

# 1 Johdanto

Otin tehtäväkseni toteuttaa tekstiseikkailupelille rungon ja samalla lyhyen demonstraatiopelin javalla. Aluksi tarkoitukseni oli tehdä isompi kontekstiton kielioppi ja tämän jälkeen koodata kieliopista tunnistin suoraan käsin ilman minkäänlaisia apuvälineitä. Tämä osoittautukin hyvin ongelmalliseksi, varsinkin silloin kun kieltä muutettiin kesken ohjelman kirjoituksen. Säästääkseni siis itseäni ja luennoitsijaa, päätin perehtyä CUP:iin (Parser Generator for java). CUP osoittautukin upeaksi työkaluksi. Se auttoi suunnattomasti kieliopin rakentamista ohjelmakoodin puolelle sekä tarjosi suoraan ominaisuuksia, joilla pystyttiin vähentämään ohjelmakoodissa tapahtuvaa parsimista.

Kun kielioppi oli saatu valmiiksi ja tätä varten tehtytunnistin käännetty, oli tehtävä maailmassa liikkumiselle jokin tehokas systeemi. Tähän olisikin varsin näppärää toteuttaa DFA (deterministinen äärellinen automaatti), jossa tilat määrittelevät pelin etenemisen. Pelissä voimme kuitenkin ottaa tavaroita haltuun ja ilman niitä emme välttämättä pääse etenemään maailmassa. Miksipä emme siis käyttäisi DFA:sta yleistettyä pinoautomaattia, joka on ilmaisuvoimaisempi kuin pelkkä DFA sellaisenaan.

# 2 Työn esittelyjärjestys

Aluksi käymme läpi yleisellä tasolla lyhyesti koodin luokkarakenteen. Ohjelmakoodin selittäminen jätetään kuitenkin ”tyngäksi”, koska se ei ole kurssin kannalta oleellista. Tämän jälkeen katsastamme läpi peliin toteutetun demonstraatiopelin pinoautomaatin. Lopuksi esittelemme koko kontekstitoman kieliopin produktioineen ja annamme pieniä esimerkkejä kieliopista, sekä näytämme miten pelin ”juostaan” läpi yhdellä syötteellä.

### 3 Koodi lyhyesti

Ohjelman kannalta tärkeät luokat ovat seuraavat:

- Action, Actions
- Case, Cases
- CaseDFA
- Main
- parser, sym, Lexer

Action -olio hoitaa yhden käyttäjän antaman *actionin*. Yhdelle textitactionille tuodaan paramaterina automaatti -olio (CaseDFA), jolloin *actioni* tietää minkä syötteen se antaa automaatille. Syötteet ovat automaatissa aliohjelmina. Case -olio taas käsittelee yksittäisen *tilanteen* maailmassa ja osaa kuvailla tämän. Case -olio osaa myös antaa Action -olion perusteella vastauksen merkkijonona, jos sinne on olion luomisen aikana tällainen *action-answer-pari* määritelty. Sillä on myös tieto *isVisited*, eli onko tilassa oltu jo kerran. Jos ollaan, annetaan joissakin tilanteissa käyttäjälle eri kuvaus kuin alkuperäinen luonnin yhteydessä määritelty kuvaus.

Parser, sym ja Lexer -luokat on toteutettu CUP:ia käyttäen parsimaan määrittelemämme kielioppi. Parser -olion tehtävänä on palauttaa lista kaikista niistä Action -olioista, mitä syntyy käyttäjän antaessa syötettä. Esimerkiksi ”open the door and then take the candle” -syöte sisältää kaksi Action-oliota. CaseDFA -olio on pinoautomaatti, josta tarkemmin seuraavassa kappaleessa.

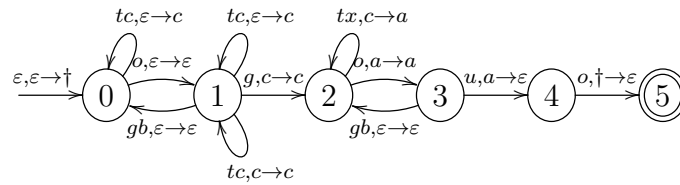
Main -luokka hoitaa yksinkertaisesti pelin käynnistämisen ja pelaamisen. Cases -olio hoitaa maailman alustamisen Case -olioilla.

### 4 Pinoautomaatti

Automaatissa olevat kuusi(6) tilaa eivät ole sama asia kuin yksittäinen Case-olio. Automaatissa erillisillä syötteillä ja pinonmuutoksilla määritellään muut-  
tuuko Case-olio, joka tarjoaa tilanteen kuvauksen. Seuraavassa kaaviossa on piirretty pinoautomaatti demonstraatiopeliä varten. Automaatti ei sisällä kaikkia syötteitä mitä lopussa esittelemämme kielioppi on kykenevä tunnistamaan. Kuten sanottu tämä on vain demonstraatiopeli, jossa osoitetaan kuinka pinoautomaattia voidaan hyödyntää pelin rakentamisessa.

