoida vielä tulevaisuudessa, mutta esittelyversiossa niitä ei ole.

Työntekijälle arvottu taitotaso määrittelee palkan suuruuden. Taitotaso vaikuttaa myös tehtävän onnistumiseen rahallisesti. Perusaate on, että kokenut työntekijä pystyy suorittamaan tehtävän ilman lisäkuluja ja -ongelmia.

Motivaation ja taitotason merkitys työtehtävään kääntyi yksinkertaiseksi ratkaisuksi. Koska työtehtäviä tulee pelin aikana paljon, jokaisen työntekijän erikseen valitseminen oli tylsää ja aikaa vievää. Ratkaisuna ongelmaan peli laskee jokaisen pelaajan palkkalistoilla olevan työntekijän motivaatio- ja taitotason ja esittää niiden keskiarvon pelaajalle onnistumisprosenttina. Nämä onnistumisprosentit lopulta ratkaisevat, mikä on tehtävän lopputulos.

4.3.5. Tehtävät

Jotta kiinteistöjen hallitseminen ei jäänyt liian yksipuoliseksi, luotiin kiinteistöaiheiset tehtävät (Kuva 14). Tehtävät määräytyvät kiinteistölle asetetun huoltorahan mukaan. Jos talonmiestä ei ole tai kiinteistöllä ei muuten vain ole asetettu lainkaan huoltoa, ongelmia kiinteistössä alkaa kasaantua enemmän kuin normaalisti. Mitä enemmän huoltoon panostaa, sitä vähemmän ongelmia ilmenee. Kaikki määräytyy todennäköisyyslaskurin mukaan.

Tasks	
Task number:	4 / 5 ID #4
Problem:	Broken door lock
Location:	Row house
Type:	Maintenance
Risk:	Low
Workers needed:	1 janitors 0 plumbers 0 builders
Need to start in:	0 days
Solving time:	1 days
Expenses:	\$57
Problem status:	Under maintenance
<input type="button" value="Previous"/> <input type="button" value="Ignore"/> <input type="button" value="Fix"/> <input type="button" value="Next"/>	
Workers available	
Overall efficiency:	80%
Overall success rate:	81%
Available	
Janitors:	1
Plumbers:	6
Builders:	2
Working	
Janitors:	1
Plumbers:	0
Builders:	2

Kuva 14. Tehtävähallinta

Ongelmat ohjelmoitiin kaikki yksitellen ja niitä aluksi ohjelmoitiin yli kaksikymmentä. Jokainen ongelma kertoo pelaajalle, mikä on ongelman sijainti, ongelman tyyppi ja mahdolliset kulut. Näiden lisäksi kerrotaan, mitä henkilöstöä tarvitaan ongelman ratkomiseen sekä milloin tehtävä pitää viimeistään aloittaa ja kuinka kauan tehtävän suorittaminen kestää. Jokainen tehtävä saa myös tunnisteiden (ID), jota käytetään kiinteistöosiossa tehtävän etsimiseen.

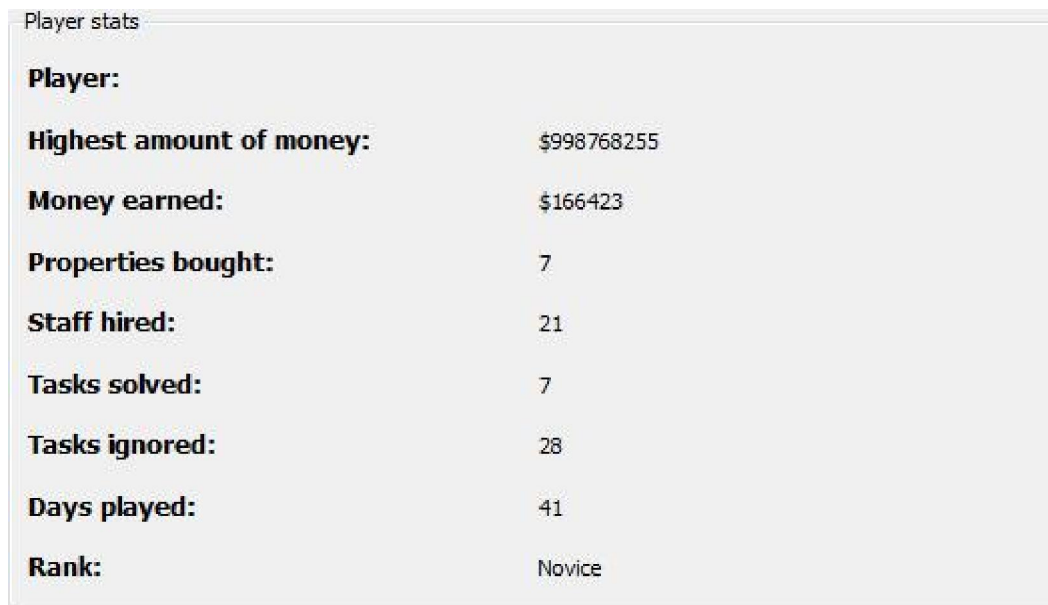
Jokaisella tehtävällä on myös oma riskiluokitus, joka jakaantuu kolmeen ryhmään: alhainen, keskiverto ja korkea. Taso vaikuttaa tehtävään tarvittavaan rahoitukseen ja työmiehien määrään. Alhaiset ongelmat kiinteistössä ovat yleisimpiä, sillä niitä tulee 60 %:n todennäköisyydellä. Keskivertoja tulee 30 %:n todennäköisyydellä ja korkeamman tason ongelmia vain 10 %:n todennäköisyydellä. Esimerkki alhaisesta tehtävän tasosta voisi olla rikkinäinen ikkuna, keskiverrosta ryöstö ja korkeasta tasosta tulipalovahinko.

Ongelman voi joko ratkaista resurssien sen salliessa tai jättää hoitamatta. Jos tehtävän aloitusaika ylittyy, tehtävä jätetään automaattisesti hoitamatta ja siitä yleensä seuraa lisäkuluja pelaajalle. Pahimmassa tapauksessa asukkaita alkaa muuttaa pois kiinteistöstä, jossa pelaaja ei hoida tehtäviään.

