



Kandidatnummer:

4601, 4602, 4604, 4605

(Hvis besvarelsen skal leveres på navn, skrives navn i stedet for kand.nr.)

FORSIDE

ved besvarelse av hjemmeeksamen, semesteroppgave, rapport, essay m.m.

Emnekode:	<i>IS-210</i>
Emnenavn:	<i>Systemutvikling med fri programvare (Open Source)</i>
Emneansvarlig (normalt faglærer):	<i>Janis Gailis</i>
Eventuell veileder:	
Innleveringsfrist/ tidspunkt:	<i>27.11.2009 kl. 23:59</i>
Antall sider inkl. forside:	<i>13</i>
Merknader:	

Jeg/vi bekrefter at jeg/vi ikke siterer eller på annen måte bruker andres arbeider uten at dette er oppgitt, og at alle referanser er oppgitt i litteraturlisten.

Ja ☒

Nei ☐

Kopiering av andres tekster eller annen bruk av andres arbeider uten kildehenvisning, kan bli betraktet som fusk.

Gjelder kun gruppeeksamen:

Vi bekrefter at alle i gruppa har bidratt til besvarelsen

Ja ☒

Nei ☐

Kan besvarelsen brukes til undervisningsformål?

Ja ☒

Nei ☐

IS-210

The Final Report

27/11/2009

Innholdsfortegnelse

Innledning	2
Lisenser	3
Utviklingssamfunn	3
Testing.....	4
Implementasjon.....	5
Git	5
Subversion.....	5
Programmeringsverktøy.....	6
Kommunikasjonsverktøy	6
Refleksjon	7
Kilder	8
Vedlegg: Brukerveiledning	9

Innledning

Transport Tycoon ble utviklet av Chris Sawyer og utgitt i 1994, og allerede i 1995 kom oppfølgeren Transport Tycoon Deluxe som i bunn og grunn er det samme spillet bare med en rekke forbedringer. Spillet er et strategispill og går ut på å starte et eget transportselskap og transportere varer mellom forskjellige lokasjoner ved hjelp av tog, båt, fly eller bil. Spillet er et såkalt sandkassespill, som betyr at det ikke har noen slutt¹.

Et av gruppe medlemmene våre har mye erfaring som bruker av spillet OpenTTD², som er en open source-klone av spillet Transport Tycoon Deluxe. Han foreslo derfor at vi kunne bruke denne oppgaven til lage noe til dette spillet, for eksempel et nytt togsett eller noe lignende. Vi startet prosessen med å legge inn en ny tråd i support- og utviklingsforumet til spillet³. Før det var gått et døgn satt vi med ti mer eller mindre gode svar. Ønsket var en tilbakemelding på om spillet kunne egne seg til en slik prosjektoppgave og hva vi i tilfelle burde/kunne fokusere på.

Togdesign viste seg å være en dårlig idé da vi ville måtte bruke lang tid på selve utseendet og kodingen ville bare blitt en slags config-fil. Valget falt til slutt på utvikling av en ny Artificial Intelligence (AI). Fra og med versjon 0.7 av OpenTTD ble det utviklet et rammeverk for utvikling av AI, kalt NoAI. For vår del vil det si at vi ikke trenger å bruke mye tid på og sette oss inn i koden fra selve kjernen av spillet, men bare en mindre del. NoAI er i tillegg svært godt dokumentert fra utviklernes side. Ved å lage en AI vil det også være gode muligheter for at utviklingssamfunnet tester den og forhåpentligvis videreutvikler eller låner kode fra den.

¹ <http://www.chrissawyergames.com/info.htm> (23.11.2009)

² <http://www.openttd.org/en/> (23.11.2009)

³ <http://www.tt-forums.net/viewtopic.php?f=29&t=45561> (23.11.2009)

Lisenser

Innenfor FOSS (Free and Open Source Software) finnes det to store miljøer som styrer og godkjenner lisenser for bruk til å lansere fri og open source programvare. Free Software foundation (FSF) og Open Source Initiative (OSI) passer på at lisensene tar hensyn til henholdsvis FSFs fire friheter⁴ og OSIs definisjonsliste⁵. Selv om miljøene har forskjellig ideologi og arbeidsmåte er de enig om de fleste kjente lisensene som er i bruk. Det viktigste innen lisensiering for FOSS er at kildekoden skal forbli åpen og tilgjengelig og at brukerne har frihet til å gjøre hva de vil med den.

