

Simulering av ett signalställverk i C#

Gymnasiearbete 100 poäng

**Patrik Olsson**

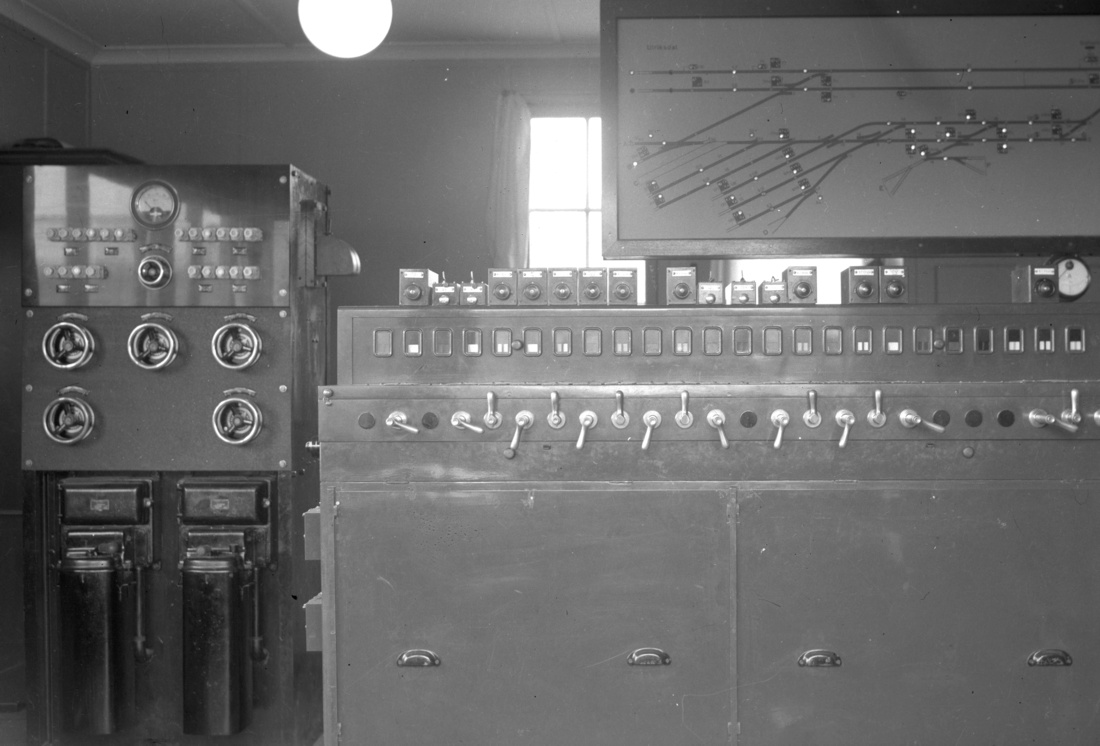
2019-04-07

Rinmangymnasiet Eskilstuna

Teknikprogrammet

Informations- och medieteknik

Lärare: Begona Ortiz Poyato



# Abstract

The goal of this project was to try and write a program that could simulate an interlocking plant at a somewhat bigger station in Sweden during the middle 1900s. The program was written in   
C# (C-Sharp) with Microsoft Visual Studio, and used MonoGame as its framework. GitHub was primarily used for version control, but will be used for issue tracking in the future. Due to improper planning and most of all a lack of time, the project did not reach all of its goals in time. I plan on further developing the program in the future to fully meet the criteria outlined in the report. The reader of this report will hopefully gain knowledge about the basic safety principles that need to be thought of when planning out an interlocking plant for a railway station.

Målet med projektet var att försöka skapa ett program som kan simulera ett signalställverk på en något större station I Sverige under mitten av 1900-talet. Programmet skrevs i C# (C-Sharp) med Microsoft Visual Studio som utvecklingsmiljö. MonoGame användes som ramverk till programmet. GitHub användes för versionshantering, men kommer också att användas för ärendehantering i framtiden. På grund av bristfällig planering och framför allt en kraftig tidsbrist nådde inte projektet upp till kraven i tid. Jag planerar att vidareutveckla programmet i framtiden för att nå upp till den kravspecifikation som anges i rapporten. Jag hoppas att läsaren får kunskap om de grundläggande säkerhetsprinciper som måste beaktas när man planerar ett signalställverk för en järnvägsstation.

# Innehållsförteckning

[Inledning 4](#_Toc5569064)

[Bakgrund 4](#_Toc5569065)

[Begreppsförklaringar och definitioner 4](#_Toc5569066)

[Järnvägsbegrepp 4](#_Toc5569067)

[Programmeringsbegrepp 7](#_Toc5569068)

[Syfte 7](#_Toc5569069)

[Mål 7](#_Toc5569070)

[Kravspecifikation 8](#_Toc5569071)

[Avgränsningar 13](#_Toc5569072)

[Tack till 13](#_Toc5569073)

[Metod 14](#_Toc5569074)

[Genomförande 15](#_Toc5569075)

[Projektering av stationen 15](#_Toc5569076)

[Namngivning 15](#_Toc5569077)

[Tågvägar 15](#_Toc5569078)

[Obevakad station och K-nycklar 16](#_Toc5569079)

[Vägskyddsanläggning 16](#_Toc5569080)

[Kodning 17](#_Toc5569081)

[Växlar och spårspärrar – Switch.cs och Derail.cs 17](#_Toc5569082)

[Signaler – Signals.cs 17](#_Toc5569083)

[Tågvägar – Route.cs 18](#_Toc5569084)

[Ställverkslogik – Interlocking.cs 19](#_Toc5569085)

[Användargränssnitt 19](#_Toc5569086)

[Resultat 20](#_Toc5569087)