Tehtäväosiossa näkyy myös pelaajan työntekijöiden yhteenveto. Jokaisen vapaana ja työssä olevan työmiehen lukumäärä on tarkasteltavissa. Työtehtävään vaikuttavat motivaatio- ja taitotasot ovat laskettuna näiden yläpuolella. Niiden avulla tarkastetaan työtehtävän lopputulos. Esimerkiksi 80 %:n tehokkuudella on 20 %:n mahdollisuus työn pitenemiselle. Edellisen työn ollessa vielä kesken samaa työntekijää ei voi lähettää uuteen tehtävään. Kiinteistöjen lisääntyessä tehtävien määrä kasvaa, mikä johtaa myös henkilöstön pakonalaiseen kasvattamiseen.

4.3.6. Tilastot

Tilastot (Kuva 15) eivät kuuluneet myöskään alkuperäiseen suunnitelmaan, mutta idea siihen saatiin melko alussa. Tarkoitus olisi nähdä pelaajan tiedot ja ennätykset siistinä listana. Tilastoilla ei pelin kululle ole mitään merkittävää vaikutusta, mutta se nähtiin sopivaksi tällaiseen peliin. Tilastoihin kirjataan myös pelaajan menestys arvotittelillä. Uuden tittelin saa pelaajan kasvattaessa liiketoimiaan. Tietyn arvonimen saavuttaessa peli päättyy.



Player stats	
Player:	
Highest amount of money:	\$998768255
Money earned:	\$166423
Properties bought:	7
Staff hired:	21
Tasks solved:	7
Tasks ignored:	28
Days played:	41
Rank:	Novice

Kuva 15. Pelaajan tilastot

4.3.7. Manuaali

Pelin viimeisessä kehitysvaiheessa luotiin muiden osioiden yhteyteen pienimuotoinen manuaali. Pelissä on paljon asioita, joita voi ensimmäisillä pelikerroilla olla vaikea ymmärtää. Sen vuoksi manuaalin kautta saa apua jokaisen osion käyttämiseen. Peli siirtyy aloituksesta suoraan manuaaliosioon, joten se tulee pelaajalle heti tutuksi. Manuaali on liitetty muiden osioiden tapaan peli-ikkunan vasempaan laitaan, joten siihen voi aina tarvittaessa palata katsomaan ohjeita.

4.3.8. Peli-ikkuna

Lopullinen näkymä pelistä (Kuva 16) on käyttöliittymäsuunnitelman mukainen. Lisäyksiä on runsaasti, mutta toiminnot ovat niille suunnitelluissa paikoissa.

The screenshot displays a game interface with a sidebar on the left containing icons for BANK, MARKET, PROPERTIES, STAFF, TASKS, and STATS. The main area is divided into several sections:

- Owned properties:** Shows details for a property (4/5), including value (\$33018522), type (High-rise), size (2246), occupants (125/182), condition (Good), rent (\$2450), and maintenance (\$4500). It also indicates the status is 'Under maintenance' with 5 days remaining.
- Funds:** Displays financial metrics: Money (\$956379591), Income/month (\$319483), Rents (\$319483), Outcome/month (-\$59815), Interest rate (-\$6208), Salaries (-\$32957), Maintenance (-\$16650), and Total income/month (\$263668).
- Info:** A scrollable log of events:
 - You hired a new worker!
 - \$1898 added to monthly outcome.
 - You fired your worker, outcome reduced by \$1988.
 - You took loan \$58340.
 - You just bought a new property!
 - \$1235328 reduced from your funds.
 - Problem 'Rats, ratholes' is now taken care of. This costs: 343\$
 - You didn't solve the problem 'Broken door lock' in your Duplex! Occupants hired a professional to solve the task and sent the bill
- Edit property:** Includes an 'Advertise' button, a slider for 'Set rent' (2517), and a slider for 'Set maintenance' (4270).
- Current tasks:** A table listing tasks:

Tasks!	Type	Task ID	Status
Task #1:	Rats, ratholes	4	Under work
Task #2:	No task	0	-
Task #3:	No task	0	-
- Date:** Shows the current date as 17.1.2010 and a 'NEXT DAY' button.

Kuva 16. Peli-ikkuna

Vasemmassa laidassa on navigointi eri osioiden välillä. Graafiset painikkeet luotiin kuvankäsittelyohjelmalla. Keskimäinen ikkuna näyttää aina valittuna olevan osion kokonaisuudessaan. Oikeassa laidassa pelaajalle näytetään kaikki pelin seuraamiseen tarvittavat tiedot. Ylimpänä ovat raha-asiat, jotka näyttävät, mihin rahaa kuluu ja mistä sitä saa. Sen alla on info ruutu, joka kertoo pelaajan toimista. Alimpana on päivämäärä ja sitä kasvattava Next day -nappi, jolla siirrytään pelipäivä eteenpäin.

4.4. Elementtien lisääminen

Pelin kehitysvaiheen alussa seurattiin melko tarkkaan alkuperäisiä suunnitelmia. Suunnitelmat tehtiin, kun Qt oli vielä vieras ympäristö, joten sen vuoksi yksityiskohtien luominen jätettiin ohjelmointivaiheeseen. Vasta kun kehitysympäristön mahdollisuudet alkoivat tulla tutuksi ja erilaisia mahdollisuuksia päästiin kokeilemaan, pystyttiin panostamaan yleiskuvan kohentamiseen.

Tekijän kokemuksen puutteen johdosta työmäärä ja tarvittavien ominaisuuksien määrä tuli aikamoisena yllätyksenä. Koodirivejä alkoi olla tietyssä vaiheessa jo reilut kaksituhatta ja välillä saattoi mennä sekaisin, mikä funktio tarkoittaa mitäkään. Koodia joutui useampaan otteeseen korjaamaan, sillä elementtien määrä oli suuri ja niitä ei erityisemmin lajiteltu.

Täydellinen siistiminen koodille ei kuitenkaan sopinut, sillä se olisi vienyt merkittävän määrän aikaa korjata. Tämä olisi myös mahdollistanut runsaan virheiden määrän kasvun, joten helpottavat merkinnät tehtiin koodiin yksinkertaisina kommentteina ja jäsentelyllä. Jokaisella osiolla olivat omat lähdetiedostonsa, mutta silti joissakin tapauksissa rivien määrä tuntui liian suurelta.

Pelin idean ja suunnitelman ollessa yksinkertainen ja avoin uusia ideoita saattoi tulla aina edellisen elementin lisäämisen jälkeen. Peliin pyrittiin lisäämään kaikki parhaat ideat ja ominaisuudet sekä jättämään vähemmän tärkeät asiat taka-alalle. Pelin kehittyessä päätettiin ylläpitää henkilökohtaista To do -listaa, johon kirjattiin ensisijaisesti tärkeät virhekorjaukset ja lisättävät ominaisuudet. Sen alle kirjattiin vähemmän tärkeät korjaus- ja kehitystehtävät, jotka ohjelmoidaan ajan salliessa tai jatkokehityksessä.