I begynnelsen, på midten av 80-tallet lanserte GNU-prosjektet hver programpakke med sin egen individuelle fri programvarelisens, men disse ble i 1989 byttet ut med den første versjonen av GNU GPL. Andre versjon av GNU GPL ble lansert i 1991 og har blitt den mest brukte lisensen for fri programvare. Tredje versjon av GPL ble lansert i 2007.

Et viktig begrep innen GNU GPL-lisensene er Copyleft. Copyleft er på en måte grunnessensen i fri programvare-lisensene og tillater at programvaren fritt kan endres, distribueres og reproduseres så lenge det nye arbeidet lanseres under samme lisens som det originale arbeidet.

OpenTTD gis i dag ut under GPLv2, og så og si alle AI'ene som allerede er utviklet gis ut under GPLv2. Deler av vår AI består av kode utgitt under GPLv2 og for å unngå kompatibilitetsproblemer⁶ velger vi også å gi ut vår AI med denne lisensen.

Utviklingssamfunn

OpenTTD har vært under utvikling i flere år allerede (noen eksakt startdato er uklart, men vi fant forumoppføringer datert tilbake til mars 2003) og har et stort og aktivt utviklingssamfunn⁷ som står for utviklingen av spillet. All aktivitet knyttet til utviklingen foregår på forumet og via en IRC-kanal. Det som til slutt blir vedtatt blir så dokumentert på en wiki. Spillet er i skrivende stund i versjon 0.7 som betyr at det ikke er komplett. Det kreves at man laster inn lyd og

⁴ <http://www.gnu.org/philosophy/free-sw.html> (23.11.2009)

⁵ <http://www.opensource.org/docs/definition.php> (23.11.2009)

⁶ <http://www.gnu.org/licenses/license-list.html#GPLCompatibleLicenses> (23.11.2009)

⁷ <http://www.tt-forums.net> (23.11.2009)

grafikkfiler fra originalversjonen, men selve spillmotoren er ferdig og en rekke moduler jobbes med eller er ferdig. Det er derfor mye relevant som kan utvikles fra vår side.

Siden kildekoden til spillet er på godt over 2GB blir dette såpass mye å sette seg inn i at vi ville brukt hele prosjektperioden på å forstå den. Vi spurte derfor om råd og tips på forumet, og fant ut at å jobbe med Artificial Intelligence⁸ er det som best passer våre forutsetninger. På denne måten kan vi jobbe med en modul der det finnes materiale som vi kan redigere, eller vi kan lage en ny AI mer eller mindre basert på det som finnes fra før. Som nevnt i innledningen har man nå et eget rammeverk i kildekoden (NoAI) som gjør integreringen av hjemmelagde AI meget enkel. Bibliotek med for eksempel veifinnere og debugger er inkludert i rammeverket, noe som gjør vår jobb mye enklere. En AI programmeres i Open Source-språket Squirrel som i følge utviklerne av OpenTTD skal ligne mye på C++ (språket som brukes ellers i spillet), men også ha elementer fra Java i seg.

En AI kan være så mangt, men i vårt tilfelle en motstander (eller medspiller) som lager enkel infrastruktur og busstopp mellom byer. Hensikten med dette er å øke størrelsen på byene. Mer aktivitet i tilknytning til en by betyr flere innbyggere, som igjen betyr høyere inntekter. Det finnes til og med dedikerte servere med brukerskapte AI'er som konkurrerer mot hverandre om å ha høyest mulig inntjening. Disse er imidlertid langt mer avanserte enn den vi har laget.

Det finnes også Open Source-verktøy for å redigere og implementere AI som er utviklet av dette samfunnet. Vi skal se nærmere på disse etterhvert som vi kommer i gang med programmeringen.

Testing

Før vi startet med prosjektet var det kun en av gruppemedlemmene som hadde spilt OpenTTD eller Transport Tycoon Deluxe tidligere. På grunn av dette og fordi vi ikke skal gjøre noen endringer i selve spillet så vi ikke noen verdi av å teste det på eksterne testpersoner. Alle gruppemedlemmene måtte imidlertid sette seg godt inn i spillet før vi kunne starte med programmering.