[Uppfyllda krav 20](#_Toc5569088)

[Ej uppfyllda krav 20](#_Toc5569089)

[Diskussion 21](#_Toc5569090)

[Förbättringar 21](#_Toc5569091)

[Konsten att börja i tid 21](#_Toc5569092)

[Bättre logik för växelomläggning 21](#_Toc5569093)

[Mer detaljarbete med basklasser för huvud- respektive försignaler 21](#_Toc5569094)

[Fler listor som kontrolleras vid tågvägslåsning 22](#_Toc5569095)

[Bättre planering 22](#_Toc5569096)

[Saker som gjordes på ett bra sätt 22](#_Toc5569097)

[Förreglingstabellen 22](#_Toc5569098)

[Versionshantering med GitHub 22](#_Toc5569099)

[Framtida utveckling 23](#_Toc5569100)

[Litteraturförteckning 24](#_Toc5569101)

[Bilageförteckning 25](#_Toc5569102)

# Inledning

## Bakgrund

Dagens järnvägstrafik är ett under av många decenniers utveckling av säkerhetssystem. De modernaste systemen, med radiosignaler som konstant berättar för tåget var det befinner sig och vilken hastighet det ska hålla fanns inte från dag ett. De har stegvis utvecklats i takt med att trafiken och hastigheten på banorna har ökat. Det räcker inte längre att bara tillåta ett tåg mellan två stationer samtidigt. Man litar heller inte på att människor klarar av att göra det felfria arbete, som alltid krävs i järnvägssammanhang, i ett område med väldigt hög informationstäthet. Därför har man delat in sträckor i sektioner och infört datorer som övervakar hastighet och signalbesked. Det blev också snabbt opraktiskt att ha mycket personal som gick runt på bangårdar och lade om växlar, därför infördes ställverk där växlar kunde läggas om centralt. I samband med detta kunde man mekaniskt säkerställa att inga farliga situationer uppstod.

Projektet är förankrat i utbildningen genom programmeringen. C# är ett programmeringsspråk som jag lärde mig i kursen Programmering 1, och det verkade som en bra möjlighet att bygga vidare på mina kunskaper samtidigt som jag ägnade mig åt ett område där jag har ett stort intresse. Jag gjorde i årskurs 1 en film om hur järnvägens säkerhetssystem fungerar, den gången i samband med kursen Teknik 1. Jag fick genom det arbetet tillgång till kunskaper som jag har tagit till vara på i detta projekt, men också kompletterat med nya kunskaper.

Järnvägens ledord är säkerhet. Konstiga regler och udda tekniska lösningar kan oftast förklaras genom att de ger större säkerhet. Ett exempel från nutid, som dock inte kommer att vara aktuellt i simuleringen, är att man inte får ange var på linjen ett havererat fordon befinner sig; det är bara tillåtet att ange mellan vilka två stationer fordonet finns. Anledningen till detta är att man vill undvika missförstånd. Ett sådant missförstånd riskerar leda till att ett lok som skickas ut för att hjälpa det havererade fordonet håller för hög fart, eftersom det havererade fordonet egentligen befinner sig närmare än vad föraren på hjälploket tror.

## Begreppsförklaringar och definitioner

Rapporten behandlar både järnvägsteori och programmering, ämnen som är svåra att förstå och förklara om inte läsaren känner till vissa saker på förhand.

### Järnvägsbegrepp

**Låsning och förregling** En växel eller spårspärr som är låst går inte att läggas om. En växel, spårspärr, eller signal som är förreglad går inte att låsas upp på grund av en låst tågväg eller signal i kör. Förreglade signaler kan inte ställas i ”kör”.

**Spårspärr** Mekanisk anordning som genom klossar som fälls upp på spåret hindrar fordon från att passera. Spårspärrar betecknas med romersk numrering, SpI, SpIV. Se bild 1.

**Tågväg** Benämningen på den logiska väg som ett tåg kör över rälsen, vilka växlar det passerar igenom. En låst tågväg innebär att alla växlar i tågvägen är förreglade och inte kan läggas om eller låsas upp.

**Ställverk** En uppsättning komponenter (förut helt mekaniska, senare gradvis ersatta av elektriska) som kontrollerar och ibland kan styra växlar och signaler. Ställverkets logik är det som ser till att tåg inte spårar ur på grund av fellagda växlar och ser till så att tåg inte krockar. Illustrationer finns i Banlärans andra band, (Kungliga Järnvägsstyrelsen, 1916).

**Mekaniska ställverk**  
Ställverk som genom långa linjaler med tappar och hakar, kombinerat med vevar som styrde tappar som gick in i linjalens hakar, säkerställde att inga farliga situationer uppstod.

**Elektromekaniska ställverk**  
Ställverk som genom stora uppsättningar säkerhetsreläer hade exakt samma funktion, men var lättare att underhålla och bygga. Manövrering skedde dock fortfarande med en fysisk manöverpanel med tryckknappar eller med särskilda spakar, s.k. ställare.

**Elektriska ställverk**  
Ställverk som är byggda enbart med säkerhetsreläer eller är helt datoriserade. Elektriska ställverk styrs med datorer och kan nästan liknas vid ett spel. Elektriska ställverk kan i regel fjärrstyras från en annan plats.

**Skydd för tågvägar Flank-/Sidoskydd**  
En växel som ligger så att den leder in i en låst tågväg får inte förekomma, därför måste den låsas i avvisande läge för en viss tågväg. Ett exempel på detta är växel 6 som alltid måste inta avvisande läge när en tågväg från A låses.