4.5. Muutokset

Pelistä oli selkeä visio alusta asti. Yksityiskohtia ei aluksi suunniteltu tarkemmin, koska Qt oli tuntematon, eikä ollut varmaa soveltuuko se pelin luomiseen. Pelin käyttöliittymästä tuli suureksi osaksi sellainen kuin sen alun perin pitikin. Ainoa puute on Suomen kartta, joka alkuperäisen idean mukaan olisi pelaajalle apuna kiinteistöjen ostossa ja hallitsemisessa. Kartta jäi kuitenkin lopulta pois, sillä kiinteistö-osion ohjelmoimisen jälkeen sillä ei ollut enää järkevää käyttötarkoitusta.

Vaikka käyttöliittymä oli suunnitelmien mukainen, se sai kuitenkin runsaasti lisäyksiä. Peliä nopeuttavia navigointipainikkeita luotiin ja pelaajalle pyrittiin pitämään kaikki tarvittava tieto aina esillä. Rahatilanne näkyy info-osiossa ja siitä näkee tarkalleen, mihin rahaa käytetään ja mistä sitä saadaan pelkän yleisen summan näyttämisen sijasta. Inforuutu säilyi suunnitelmien mukaisesti, ja se on jatkuvasti pelaajalle näkyvissä, mutta se sai myös toiminnon tyhjentää ylimääräinen tieto tarvittaessa pois näkyvistä.

Alkuperäiseen suunnitelmaan lukeutui myös kiinteistöjen luominen manuaalisesti. Tarkoitus oli tehdä muutama kiinteistö pelaajan haltuun ja jos aikaa jää, niitä tehdään lisää. Järkevämmäksi vaihtoehdoksi ohjelmoitiin automaattiset ja satunnaisilla arvoilla olevat kiinteistöt. Tämä onnistui melko yksinkertaisella satunnaislukugeneraattorilla, joka arpoo kiinteistön tyyppin, kunnan, hinnan, asukasmäärän ja koon. Näin jokaisella kiinteistöllä on erilaiset ominaisuudet jokaisella pelikerralla. Tätä samaa tekniikkaa hyödynnettiin myös henkilöstön palkkaamisessa.

Kun idea kiinteistöissä ilmenevistä ongelmista syntyi, oli alun perin tarkoitus valita työhön lähetettävät työmiehet käsin. Pelin alkuvaiheilla tämä tuntui sopivalta, sillä työhön sai valita parhaat mahdolliset työntekijät. Testien perusteella kuitenkin huomattiin, että kiinteistöjen sekä henkilöstön kasvun myötä jatkuvasta valintaongelmasta tuli rasite. Tilanne korjattiin laskemalla työntekijöiden taitojen ja motivaation yhteenlaskettu taso, joka laskee vaikutukset tehtävän onnistumiseen. Tällä tavalla pelaajalta jäi yksittäinen valitseminen kokonaan pois, mutta kaikkien työntekijöiden ominaisuudet tehtävässä silti vaikuttavat lopputulokseen.

5. TESTAUS

Yksikkötestausta suoritettiin jatkuvasti, jokaisen uuden toiminnon lisäämisen jälkeen. Pelissä oli useita toisiinsa vaikuttavaa asiaa, että ensin saattoi saada vaikutuksen täysin toimivasta ominaisuudesta. Myöhemmin kuitenkin saattoi ilmetä sen aiheuttavan ristiriitaa jossakin toisessa luokassa, jonka korjaaminen aiheutti virheen jälleen toisessa osiossa. Tämän vuoksi tehtiin myös integrointitestausta.

Testauksessa löydettyjen virheiden kirjaamiseen käytettiin aiemmin mainittua muistilistaa, johon kirjoitettiin virhe tärkeysjärjestyksessä sen ilmetessä. Listasta oli helppo siirtyä ongelmasta toiseen ja merkitä virhe korjatuksi. Kattavin testaus oli pelin esittelyversion valmistumisen jälkeen.

Pelin ollessa esittelyvalmis vaikeusasteen säätäminen oli ohjelmointivirheiden jälkeen ensisijainen tehtävä. Vaikeusaste pidettiin mielessä ohjelmoinnin aikana, joten sen muuttaminen ei ollut haastavaa. Vaikeutta voitiin säädellä muuttamalla kiinteistöjen vuokra- ja huoltorajoja, ongelmien määrää tai pankin maksimilainaa.

5.1. Merkittävimmät ongelmat

Ongelmia projektin aikana oli runsaasti. Kaikki ongelmat ilmenivät ohjelmoinnissa. Jokin yksinkertainenkin asia saattoi viedä jopa useampia päiviä, ennen kuin sopiva ratkaisu löydettiin. Suurin osa ongelmista ja virheistä oli kuitenkin liian pieniä mainittavaksi. Jos jokin ongelma näytti vievän erittäin paljon aikaa, turvaututtiin kysymään apua foorumeilta tai siirryttiin toistaiseksi ohjelmoimaan jotakin muuta. Eniten aikaa vievistä ongelmista kerrotaan seuraavaksi.

5.1.1. Signaalit ja slotit

Signaalit ja slotit ovat Qt:n mainio tapa yhdistää kaksi tai useampaa asiaa toisiinsa. Esimerkiksi liukupalkki voidaan yhdistää yhdellä koodirivillä numerolaatikkoon. Näin liukupalkkia liikuttamalla luku lukulaatikossa pienenee tai kasvaa riippuen mihin suuntaan liukupalkkia vierittää. Signaalit ja slotit olivat helppoja käyttää ja niitä esiintyykin runsaasti lopullisessa pelissä.

Ongelmia esiintyi ohjelmoinnin alkuvaiheilla, kun pankkiosiota yritettiin siirtää pelaajan asettamaa lainan määrää info-osioon. Tämä vaati kommunikointia kahden eri luokan välillä. Ongelma saattoi olla enemmänkin C++:n taidoissa kuin Qt:n hallinnassa, mutta ratkaisua ongelmaan ei löytynyt useampaan päivään. Muiden osioiden kehitystä pystyttiin suorittamaan vaihtoehtoisesti, mutta yhteyden luominen luokkien välille oli välttämätöntä, sillä kaikki luokat ovat lupulta lähes jatkuvassa yhteydessä toisiinsa.