## 4.1 Symbolit ja automaatti

- $\dagger$  = tyhjä pino,  $\varepsilon$  = epsilon
- $o$  = open door,  $tc$  = take candle
- $c$  = candle,  $a$  = axe
- $gb$  = go back,  $g$  = go hallway
- $u$  = use axe,  $tx$  = take axe



## 4.2 Selvitys maailman tiloista ja vaihtoehtoista

0. Herätään huoneesta, vaihtoehtoina oven avaus tai kynttilän ottaminen pöydältä. Päätehtävänä on pelastaa prinsessa, joka on vankina linnassa.
1. Oven voi avata ilman kynttilää, mutta ovesta käytävään ei voi mennä ilman kynttilää.
2. Käytävä mennään automaattisesti päästä päähän, jolloin kynttilä sammuu. Edessä on näkyvissä ovi, jonka vierellä nojaa seinää vasten kirves. Vaihtoehtoina on oven avaus tai kirveen ottaminen haltuun. Ovea ei kuitenkaan voi avata ilman kirvestä, koska sen takana on pelottava örkki. Kirvestä ottaessa kynttilä tiputetaan maahan samalla.
3. Oven avauduttua örkki haastetaan taisteluun. Vaihtoehtoina on tappaa örkki tai paeta takaisin oven taakse.
4. Örkki kuolee suorasta kirveen heitosta, jolloin jäljelle jää vankilan oven avaaminen, jossa oletetusti prinsessa on. Tässä vaiheessa kirvestä ei enää ole kädessä. Vaihtoehtoina oven avaaminen.
5. Karu totuus kuitenkin paljastuu sankarillemme. Vankila on nimittäin tyhjä ja prinsessa on toisessa linnassa.

## 5 Kontekstiton kielioppi

### 5.1 Esimerkkejä kielestä

Pelissä keskustellaan maailman kanssa kirjoittamalla komentoja. Kuten alussa mainitsimme, voi komentoja olla monta yhdessä syötteessä. Yksi komento koostuu yleensä kahdesta osasta: *toiminnosta* ja *kohteesta*, mutta on myös mahdollista, että komennossa on vielä kolmaskin osa: *kohteen kohde*.

Esimerkkejä kaksiosaisista käskystä:

*take axe*  
*take the axe*  
*use a candle*  
*kill the goblin*

Esimerkkejä kolmiosaisista käskyistä (kielessämme mahdollista käyttää vain *TOOL* -tyyppisissä lauseissa):

*use the axe on the goblin*  
*use pillowcase on princess*  
*drop the hammer on workbench*

Esimerkkejä komentojen yhdistelemisestä:

*take the axe and then use the axe on the goblin*  
*use pillowcase on princess then hug princess*  
*kill the goblin and then open the door*

## 5.2 Kontekstiton kielioppi produktioineen

Seuraavaksi esitellään kielioppi produktioineen. Huomaa, että kaikki kieliopin määrittelemät syötteen eivät ole demonstraatiopelissä mukana.

$$E \rightarrow TOOLACTION \ TOOL \ TOOLTARGET \ CONT$$
$$E \rightarrow OBJECTACTION \ OBJECT \ CONT$$
$$E \rightarrow NPCACTION \ NPC \ CONT$$
$$TOOLACTION \rightarrow use \mid take \mid get \mid drop \mid dismiss \mid toss$$
$$TOOL \rightarrow PAS \ candle \mid PAS \ hammer \mid PAS \ axe \mid PAS \ pillowcase$$
$$TOOLTARGET \rightarrow NPC \mid TOOL \mid OBJECT$$
$$OBJECTACTION \rightarrow go \mid gotu \mid open \mid push \mid pull \mid examine \mid look$$
$$OBJECT \rightarrow PAS \ wall \mid PAS \ workbench \mid PAS \ chair \mid PAS \ door \mid PAS \ hallway$$
$$NPCACTION \rightarrow talk \mid talkto \mid kill \mid hug$$
$$NPC \rightarrow PAS \ goblin \mid PAS \ boss \mid PAS \ princess \mid PAS \ gustav$$
$$PAS \rightarrow \varepsilon \mid the \mid a \mid an \mid at \ PAS \mid on \ PAS$$
$$CONT \rightarrow \varepsilon \mid then \ E \mid and \ CONT \mid E$$

Pelin läpäiseminen suoraan onnistuu esimerkiksi seuraavalla syötteellä:

*take the candle then open the door and then go to the hallway then take the axe and open the door and then use the axe on goblin then open the door*

Kieli pyrkii näyttämään mahdollisimman selkokieliseltä. Kuitenkaan ei pois-suljeta mahdollisuutta kirjoittaa vielä alkeellisempiakin lauseita. Pelin siis voi läpäistä myös seuraavalla komennolla:

*take candle open door go hallway take axe open door use axe goblin open door*

## 6 Loppusanat

Harjoitustyö osoittautui varsin mielenkiintoiseksi projektiksi ja antoi erittäin paljon uusia lähtökohtia asioiden tekemiseen. Erityisesti tuli huomattua, kuinka tärkeitä asioita tässä työssä käytetyt metodit ovat. Jälleen tuli itselle todistettua miksi harjoitustyön tekeminen on paljon antoisampaa ja opettavaisempaa kuin pelkän tentin teko. Ohjelmakoodissa olisi voinut panostaa vielä enemmän olio-ohjelmoinnin rakenteeseen, mutta koska tämä ei oikeastaan kuulunut kurssin asioihin, en antanut sille niin suurta painoarvoa. Sama politiikka oli myös testejä sekä kommentoinnin kanssa.

Työ antoi sivussa myös paljon muutakin kuin ainoastaan kontekstittoman kieliopin ja pinoautomaatin soveltamisen ohjelmointiin. Erityisesti tuli perehdyttyä CUP:iin ja tätä kautta oppi myös YACC:ia. Esseen kirjoituksestakin tuli tärkeä osa työtä ja tässä saikin ensikosketusta Latexin ihmeelliseen maailmaan. Erityisesti kaavioiden piirtämisen helppous hämmästytti. Kaikinpuolin kurssi oli hyödyllinen ja mielenkiintoinen.