⁸ http://wiki.openttd.org/AI:Main_Page (23.11.2009)

En fordel med å utvikle en AI er at dette er noe som skal virke uten innvirkning fra oss når den er ferdig. Testingen vår består derfor av å starte AI'en og se hvordan den oppfører seg, også sammen med andre AI'er. Vi trenger derfor ingen testpersoner i vårt prosjekt, noe som har spart oss for en del tid.

For at sensorene skal få testet koden vår har vi lagd en brukerveiledning som ligger som vedlegg til rapporten.

Implementasjon

Git

Vi bestemte oss i første omgang å bruke Git som revisjonskontroll. Vi opprettet hver vår konto på GitHub, og prøvde å samkjøre disse slik at vi alle fikk tilgang på en repository. Etter å ha brukt mye tid på å få det til å fungere, noe vi forsåvidt fikk til, fant vi ut at Git var noe tungtvindt og mer enn hva vi trengte til vårt bruk. Det er ingen tvil om at Git er et kraftig verktøy, men tunge brukerveiledninger og kompliserte rettighetsoppsett førte til at vi valgte subversion i stedet. I tillegg så bruker OTTD-prosjektet subversion til både kildekoden og tilleggsmodulene, så da er det greit at vi også benytter det til vårt prosjekt.

Subversion

Public tilgang: <http://is210-ottd.googlecode.com/svn/trunk/>

Vår repository: <https://is210-ottd.googlecode.com/svn/trunk/>

Prosjektside: <http://code.google.com/p/is210-ottd/>

Til å hoste vårt prosjekt har vi tatt i bruk tjenesten Google Code. Det var enkelt å opprette et nytt subversionprosjekt, og vi var fra tidligere godt kjent med bruken av subversionklientene med GUI.

Etter å ha investert mye tid til å få git til å virke var det en letteste å kun bruke et kvarter på å få subversion klart til bruk.

1. Logget inn på Google og gikk til Google Code.

2. Opprettet nytt prosjekt og la inn informasjon om prosjektet. La også inn alle gruppemedlemmene som eiere av prosjektet.
3. Installerte TortoiseSVN og utførte checkout av prosjektet.

Programmeringsverktøy

Visual Studio var det anbefalte verktøyet fra utviklingssamfunnets side, men det var nok mest ment hvis man skulle være med på prosjektet å endre på kildekoden. Noen av gruppemedlemmene installerte dette, men endte opp med Notepad++⁹ da vi fant ut at et komplekst verktøy som Visual Studio ikke var nødvendig til oppgaven med å lage AI-er. En av grunnene er at AI'en skrives i Squirrel, og en trenger ikke egen kompilator for å teste koden, men derimot selve spillet.

Eclipse¹⁰ med SQDEV plugin(Squirrel-plugin)¹¹ er også et alternativ til verktøy. Riktignok gjenkjenner ikke pluginen alle definisjonene av Squirrel, og med litt god kodeskikk er Notepad++ et mer egnet verktøy for vår oppgave.

Kommunikasjonsverktøy

Til skrivingen av rapporten bruker vi MediaWiki (som ble satt opp i IS-209). Dette gjør at alle til enhver tid har alt vi har skrevet tilgjengelig og alle har alltid nyeste utgave tilgjengelig. Vi har valgt å legge til en extension kalt Cite som gir oss en enkel måte og føre opp referanser i dokumentene våre. Først når oppgaven var ferdig kopierte vi innholdet inn i et tekstdokument.

Mesteparten av arbeidet vil vi gjøre som en gruppe på skolen da vi synes dette gir oss best mulighet til et godt samarbeid og involvering av alle medlemmene. Skal vi imidlertid kommunisere uten å være i samme rom bruker vi direktemeldingsprotokollen MSN og eventuelt e-post.

Online versjon av rapporten kan ses på <http://158.36.166.73:8080/mediawiki/index.php/Is-210>.