Tämä oli ensimmäinen kerta, kun jouduttiin turvautumaan keskustelupalstan puoleen. Aiheeseen löytyi sopiva Qt-foorumi, johon haettiin apua kyseiseen ongelmaan. Erityisesti apua tarjosi eräs ohjelmoija, joka jopa lähetti avuksi pienen esimerkkiohjelman, jossa tarvittua toimintoa käytetään. Menettelytapa oli esimerkissä erilainen, mutta sen avulla ratkaisu yhteyden luomiselle lopulta onnistui ja työ etenikin sen jälkeen vauhdilla. Varsinainen ongelma oli luokkien sekä signaalien ja slotin oikeatapainen esittely. Signaalien ja slottien käyttö opittiin tämän virheen kautta, eikä vastaavaan ongelmaan enää päädytty projektin aikana.

5.1.2. Muistitoiminto

Kiinteistöjä ja henkilöstöä pystyi aluksi luomaan satunnaisilla arvoilla nappia painamalla. Täytyi keksiä jokin tapa, millä peli muistaa tarvittaessa edellisen selatun kohteen sekä pelaajan halutessa tallentaa kohteen muistiin. Ensimmäisen idea oli käyttää QMap-toimintoa. QMap on muistitoiminto, joka sisältää ns. avaimen ja sille omistetun datan. Käytännössä se voi tarkoittaa tiedon hakemista asetetun avaimen avulla. Toimintoa sovellettiin aluksi käyttämällä henkilöstön palkkaamisessa työntekijän nimeä avaimena ja muita tietoja - kuten palkkaa ja taitoja - datana.

Lukuisten epäonnistumisien jälkeen kuitenkin turvauduttiin kysymään jälleen apua Qt-foorumilta. Aiheeseen saatiin pari ohjetta, mutta ne eivät soveltuneet täysin ongelmaan ja ratkaisu jäi löytämättä. Muistitoiminnon puutteen vuoksi ohjelmointi tässä osiossa jätettiin taka-alalle ja muiden osioiden työstäminen aloitettiin. Myöhemmin ongelmaan palatessa päätettiin aloittaa puhtaalta pöydältä ja katsoa ongelmaa aivan uudesta suunnasta. Erään opiskelijaystäväen ehdotuksena päädyttiin tutkimaan QList-toimintoa.

QList vaikutti erittäin yksinkertaiselta tavalta ylläpitää tietoa muistissa. Ensiksi luotiin luokka ominaisuuksineen. Esimerkiksi luokka ”työntekijä” sisälsi sen ammatin, taitotason, motivaatiotason ja palkan. Jokainen työntekijälle arvottu ominaisuus tallennettiin QList-muistiin, joka pystyi muistamaan useamman työntekijän eri ominaisuudet niiden lisäysjärjestyksessä.

Tallennettuja tietoja pystyi tarkastelemaan syöttämällä halutun ominaisuuden perään hakasuluissa järjestysnumero, joka kasvaa aina uuden tiedon lisäämisen yhteydessä. Tällä tavalla työntekijöitä ja kiinteistöjä pystyi selaamaan; pienentämällä ja kasvattamalla järjestysnumeroa. Tiedon poistaminen tai siirtäminen onnistui myös järjestysnumeroiden avulla. QList-toimintoa käytettiin lopulta myös kiinteistö- ja tehtäväosiossa.

5.1.3. Todennäköisyyslaskurit

Pelin toiminnot ovat usein sattumanvaraisia, joten se sisältää myös paljon todennäköisyyslaskureita. Useimmissa tapauksissa käytettiin yksinkertaista keinoa arpomalla jokin luku, joka vastaa jotakin pelin ominaisuutta. Tätä käytettiin mm. kiinteistöjen ja henkilöstön ominaisuuksia arvottaessa.

Haastavin todennäköisyyslaskuri oli kuitenkin huoltorahan vaikutus ongelmien esiintymiseen. Ensimmäinen kokeellinen todennäköisyyslaskuri jäi puutteelliseksi. Se lisäsi asetettuun huoltorahaan tietyn arvon, jonka jälkeen jäljellä oli alkuperäinen pelaajan asettama huoltoraha ja vasta saatu ”korotettu” huoltoraha. Tämän jälkeen peli arpoi satunnaisen luvun nollan ja korotetun huoltorahan väliltä ja jos tämä luku osui alkuperäisen ja korotetun huoltorahan välille, ongelma ilmeni. Idea alun perin näytti toimivalta mutta tarkemmin tutkittuna ilmeni, että se antoi vain muutaman prosentin skaalan ongelmien ilmenemiselle. Jotakin uutta siis piti keksiä tilalle.

Alkuperäistä ideaa ei kuitenkaan kokonaan hylätty. Useiden muuntelujen ja yritysten jälkeen kehittyi pieni yhtiö, joka antoi alkuperäisen idean tapaan korotetun huoltorahan, mutta tällä kertaa se määräytyi aina alkuperäisen huoltorahan mukaan. Mitä suurempi huoltoraha, sitä enemmän sitä korotetaan vertausta varten. Tämä mahdollisti ongelmien todennäköisyyksien 60–95 %:n välille alkuperäisen idean vertauskaavalla, joka oli juuri sopiva.

Samantyylistä yhtiötä käytettiin myös asukkaiden halukkuuteen muuttaa kiinteistöön. Tässä tapauksessa vain asukkaita piti muuttaa enemmän, kun vuokra on pieni ja vähemmän, kun vuokra on suuri. Pienillä muutoksilla yhtiö saatiin toiminnolle sopivaksi. Vuokran määrä vaikuttaa prosentuaalisesti kiinteistöön muuttavien asukkaiden määrään, mutta kiinteistö ei tule koskaan kerralla täyteen. Jatkuva kiinteistön mainostus johti lopulta aiemmin mainittuun uuteen ominaisuuteen: automaattiseen mainostukseen.

5.2. Systeemi- ja hyväksyntätestaus

Ennen lopullista pelin esittelyä peliä suoritettiin ystävien avuin. Virheitä ei ollut ennen testivaihetta tiedossa lainkaan, mutta aikaisimpien kokemusten perusteella niitä yleensä vielä jälkeinpäin löytyy. Huomaamatta jääneet virheet korjattiin ja peli oli lopulta esittelyvalmis. Tämä ei kuitenkaan tarkoittanut, että peli olisi täysin valmis, sillä ideoita pelin kehittämiseksi olisi vielä runsaasti. Silti projekti täytti opinnäytetyön vaatimukset hyvin ja peli oli saatava arvioitavaksi.