**Skyddsväxel, Spårspärr**  
På vissa stationer kan tåg också mötas. Det innebär att två tåg som går in på två parallella spår i var sin riktning riskerar att krocka när tåget passerar växeln som för samman de två mötesspåren till ett. Därför finns ofta en skyddsväxel som förreglas i avvisande läge. Det spår som inte går ihop med enkelspåret efter skyddsväxeln är ett mycket spår som slutar i en stoppbock för att kunna ”fånga upp” tåg som inte hinner stanna i tid. Se bild 2. En skyddsväxel kan ersättas av en spårspärr.

**Skyddssträcka**  
Finns varken skyddsväxel eller spårspärr måste en skyddsträcka finnas. Skyddssträckan sträcker sig cirka 100 meter bakom tågvägens slutpunkt, och finns som en redundans i fall att ett fordon inte skulle lyckas stanna vid slutpunkten. Precis som för en vanlig tågväg får inga hinder finnas på skyddssträckan. Växlar och spårspärrar måste ligga rätt för att fordon inte ska spåra ur direkt om det inte lyckas stanna före tågvägens slutpunkt.

**Säkrad rörelse** Innebär att färden är planerad i minsta detalj, och föraren behöver i en teoretisk utopi inte veta vad signaler visar, eftersom det står på pappret hur färden ska gå till. Andra rörelser sker som siktrörelser, alltså där man bara får köra så fort så att man till exempel kan stanna vid ett hinder som oväntat uppenbarar sig.

**Tågfärd** Rörelseform som framförs som säkrad rörelse, och därmed ställs krav på att samtliga växlar i tågvägen måste vara förreglade.

**Växling** Rörelseform som används för att förflytta fordon inne på en station.

**Tågspår** Spår som är avsett att trafikeras i första hand av rörelseformen Tågfärd.

**Tågklarerare** Även tkl. Person som är högst säkerhetsansvarig för driften på en station, och för anmälan av vilka tåg som befinner sig på linjen som angränsar till stationen. Tkl hanterar tågmöten på den egna stationen.

**Linjen** Den sträcka som finns mellan två stationer. I simuleringen får högst ett tåg i taget befinna sig på linjen.

**Obevakad station** Station som inte är bemannad av en tkl och där tågmöten inte kan ske.

**Infartssignal** Även infsi. Den signal som visar om ett tåg får föras in på stationen, eller om det måste vänta utanför. Från en infsi finns i regel alltid en tågväg till ett spår inne på stationen.

**Utfartssignal** Även utfsi. Den signal som visar om ett tåg får föras ut på linjen, eller om det måste vänta inne på stationen. Finns inte på alla stationer, kan ersättas med tkl:s handsignal.

**Försignal** Även fsi. Visar vad nästa signal visar, så att ett fordon hinner stanna i tid.

**Semafor** Äldre signal som med vingar visar vilket spår tågvägen är lagd in till.

**Ljussignal** Ny signal som med ljussken visar vilket spår tågvägen är lagd in till. Kan även visa om tågvägen är förkortad, och vad nästa signal visar.

**K-lås och K-nyckel** Förkortning för Kontrollås/-nyckel. Ett lås som genom en särskild nyckel kan låsas i ett visst läge, varpå nyckeln inte kan tas ut ur låset. En nyckel som är inlåst i ställverk ger ställverket en kvittens på att objektet som nyckeln används att låsa, intar ett förutbestämt läge. K-nycklar finns av olika typer, och betecknas K1-K16.

**Växel- och spårspärrlägen** **Normalläge**  
Betecknas (+). Det läge som en växel eller spårspärr ska ha i normala fall, i regel det spåret som blir rakast. För spårspärrar varierar vad som är normalläge, pålagt eller avlagt läge, men oftast pålagt läge.

**Omlagt läge**  
Betecknas (-). Det läge som en växel ska ha i undantagsfall, då till exempel en tågväg måste läggas över växeln. I regel det mest krökta spåret.

**Vägskyddsanläggning** En plankorsning är den del som vägskyddsanläggningen skyddar. Vägskyddsanläggningen inkluderar sensorer och varningsinrättningar för både väg- och spårbunden trafik.

**Vägsignal**  
Även V-signal. Signal som med ljussken visar om korsningen får passeras. När de enligt ställverket åtgärderna för att varna vägtrafiken har genomförts slår den om från rött till vitt ljus, ”kör”. Signalen står jämte plankorsningen och ger samma sken åt båda hållen.

**Vägförsignal**  
Även V-fsi. Signal som står på långt avstånd från plankorsningen, och som med ljussken visar vad V-signalen visar. Detta för att fordon ska hinna stanna i tid om vägskyddsanläggningen inte aktiveras på rätt sätt.

**Spårledning** En svagspänning som läggs över rälerna för att kunna se att en viss sträcka är fri. Ett tågs axlar sluter kretsen och spänningen faller, något som ställverket känner av, och tolkar som att det finns ett fordon på den sträckan.

**Säo/TTJ** Säkerhetsordning, den föreskrift som beskriver vilka grundläggande säkerhetsregler som finns, och hur de ska tillämpas. Säkerhetsordning är ett föråldrat begrepp, den moderna motsvarigheten heter TTJ, Trafikverkets Trafikbestämmelser för Järnväg.

### Programmeringsbegrepp

**Klass** En slags mall för hur ett objekt ska se ut. Klassen är fylld med egenskaper och funktioner. En klass för ett tåg skulle bland annat kunna ha egenskapen ”hastighet” och funktionen ”kör”.

**Objekt** Ett objekt som är skapat efter den ”mall” som klassen anger. Objektet har ett eget namn, till exempel ”Tåg1”. Det går att skapa flera objekt av samma klass, så länge de har olika namn.

**Inkapsling** Ett sätt att styra vilka egenskaper och funktioner inuti objektet som är tillgängliga för ”omvärlden”, och vilka som är interna och bara kan användas som funktioner som jobbar inuti objektet.