⁹ <http://notepad-plus.sourceforge.net/uk/site.htm> (23.11.2009)

¹⁰ <http://www.eclipse.org/> (23.11.2009)

¹¹ <http://wiki.squirrel-lang.org/default.aspx/SquirrelWiki/SQDEV.html> (23.11.2009)

Refleksjon

Vi har utviklet et frittstående tillegg til et eksisterende spill. Dette gjør at videreutvikling ikke er nødvendig for at vårt produkt skal kunne brukes. Det betyr imidlertid ikke at det ikke kan videreutvikles, og vi vil legge opp til at hvem som helst kan forbedre vår AI når dette faget er ferdig. Vi har derfor lagt vekt på at hovedklassen vår er godt dokumentert slik at det vil være lett å forstå hva vi har gjort for en eventuell ny utvikler. AI'en har begrensede funksjoner, men gir oss et godt utgangspunkt for å lære mer for å til slutt kunne lage en fullverdig AI. Ting som vi ser nå at AI'en gjerne skulle inneholde er en bedre finansiell håndtering slik at alt lån blir tilbakebetalt når firmaet har penger til det, og en bussbehandlingsfunksjon som bytter ut bussene når de blir gamle og slitte.

Slik som samfunnet rundt OTTD er nå så skal det ikke være noe problem og kunne la prosjektet leve videre, enten ved at AI-en legges ut på utviklerforumet slik at alle kan ta en del av videreutviklingen, eller at vi selv fortsetter og tilpasse den til våre ønsker og behov. I tillegg til at en av gruppemedlemmene på forhånd brukte mye tid på dette spillet har nå også flere i gruppen fått øynene opp for mulighetene i OTTD og vil kanskje i fremtiden bruke tid på å perfeksjonere en AI. Oppgaven har også gitt oss et innblikk i hvordan utviklingssamfunn kan foregå, om hvordan folk samarbeider og deler kunnskap og kode.

Som nevnt tidligere valgte vi å lage et tillegg til spillet for å unngå at vi måtte sette oss inn i hele kildekoden før vi kunne begynne med noen form for programmering. Grunnen til at valget falt på en AI var fordi dette var noen nytt i utviklingen av OpenTTD og fordi vi ble anbefalt dette av eksisterende bidragsytere. Det at vi kunne prøve oss på et nytt, men likevel kjent språk, Squirrel, var også noe som appellerte til oss.

Det endelige produktet vårt i denne sammenhengen er selvfølgelig ikke perfekt, og er ikke klar for å lanseres som ferdig. Vi mener likevel at vi har kommet langt på vei og det blir interessant å se om vi eller noen andre ønsker å videreutvikle dette slik at det senere kan lanseres som versjon 1 enten under navnet rocketAI eller noe annet.

Kilder

Transport Tycoon Forums [Online] Tilgjengelig:

<http://www.tt-forums.net> [2009, 23. november]

OpenTTD (2009) [Online] Tilgjengelig:

http://wiki.openttd.org/AI:Main_Page [2009, 23. november]

Wikipedia (2009). Free Software Licence. [Online]

Tilgjengelig: http://en.wikipedia.org/wiki/Free_software_licence [2009, 23. november]

Wikipedia (2009). GNU General Public Licence. [Online]

Tilgjengelig: http://en.wikipedia.org/wiki/GNU_General_Public_License [2009, 23. november]

Wikipedia (2009). Copyleft. [Online]

Tilgjengelig: <http://no.wikipedia.org/wiki/Copyleft> [2009, 23. november]

Vedlegg: Brukerveiledning

1. Først trenger vi filene fra originalspillet, Transport Tycoon Deluxe for Windows. Originalen trengs uavhengig om du skal kjøre spillet i Linux eller Windows, men spillet ble opprinnelig laget for MS-DOS og deretter Windows og bærer derfor dette navnet. Disse lastes ned her:

<http://www.owenrudge.net/download/download?t=1&id=129>.

Det er ikke nødvendig å installere Transport Tycoon Deluxe for Windows, men det trengs noen filer fra originalspillet for at Open Source versjonen skal kunne kjøre. Dette er ikke ulovlig nedlasting siden spillet nå er så gammelt at det blir regnet som Abandonware¹².

2. Pakk ut zip filen du nettopp lastet ned i en katalog du enkelt finner igjen.
3. Last så ned siste stabile versjon av OpenTTD, som i skrivende stunde er versjon 0.7.3 her:

<http://www.openttd.org/en/download-stable>.