Viimeistelyyn kuului myös dokumenttien ja kaaviokuvien tekeminen. Ohjelmointityön jälkeen keskityttiin UML-kaavioiden täydentämiseen. UML-kaavioilla kuvattiin pelin rakennetta, käyttäytymistä ja vuorovaikutusta. Nämä ovat insinööriydessä yleisiä tapoja kuvata ohjelmiston käyttäytymistä. UML-kaaviot olivat lähes täysin uusi asia kirjoittajalle, joten niiden tekeminen oli aikaa vievää.

6. YHTEENVETO

Peliprojekti oli erittäin kiinnostava aihe tekijälleen. Koska pelit ovat kiinnostavia aiheita, oli helppo pitää myös kiinnostusta yllä projektin aikana. Koko projekti oli valtava oppimisaskel; suurempi kuin mikään koulussa opittu. Opin määrä oli suuri sekä C++:n että Qt:n hallinnassa. Työn määrä oli arvelujen mukaisesti runsas. Opiskelu aiheeseen aloitettiin huhtikuussa ja ohjelmointi loppui joulukuussa. Dokumentointiin meni reilu kuukausi. Ajallisesti suunnitelmat siis pitivät paikkansa.

Kiinteistökauppapelin kehitys jatkuu vielä koulun jälkeenkin, sillä ideoita on vielä peliin paljon. Vaikka peli on ainoastaan yksityiseen käyttöön, sen tekeminen tuntui palkitsevalta ja hauskalta. Tämä projekti myös innoitti aloittamaan muitakin kokeiluja Qt:n parissa, esimerkiksi erään kurssin harjoitustyönä ohjelmoitiin korttipeli Qt-kehitysympäristössä.

Projekti oli kaikin osin onnistunut. Se saavutti lopulta ohjelmistosuunnalle tärkeän kynnyksen sopivana ohjelmointityönä ja enemmänkin. Peli sisälsi enemmän asioita kuin sen alun perin piti, mikä on vain positiivinen asia. Aikaa projektille oli runsaasti ja sen vuoksi pelin kasvu alkuperäisestä suunnitelmasta oli mahdollista.

Ainoana huomiona tulevaisuuden projekteja ajatellen on syytä panostaa enemmän suunnitteluun ja koodin yleissiisteyteen. Tässä projektissa suunnitelmat olivat suhteellisen avoimia, joten lisäyksiä tuli paljon. Tarkempi suunnittelu voisi auttaa ehkäisemään ideavyörymää ja pitämään tehtävien määrän tasaisena. Koodin jäsentely oli projektissa aluksi puutteellista, koska ensin ei ollut käsitystä lopullisesta koodin määrästä. Tulevaisuudessa näihin varaudutaan varmasti paremmin.

Opinnäytetyön tekeminen oli haastavaa, mutta palkitsevaa. Qt oli erittäin positiivinen yllätys ja C++:n opetteleminen tulee varmasti hyödyksi. Qt:n parissa voisi mielellään työskennellä jatkossakin ja varmasti jonkinlaisia projekteja sillä tullaan tulevaisuudessa tekemään.

7. LÄHDELUETTELO

/1/ Blanchette, Jasmin, Summerfield, Mark, C++ GUI Programming with Qt.4.2nd Edition, Prentice Hall, 2008.

/2/ Qt-foorumi, [WWW-sivusto], [<http://www.qtforum.org/forum/2/qt-programming.html>], 01.05.2010.

8. LIITELUETTELO

- LIITE 1 Käyttöliittymäsuunnitelma
- LIITE 2 Pankkiosion suunnitelma
- LIITE 3 Kiinteistöosion suunnitelma
- LIITE 4 Työntekijäosion suunnitelma
- LIITE 5 Tehtäväosion suunnitelma
- LIITE 6 Tilakaavio
- LIITE 7 Luokkakaavio
- LIITE 8 Sekvenssikaavio
- LIITE 9 Käyttötapauskaavio

Real Estate Business

Navigation

Bank

Properties

Staff

Tasks?

?

?

Different gamewindow titles

This is the only dynamic window. Through navigation player can change to bank-window or properties-window etc. Navigation- and info-window never change.

Info

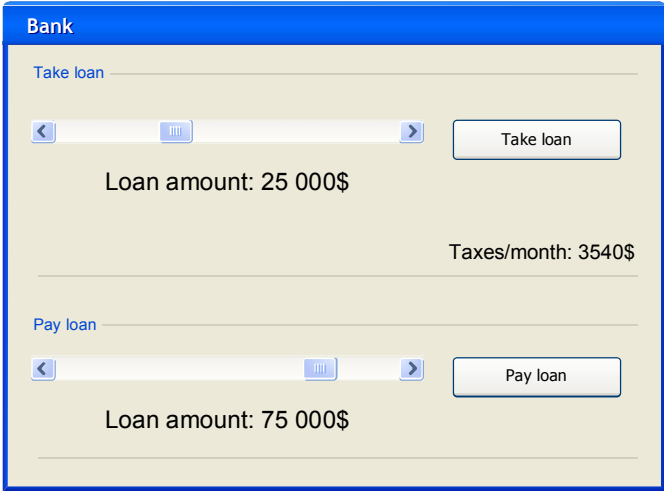
Every piece of info is presented here:

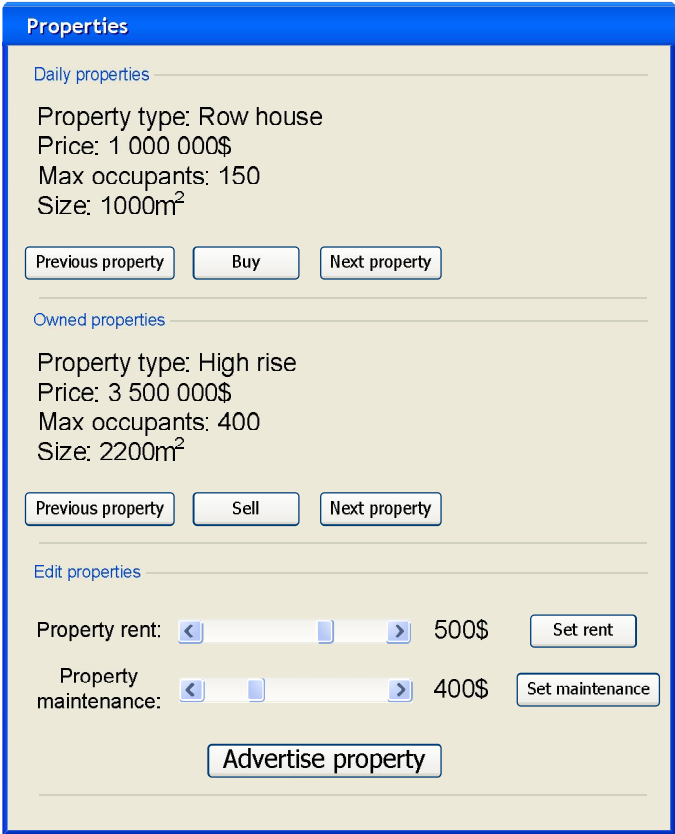
- You bought new property!
- Your raised Bill's salary for 20\$!
- Some occupants moved out from your high rise.
- You took a loan for 200 000\$.