**Klassarv** Är ungefär vad det låter som. En klass kan ärva funktioner och egenskaper från en annan klass, en basklass. Ett exempel är att både en bil och ett tåg är saker som rullar, och kan därmed ha många gemensamma egenskaper. Basklassen blir då gemensam för alla saker som rullar. I ”Tåg”- och ”Bil”-klasserna lägger man sedan till de egenskaper som bara är relevanta för respektive fordonstyp.

**Polymorfism** Betyder något förenklat att man döper funktioner i flera olika klassen till samma sak, men de uträttar en uppgift som är unik och relevant för den klassen.

**If-sats** En if-sats kontrollerar om något är uppfyllt. I ett modernt datorställverk kan till exempel en if-sats användas för att kolla om bommarna har gått ner vid en vägskyddsanläggning. If-satsen tillåter då programmet att exekvera kod som är kopplad till if-satsen.

**Konstruktor** En funktion i vissa programmeringsspråk som tillåter att man bland annat ger ett objekt värden på sina egenskaper i det moment som det skapas.

## Syfte

Projektet genomförs av flera anledningar. Först och främst så ger det mig en produkt som går att vidareutveckla till fullo i framtiden. Produkten skulle teoretiskt kunna användas i utbildningssyfte vid de järnvägar som tillämpar MRO (Museibanornas Riksorganisation) Säo. Den skulle också kunna användas i utbildningssyfte för att utbilda allmänheten om hur ett ställverk fungerade förr i tiden, om simuleringen kompletteras med kod och grafik för att visa hur insidan av ställverket fungerar.

## *Mål*

Målet med gymnasiearbetet är att skapa en datorsimulering som simulerar hur främst ställverkslogiken och i andra hand tågklarerartjänsten kunde ha fungerat på en något större mötesstation under mitten av 1900-talet.



Bild 1. Spårspärr i pålagt läge som hindrar fordon från att rulla ut från stickspåret.   
Stoppbocken skymtar i bakgrunden.

  
Bild 2. Skyddsväxel på Farsta Strand pendeltågsstation, Stockholm.  
Fordon som oavsiktligen kommer i rullning eller inte kan stanna före stoppsignalen åker in i den gula stoppbocken istället för ut på spåret där det kan stå pendeltåg fullastade med människor.

## Kravspecifikation

Stationen har tre tågspår med tillhörande plattform, och ett fjärde stickspår för uppställning av godsvagnar. Stationen har totalt 21 tågvägar, varav 12 är infartstågvägar och 8 är utfartstågvägar. Även en tågväg för obevakad station finns. Närmare krav för varje tågväg finns i bilaga 2, Förreglingstabell. Bilaga 1, Signalplanritning, och Teckenförklaringen för Växel- och Signalanläggningar (Larsson, ÖSlJH 32, 2013), bör studeras för att förstå nedanstående text och förreglingstabellen.



*Bild 3. Signalplanritning för Läggesta station (utanför Mariefred).*



Bild 4. Tågvägar a1/2/3, från infartssignal A till spår 1, 2, respektive 3.



Bild 5. Tågväg b1, från infartssignal B till spår 1.  
Sp II är pålagd för att fordon på spår 2 eller 3 ska hindras från att rulla ut i tågvägen.



Bild 6. Tågvägar b2/3, från infartssignal B till spår 2 och 3.  
Notera att b2/3medger samtidig infart med mötande tågväg a1.



Bild 7. Tågvägar c1/2, från infartssignal C till spår 2 och 3.



Bild 8. Tågvägar a2/3k, från infartssignal A till spår 2 och 3, men med Sp II pålagd.  
Tågvägarna medger samtidig infart med mötande tågväg b1.



Bild 9. Tågvägar c1/2k, från infartssignal C till spår 2 och 3, men med Sp II pålagd.  
Tågvägarna medger samtidig infart med mötande tågväg b1.



Bild 10. Tågvägar dI/II/III, från spår 1, 2 och 3 till utfartssignal D.



Bild 11. Tågväg eI, från spår 1 till utfartssignal E.   
Sp II är pålagd för att fordon på spår 2 eller 3 ska hindras från att rulla ut i tågvägen.



Bild 12. Tågvägar eII/III, från spår 2 och 3 till utfartssignal E.



Bild 13. Tågvägar fI/II, från spår 2 och 3 till utfartssignal F.



Bild 14. Tågväg a1o, obevakad station. Samma som a1, från infartssignal A till utfartssignal E,  
men som tillåter att motriktad tågväg b1 ställs samtidigt. Tågvägen innebär att  
K14 är isatt i ställverket. För övriga tågvägar ska den vara urtagen.

För tågvägar förbi A, B, D, och E som går på spår 1, gäller att växel 3 och därmed Sp I är förreglade i (+) med K3. För samtliga tågvägar förbi B och E ska V1 vara förreglad i (-). Vissa utfartstågvägar medger samtidig utfart från andra utfartssignaler. K-nycklar ska simuleras på stationen. För kontroll av att växel 3 och Sp I är låst i (+) ska K3 vara låst i ställverket. För att hindra att A och B visar ”kör” samtidigt ska K14 användas. För att låsa en tågväg ska K15 användas. I ställverket ska även en vägskyddsanläggning med V-signal med tillhörande V-fsi simuleras. Inne på stationen finns bara en V-fsi, för spår 1.

Programmet ska använda sig av klassarv, polymorfism, och inkapsling för att tillgodose kunskapskraven som ställs i Programmering 2. Programmet ska även kunna spara inställningarna som ställverket har (växlars lägen, låsta tågvägar, etc.) för att kunna läsa in dem vid senare tillfälle.