Vi har testet OpenTTD på Windows 7 64-bit og Ubuntu 32-bit. Spillet finnes i Ubuntus pakkebrønn men er ikke oppdatert der så man må derfor laste ned nyeste versjon manuelt fra OpenTTDs hjemmeside. Vi har btlukt «lenny» versjonen for Debian. I Ubuntu kreves forøvrig root-tilgang til deler av installasjonen. Til Windows anbefales det executable installer.

4. Kjør nå filen du nettopp lastet ned. (openttd-0.7.3-windows-win64.exe eller openttd-0.7.3-linux-debian-lenny-i386.deb)

Du vil i løpet av installasjonen bli bedt om å legge til de filene som trengs fra originalversjonen. Windows: Velg mappe til originalversjon og installeren tar seg av resten. Ubuntu: Kopieringen av de nødvendige filene gjøres manuelt fra mappen du pakket ut

¹² <http://en.wikipedia.org/wiki/Abandonware> (23.11.2009)

originalen, til mappen OpenTTD bruker. Du vil bli henvist til en readme fil (i mitt tilfelle ligger denne i katalogen /usr/share/doc/openttd) som sier hvilke filer du skal kopiere, og hvor de skal kopieres til. Vi opplevde denne prosessen noe tungvinn.

5. For å få i gang vår AI inne i spillet må koden lastes ned fra vår repository. Vi har brukt Google Code og SVN. Adressen til den er som nevnt tidligere <http://is210-openttd.googlecode.com/svn/trunk/>. Det trengs ingen kompilering da spillet tolker selv filene.

Avhengig av hvilket operativsystem du bruker må AI filene legges i en bestemt mappe slik at spillet finner dem. Lag en mappe med filene fra repository og kall den rocketAI. Deretter kopieres denne mappen til riktig sted: Windows: Spillet lager en egen mappe på hjemmeområdet ditt (Mine Dokumenter) /OpenTTD/content_download/ai/. Ubuntu: Det samme gjelder for Ubuntu, men her er OpenTTD-mappen skjult. Så filbanen blir ~/.openttd/content_download/ai. Windows og Linux: Hvis du får beskjed om mappen ikke finnes kan du enten opprette den selv, eller prøve å starte spillet en gang og endre en innstilling (f.eks. trykke på fullskjermknappen) slik at det blir laget noen configfiler som lagres på dette stedet.

6. Siden AI'en vår bruker et bibliotek må dette også lastes ned for å få kjørt AI'en. Det letteste er å bruke OpenTTDs egen «pakkebrønn». Denne kalles BaNaNaS og kan sjekkes ved å starte spillet og deretter trykke på «check online content». Søk etter «Pathfinder.Road», merk denne og velg last ned. Den letteste måten å laste ned vår AI på er også via samme pakkebrønn, men om du har fulgt forrige steg riktig ligger den allerede klar til bruk.
7. Før du starter et nytt spill så må du gå inn på «AI settings» og sett «Maximum no. of competitors» til 0. Dette gjøres for å unngå forstyrrelse fra andre AI'er.
8. Start et nytt spill ved å trykke på «New Game», deretter «Generate» uten å endre andre innstillinger og du er i gang.

9. Spillet har en innebygd AI debugger (som forøvrig også logges i terminalen om du starter spillet fra bash i Linux). Du finner denne ved å trykke og holde inne venstre museknapp på spørsmålsteget i verktøylinjen og velge «AI Debug». I samme meny finner man «Toggle Console» for og åpnet konsollen. Eventuelt kan man bruke paragrafnappen som er hurtigtast.

For å starte AI'en skriver du følgende kommando.

```
start_ai rocketAI
```

10. Deretter er det bare og flytte rundt i kartet for å se på byggingen og følge med i debuggeren. Fordi AI'er operer på såkalte ticks, kan man ikke stille byggefarten på den sånn helt uten videre. Det tar gjerne flere år i spillet før AI'en er ferdig med det det er programmert til å gjøre. Vi anbefaler derfor å trykke på de to små pilene helt til venstre på verktøylinjen (ved siden av pauseknappen) for å få litt fart på sakene. Når så AI'en er ferdig med jobben vil den si ifra om det i debuggeren.