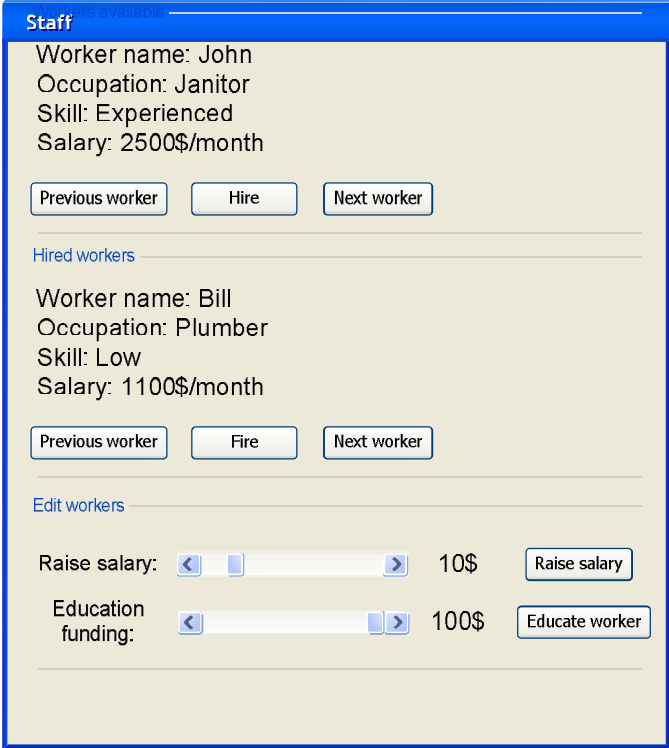
Date: 20.08.2010

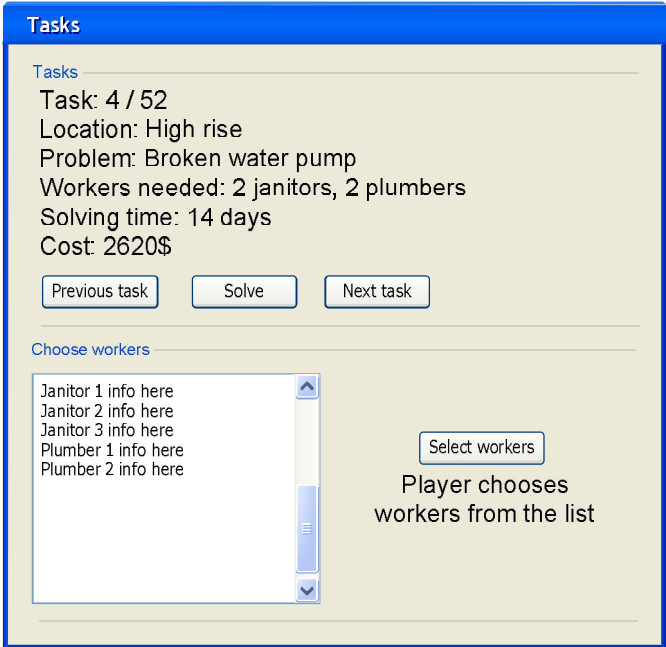
Next day

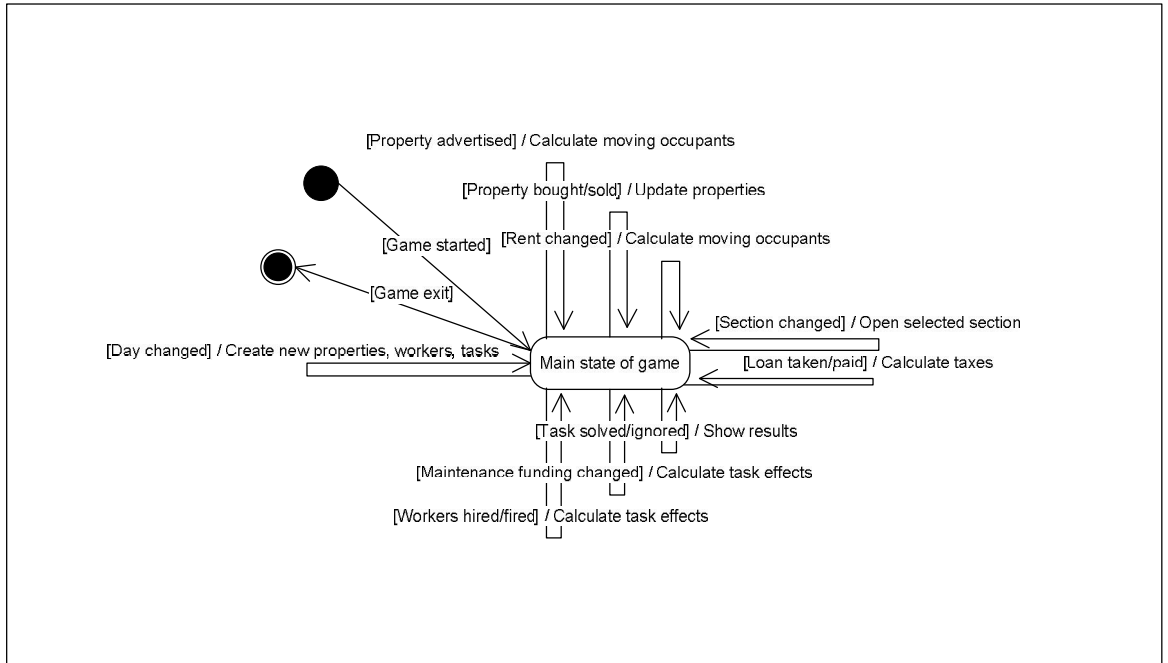
Author: Lassi Laukkanen	Title: UI plan	V.1.0
	Date: 01.01.2011	