## *Avgränsningar*

Järnvägsrelaterad teknik är ett omfattande ämne, med krångliga regelverk, normer och termer. Även programmering är ett omfattande ämne, och kan lätt ta mer tid än planerat. Därför lägger jag i första hand tid på ställverkslogik, utan avancerad grafik eller omkringliggande funktioner. Följande avgränsningar har därför gjorts:

* Simulering av tåg som färdas genom en tågväg. Manövrering av ställverk sker på en ”tom” station utan fordon.
* Bättre grafik, med lampor som kan visa fordons lägen på stationer, och en manövrerbar ställverksapparat, som eventuellt visar hur ställverket fungerar inuti.
* Eventuella spårledningar som känner av var fordon befinner sig på stationen.
* Tkl-tjänst, alltså interaktion med kringliggande stationer och tillhörande pappersarbete.
* Simulera särfall och fel på ställverket, eller förseningar, extratåg, och växlingsrörelser.

Simuleringen inte är särskilt realistisk. Jag har inte använt mig av en faktakontrollant eller följt alla tekniska normer och ritningar från stationen som simuleras. Det är i nuläget odefinierat om ställverkslogiken är uppbyggd av linjaler eller reläer, även om den bör vara elektromekanisk och använda säkerhetsreläer och fysiska ställare. I vilket fall som helst är ställverkslogiken förenklad, vilket innebär att många kontrollåtgärder inte simuleras. Till exempel finns det ingen kontroll av att lampa verkligen lyser, något som finns i vissa riktiga ställverk för att tidigt upptäcka signalfel. I en perfekt simulator skulle varje relä simuleras exakt så som det fungerar, alltså i princip rad efter rad av if-satser, eller att man med hjälp av kollisionslogik ritar upp linjaler som ”mekaniskt” förhindrar omläggning av växlar och dylikt.

## *Tack till*

Lärare, mina föräldrar, och vänner som har hjälpt och stöttat mitt arbete med gymnasiearbetet. Ingen nämnd, ingen glömd.

# Metod

Eftersom gymnasiearbetet görs parallellt med kursen Programmering 2, har det inneburit att projektet har anpassats till kursens lärobok. Kursen ges i programmeringsspråket C# (”C-Sharp”). Boken har sin utgångspunkt i att eleven ska använda ramverket Monogame eller Xna för att på bästa sätt lära ut kursens centrala innehåll. Ramverken är i princip samma sak, men Monogame baseras på Xna och är modernare; därför har jag valt att använda just Monogame. Monogame har även stöd för att kunna utveckla för flera plattformar, jämfört med Xna som bara tillåter utveckling för Microsofts produkter, som till exempel Windows och Xbox. Detta är en fördel, eftersom planen är att vidareutveckla programmet efter att gymnasiearbetet är slutfört. Som utvecklingsmiljö används Microsoft Visual Studio, på grund av att det har många verktyg som underlättar utveckling, särskilt för personer med bristande kunskap. Det lämpar sig bra att använda i kombination med MonoGame.

Projektet har använt GitHubs versions- och projekthantering. Det innebär att en kommentar skrivs när en ändring i koden görs. Dessa ändringar kan följas upp, men kan även visualiseras med ett insticksprogram för GitHub, Gource. Gource skapar animeringar som visualiserar mappstruktur och kodens utveckling över tid.

Vad gäller läromedel, var den enda faktakällan inom programmering i början läroboken *Programmering 2 C#* (Trangius, 2013)*.* Jag insåg snabbt att den skulle vara otillräcklig, varefter internet har varit den huvudsakliga informationskällan. Webbplatser som har utgjort en betydande del av faktainsamlingen är till exempel: Stackoverflow.com, Monogame.net, och Microsofts dokumentation för .NET och C#. På grund av att källförteckningen skulle bli översvämmad av hundratals länkar och referenser till foruminlägg, anges inte de källorna närmare än så här.

Programmeringen är bara den sista halvan av projektet. Den första halvan bestod av att planera station och ställverk för att sedan skriva om det till kod. Kunskap om ställverk är svårtillgänglig, men går att hitta. På IDG.se finns utförliga artiklar om hur moderna ställverk fungerar. För fördjupning i ämnet har jag använt *Banlära, 2:a bandet* (Kungliga Järnvägsstyrelsen, 1916) bidragit, samt vissa interna föreskrifter och handböcker vid ÖSlJ. ÖSlJF 310 (Sundberg, 2011), (Larsson, ÖSlJF 319, 2011), (Larsson, ÖSlJH 31, 2013), (Larsson, ÖSlJH 32, 2013). ÖSlJF 310 och ÖSlJF 319 innehåller information som inte är offentlig och inte heller får spridas till obehörig personal. På förfrågan kan tillstånd för begränsad spridning ordnas.

Mycket av den använda litteraturen är flera decennier gammal, men är ändå relevant för projektet, dels eftersom simuleringen är tänkt att utspela sig i den tidsepok som litteraturen ofta avser, dels för att järnvägen är ganska oförändrad sedan dess. Det finns exempelvis ingen annan litteratur än *Banlära* (Kungliga Järnvägsstyrelsen, 1916) som beskriver signalteknik på ett så ingående sätt.

För att undvika att tänka igenom all logik i samma ögonblick som den skrivs gjorde jag tabeller och diagram som visar ställverkslogiken. På så sätt slipper man haka upp sig på den faktiska logiken när koden skrivs, utan kan istället fokusera på att översätta diagrammen till kod. Vissa av dessa följer vedertagna normer och standarder så gott det går, andra gör det inte alls. Standarderna i fråga finns lättast åtkomliga i ÖSlJs interna handböcker, och rör vissa specialtecken och konventioner som används vid ställverksplanering.