		
Author: Lassi Laukkanen	Title: Bank plan	V.1.0
Project: Kiinteistöpelin suunnittelu ja toteutus	Date: 01.01.2011	

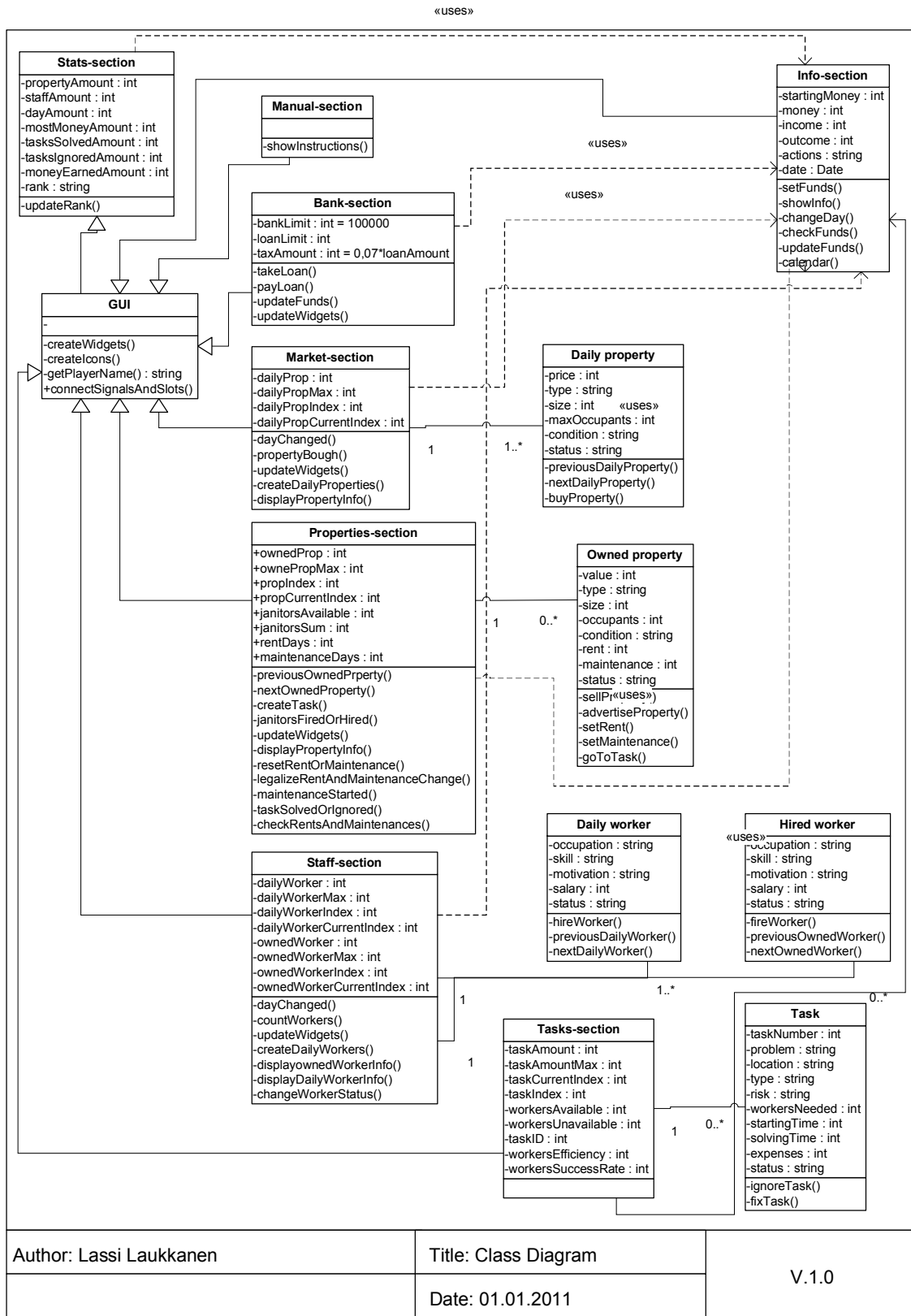
		
Author: Lassi Laukkanen	Title: Properties plan	V.1.0
Project: Kiinteistöpelin suunnittelu ja toteutus	Date: 01.01.2011	