Notera att ÖSlJ är ett museum som ska återspegla svenska banor med 600 millimeters spårvidd. Alltså är de föreskrifterna inspirerade av privatägda banor, som i sin tur har fått inspiration från dåvarande SJ. Det innebär att till exempel konventioner om hur en station ritas upp kan avvika från vad SJ egentligen tillämpade, vilket gör simuleringen ännu mindre realistisk.

Jag har pratat med sakkunniga människor som jobbar med eller har god kunskap om ställverk tidigare. Jag har även gjort ett studiebesök på Trafikverket i samband med ett annat projekt. Detta var dock 2017, två år sedan. Jag kommer därför inte ihåg alla detaljer utan bara övergripande information om hur ställverk fungerar. Detta är varför det bara nämns förbigående.

# Genomförande

Projektets genomförande har varit någorlunda komplicerat, och kan huvudsakligen delas upp i två faser. Fas 1 bestod av att rita upp en ritning av stationen, nummerge växlar, spårspärrar och signaler, samt att bestämma vilka K-nycklar som skulle användas. Fas 2 bestod av att omsätta teori till kod.

## Projektering av stationen

För att förstå innehållet i denna rubrik rekommenderas det starkt att ha Bilaga 1, Signalplanritning, nära till hands. I denna fas har jag försökt följa föreskrifter och regler så långt som det har varit praktiskt möjligt.

### Namngivning

Varje tåg har ett tågnummer, som anger vilken tidtabell som det fysiska tåget ska följa. Udda tågnummer går söderut, och jämna tågnummer går norrut. Finns det flera spår, går tågen söderut på nedspåret, och norrgående tåg på uppspåret. Principen för att ge en signal, växel, eller spårspärr sitt nummer eller bokstav, är baserat på tågnummer. Udda tågnummer, till exempel tåg 5, möter udda växlar efter att ha passerat en infartssignal med ”udda” bokstav, alltså A i detta fall. Därefter möts det av växel 1, 3, 5, och så vidare. Jämna tågnummer möter istället växel 2, 4, 6 efter att ha passerat infartssignal B, (Larsson, ÖSlJF 319, 2011). Spårspärrar verkar inte följa dessa regler, och är numrerade med romerska siffror.

### Tågvägar

I fas 1 gjordes en så kallad förreglingstabell. Förreglingstabellen har till uppgift att på ett enkelt sätt visa vilka beroenden som behöver uppfyllas för en viss tågväg. En tågväg skrivs med liten bokstav, och har samma bokstav som infartssignalen den börjar vid. För att skilja på olika tågvägar från samma infartssignal anges ett nummer som exponent till bokstaven. a1 är alltså tågväg 1 från signalen A. Det behöver dock inte betyda att tågvägen går in på spår 1. I simuleringens fall finns tågvägarna c1 och c2, men de går in på spår 2 respektive spår 3. Det är för att spår 1 inte är åtkomligt från infartssignal C, och därför blir tågväg 1 från C det ”lägsta” spår som kan nås, alltså spår 2. Viss skillnad finns när det gäller en utfartssignal. Det följer samma princip att udda tågnummer möter ”udda” bokstäver, men numret för tågvägen anges med romerska siffror.

En infartstågväg börjar vid en infartssignal, och slutar logiskt nog vid tågvägens slutpunkt, som är den tekniska benämningen. Infartstågvägens slutpunkt behöver inte vara nästa signal, utan kan vara utmärkt med tavlor. En utfartstågväg börjar från ett av tågspåren inne på stationen, och slutar vid utfartssignalen, eller strax bortom den; dock aldrig längre än till den motriktade infartssignalen. Bland stationens 12 infartstågvägar är fyra av dem korta tågvägar. Det är tågvägarna från infartssignal A och C in på spår 2 och 3, när Sp II är i pålagt läge. Skulle inte spårspärren vara i pålagt läge måste växel 4 ligga rätt för respektive tågväg för att utgöra skyddssträcka. Samma krav på skyddssträcka finns också för tågvägar från infartssignal B till spår 2 och 3. Tågvägar som går på spår 1 anses ha tillräckligt lång skyddssträcka från tågvägens slutpunkt fram till nästa växel.

Tågvägar som går genom växel 1 och 6 kräver alltid att båda växlarna intar samma läge. Ligger ena växeln i omlagt läge, måste den andra också göra det för att inte fordon ska åka in i en felaktigt lagd växel. Ligger en växel i normalläge måste även den andra växeln ligga i normalläge för att skydda tågvägen från sidan. Detta görs för att ett fordon som kommer i rullning på spår 2 eller 3 inte ska kunna kollidera med ett fordon som finns på spår 1, snarare än tvärt om.

### Obevakad station och K-nycklar

Det är inte tillåtet att använda K-nycklar hur man vill. Exakta krav finns i BVF 544.94005, (Banverket, 1999), men i huvudsak gäller att vissa K-nycklar ska användas till vissa saker, får inte finnas i samma K-lås etc.

Det finns ofta ett behov av att ha en station obevakad då det inte förekommer några tågmöten där. En obevakad station betraktas då precis som vilken annan del av linjen som helst. Ställverket måste alltså kunna tillåta att tåg från båda hållen passerar utan att någon justerar ställverket. För att lösa detta så finns tågvägen a1o (o för obevakad). K-nyckel K14 får bara användas för att ställa in ställverket för obevakad körning. När stationen lämnas obevakad ska K14 sättas in i ställverket och vridas om, så att nyckeln låses fast. Då ställs ställverket in för obevakad körning, varpå tågväg a1o kan ställas, samt tågväg b1. Tanken med K14-systemet är att om någon obehörig skulle vrida ur nyckeln ur ställverket, kan man ändå inte ställa infartssignaler i ”kör” utan att använda K15. Alltså blir en kollision omöjlig. Skulle nyckeln istället sitta i när stationen är bevakad, riskerar den att stjälas när två mötande tåg är på ingående. Det är därmed möjligt att få ”kör” i respektive infartssignal samtidigt, så att tågen krockar. En K14 som inte används måste förvaras enligt lokala bestämmelser, ofta på en plats där bara tågklareraren kan komma åt den.

Som kravspecifikationen anger ska växel 3 vara låst i (+) för vissa tågvägar. Tågvägarna som berörs är de med risk för att fordon som finns på stickspåret oavsiktligt kommer i rullning, och därmed kan komma ut i en tågväg. Även alla tågvägar över spår 1 omöjliggörs, eftersom de tågvägarna förutsätter att växel 3 ligger i (+). Eftersom växeln inte kan läggas om centralt är den utrustad med ett dubbelt kontrollås som beskrivs i ÖSlJH 31 (Larsson, 2013), sida 3, fig. 22.

När växeln behöver läggas om tas K3 ur ställverket, som därmed inte längre får en kvittens på att växeln är låst i (+), och tågvägslåsning omöjliggörs för alla tågvägar som kräver växel 3 i (+). När K3 sätts i K-låset, och vrids om kan K9 vridas om och tas ur det andra låset. K9 i sin tur låser upp ett K-lås som låser spårspärren i pålagt läge, varpå spårspärren kan sänkas, och fordon på stickspåret kan växlas ut. När växlingen avslutas sker proceduren baklänges.

### Vägskyddsanläggning

I stationen ingår en vägskyddsanläggning som till sin funktion är relativt enkel. Den aktiveras antingen av tågklareraren när stationen är bevakad, eller automatiskt med hjälp av spårledningar när stationen är obevakad. Den aktiveras inte alltid automatiskt eftersom den måste aktiveras så långt i förväg att bommarna är nere i god tid även för ett tåg som inte stannar på stationen. Det innebär samtidigt att tiden mellan att varningssignaleringen startar och att tåget passerar plankorsningen blir väldigt lång då ett tåg gör uppehåll vid stationen när den är obevakad. Detta gäller dock inte när tåg kommer från infartssignal B, eftersom tåget stannar efter att plankorsningen har passerats. Oavsett riktning och om stationen är bevakad eller inte, så är målsättningen att minimera tiden som vägtrafikanter måste vänta vid vägskyddsanläggningen.

Till vägskyddsanläggningen hör även två V-försignaler. En av dem finns inne på stationen, medan den andra finns ute på linjen, före infartssignal B. V-försignaler sitter på sådant avstånd från vägskyddsanläggningen att ett fordon ska kunna stanna före korsningen om vägskyddet inte aktiveras på rätt sätt. Eftersom tåg från infartssignal C i regel alltid gör uppehåll på stationen, finns det inget behov av att försignalera vägskyddsanläggningen. Det är tågklarerarens uppgift att se till att vägskyddet är aktiverat innan denne ger avgångssignal till tåget.

## Kodning

Allt ovan har beaktats i projekteringen av stationen och ställverket. Därefter påbörjades fas 2 som innebar att översätta allt ovan till faktisk kod. Innan någon kod skrevs, sattes versionshantering med GitHub upp, och MonoGame installerades. GitHub möjliggjorde att kommentarer skrevs varje gång som koden ändrades på något sätt. Ett plugin till GitHub, Gource, möjliggör att skapa korta filmklipp där projektets utveckling lätt kan visualiseras.

Koden är uppbyggd kring ett ställverksobjekt, interlocking, som anropar publika funktioner hos objekt som växlar och signaler. På så vis kan ”ställverket” manövrera de saker som de behöver. I teorin kan flera olika ställverksobjekt skapas för att simulera grann-ställverk på andra stationer. Lämpliga namn för sådana skulle vara stationens namn, alternativt stationens namn kombinerat med ”ställverk” eller ”interlocking”.

### Växlar och spårspärrar – Switch.cs och Derail.cs

Den första ”riktiga” klassen som skapades innehöll koden för växlar, översatt till *switch* på engelska. Det enda som inte är åtkomligt för ställverket är om växeln rör på sig och om ett fordon befinner sig på växeln; \_isMoving, \_isOccupied. Understrecken är ett sätt att tydliggöra att variabeln är privat. De variablerna kontrolleras i funktionerna för omläggning och låsning av växeln. Det finns fyra publika funktioner som kan anropas, det är funktionerna för omläggning och låsning: StraightSwitch(), CurveSwitch(), LockSwitch(), UnlockSwitch(). Funktionerna är gjorda så att de returnerar true om åtgärden går att utföra, annars false. Detta för att ställverket ska få en kvittens på att en åtgärd utfördes korrekt. Ställverket kan också se i vilket läge en växel ligger, samt om den är låst eller ej: IsStraightTrack, IsCurvedTrack, IsLocked. En växel har också ett publikt namn som kan användas för att förenkla felsökning, displayName.

När en växel läggs om kontrollerar växeln först att den inte är låst eller att något befinner sig på den. Skulle någon av de variablerna vara true returneras false genast till ställverket. Den process som ställverket eventuellt gör avbryts, eftersom ett steg i processen misslyckades. Är båda variablerna false, ändras \_isMoving till true och IsStraightTrack till false. Sedan pausas programmet i en sekund, för att simulera omläggningstiden. Därefter ändras \_isMoving till false och IsCurvedTrack till true. Ett meddelande skickas till loggen om att växeln nu ligger i (-), och till sist returneras true om inget oväntat har inträffat. Ställverket kan då fortsätta till nästa steg i sin process.