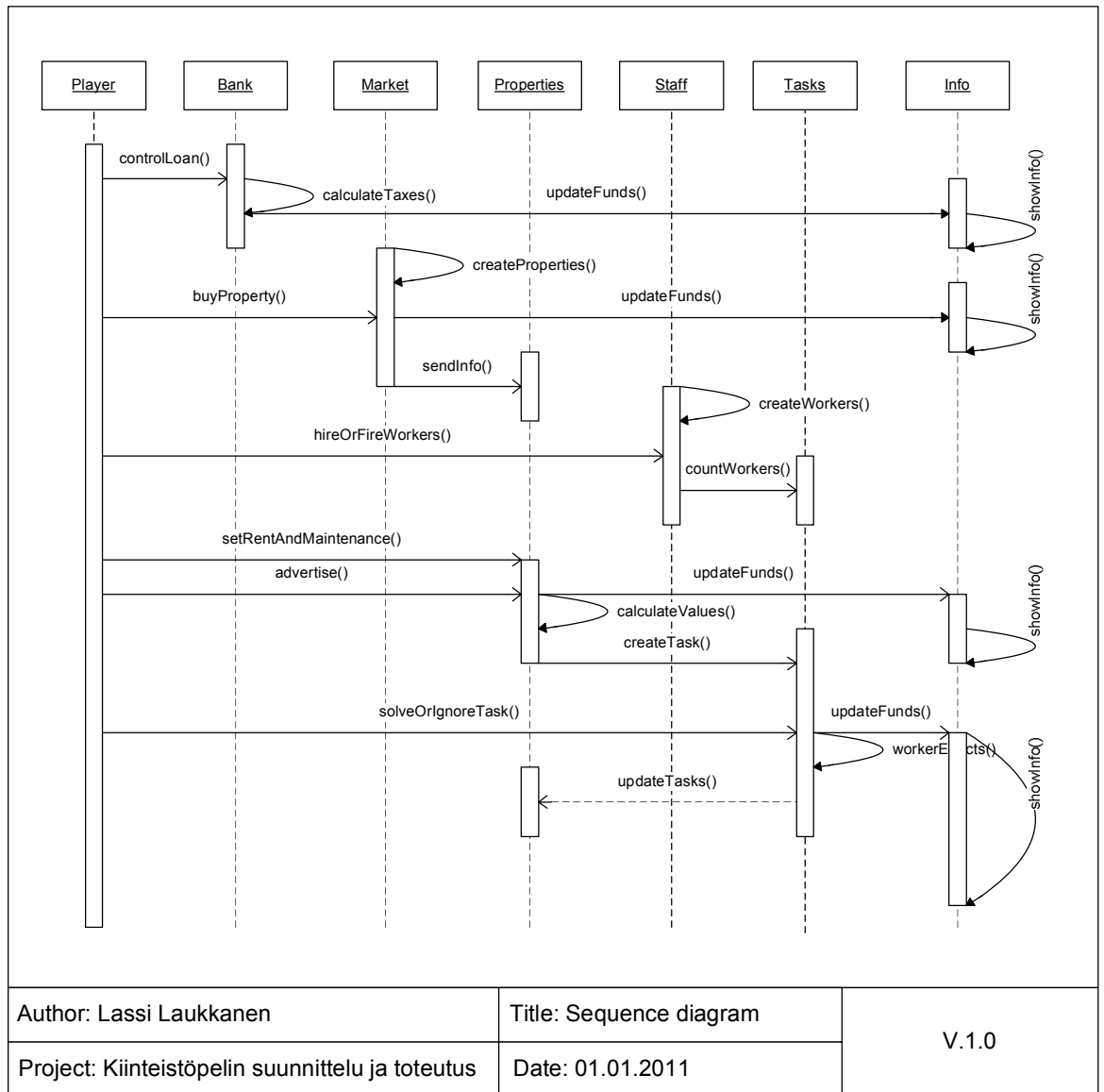
		
Author: Lassi Laukkanen	Title: Workers plan	V.1.0
Project: Kiinteistöpelin suunnittelu ja toteutus	Date: 01.01.2011	

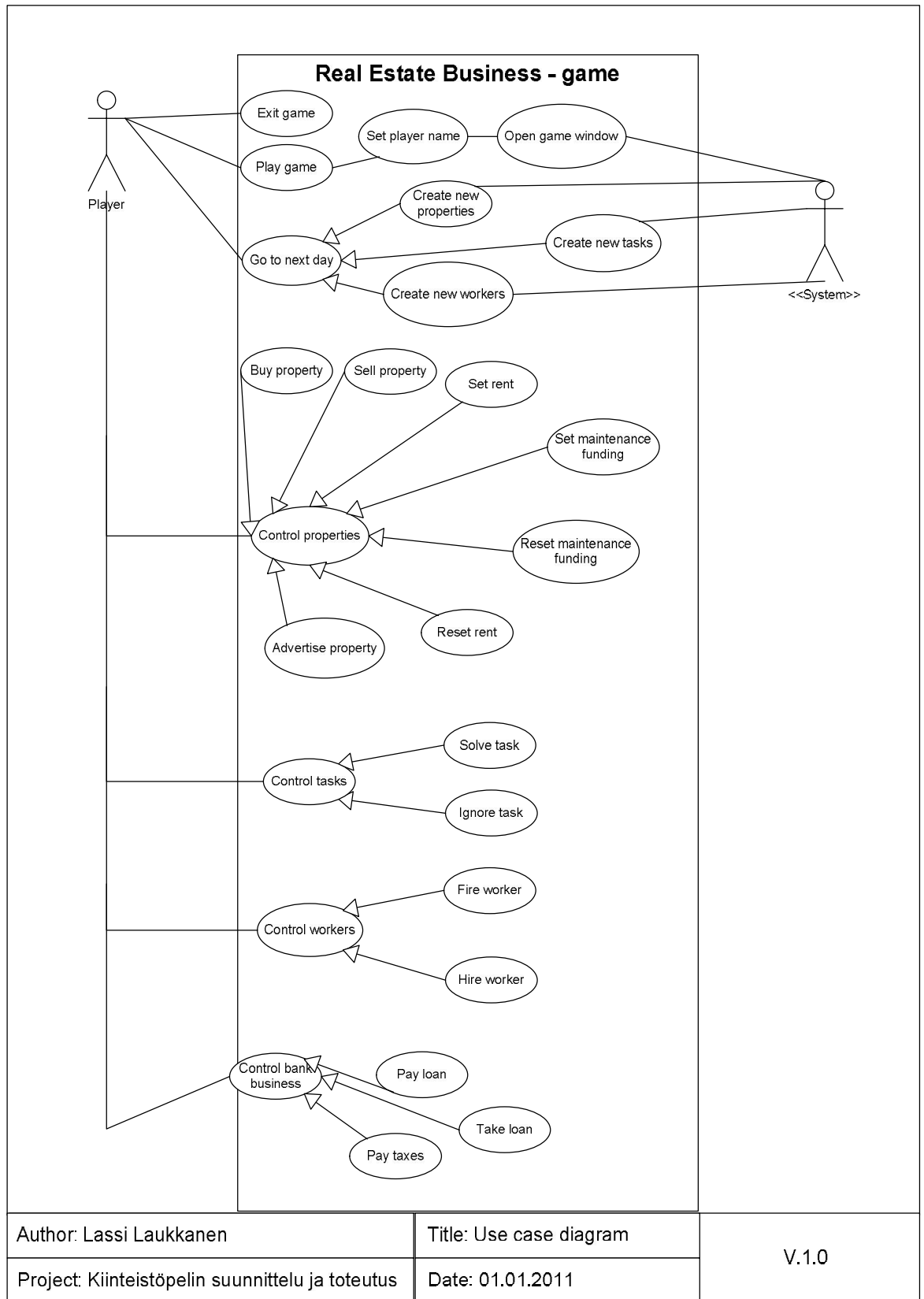
		
Author: Lassi Laukkanen	Title: Tasks plan	V.1.0
Project: Kiinteistöpelin suunnittelu ja toteutus	Date: 01.01.2011	



Author: Lassi Laukkanen	Title: State diagram	V.1.0
Project: Kiinteistöpelin suunnittelu ja toteutus	Date: 01.01.2011	







Author: Lassi Laukkanen

Title: Use case diagram

V.1.0

Project: Kiinteistöpelin suunnittelu ja toteutus

Date: 01.01.2011