Spårspärrar fungerar på nästan exakt samma sätt. Skillnaden är att spårspärrar inte kan känna av om ett fordon skulle befinna sig på den. Ett fordon kan inte befinna sig på en pålagd spårspärr. Om ett fordon står vid en avlagd spårspärr så har den som fäller upp spårspärren en skyldighet att kontrollera att den inte fälls upp mellan ett fordons hjul. För en spårspärr är vissa variabel- och metodnamn är utbytta; Curve, Straight, och Switch, ersätts med Raise, Lower, och Derail.

### Signaler – Signals.cs

Den andra klassen som skapades innehöll koden för signaler. Signaler lämpade sig bra för att använda klassarv och polymorfism, två kursmål i programmering 2. Basklassen är abstrakt, förenklat innebär det att den inte kan bli ett objekt ensam. Den kräver att en annan klass ärver basklassens innehåll för att kunna användas.

Alla signaler har gemensamt att de ska kunna visa ett signalbesked; signalState. De har också gemensamt att de måste kunna förreglas i ”stopp”, alltså genom en låst tågväg förhindras de att ställas i kör: isProtected. Klassen har två publika funktioner för att förregla en signal: SetProtected() och SetUnprotected(). Alla signaler måste kunna ställas i ”stopp” eller ”vänta stopp”, men alla signaler visar inte samma typ av ”kör”. Därför har basklassen bara en funktion för att ställa en signal till stopp: SetStop(). Funktioner för att ställa signal till kör finns alltså bara i den klass där det behövs.

Infartssignaler har tre funktioner som alla påverkar signalState i basklassen. Vilken metod som anropas beror på vilket signalbesked som ska lämnas. signalState sätts till 1, 2, eller 3, och motsvarar samma antal tända gröna lampor i signalen. En grön lampa innebär att tågvägen är lagd in på huvudspåret, och att färden kan fortsätta utan begränsningar. Två gröna lampor betyder att tågvägen är lagd in på ett sidospår. Största tillåtna hastighet är 40 km/h från och med signalen. Tre gröna lampor har samma signalbetydelse som två gröna, men visar också att tågvägen är avkortad. Tågvägens slutpunkt kan då ligga så tidigt som 450 meter efter signalen (Trafikverket, 2019). Tre gröna lampor används när tågväg läggs in på spår 2 eller 3, med uppfällda spårspärrar.

Det enda villkoret för att få ställa en signal i kör är att signalen inte är förreglad i ”stopp”. Är en tågväg låst, ska ställverket förregla de infartssignaler som inte får ställas i ”kör”. Således säkerställs det att fel tåg inte kommer in på en låst tågväg. Tågklareraren kan inte manövrera signaler direkt. Ställverket ställer automatiskt rätt signaler i ”kör” när tågklareraren låser en tågväg. Att tågklarerarens order om att ställa signalen till ”kör” måste gå via ställverket, gör att säkerheten höjs ytterligare ett snäpp.

En infartssignal har även möjligheten att ha en inbyggd försignal. I infartssignalens konstruktor går det att ange vilken signal som den infartssignalen ska försignalera, referenceSignal, i detta fall utfartssignalen på andra sidan stationen. Tanken är att signalen hela tiden ska titta på vad nästa signal visar, och visa respektive försignalbesked. Om signalState hos den försignalerade signalen är 1, visas ett vitt blinkande sken, annars två gröna blinkande sken. Om signalState är 0, visas ett grönt blinkande sken. En vanlig fristående försignal har exakt samma funktion. En signal försignaleras alltid med antingen en fristående försignal, eller en signal med inbyggd försignalering. Denna funktion är dock inte implementerad än.

Utfartssignaler kan i simuleringen bara visa ”stopp”, eller ”kör” med ett grönt sken. Därför har den bara en metod: setClear(). Även här är det enda villkoret att den inte är förreglad i ”stopp”. Det beror på att man inte ska kunna skicka ut tåg om den motriktade infartssignalen visar ”kör”. Detta skulle vara fullt möjligt om både ett tåg inne på stationen och ett ankommande tåg ska åka på spår 1. Då står växlarna rätt för båda tåg. Om en av de motriktade signalerna förreglas i ”stopp”, blir det omöjligt att ställa motriktade tågvägar.

V-signalens funktion är likt försignaler inte implementerad än, men funktionen beskrivs här. Kravet för att en V-signal ska kunna visa ”kör” är bara att vägskyddsanläggningen har meddelat att den är i skyddat läge. V-signalen växlar då från rött till vitt sken åt båda hållen. Det spelar alltså ingen roll från vilket håll tåget kommer.

V-signalen försignaleras för att fordon ska kunna stanna före en röd V-signal. Funktionen för en V-försignal är densamma som för en vanlig försignal, förutom att en V-försignal bara kan försignalera ”stopp” eller ”kör”, och inte kommande hastighetsnivåer som en vanlig försignal kan. Båda V-försignalerna visar alltid samma signalbesked, eftersom V-signalen ger sin signal åt båda håll.

### Tågvägar – Route.cs

Arbetet fortsatte med att skapa tågvägar, översatt till *Routes* på engelska. När en tågväg skapas finns det ett antal signaler och växlar som är relaterade till tågvägen. Det är både växlar som direkt ingår i tågvägen, men också växlar och signaler som måste förreglas i avvisande läge eller ”stopp” för att tågvägen ska kunna låsas. I konstruktorn för tågvägsklassen anges olika sätt att skapa tågvägsobjekt eftersom olika tågvägar inte har samma antal bangårdsobjekt som berörs av tågvägen. Dessa växlar och signaler förs in i sex olika listor beroende på vilket läge de ska inta för tågvägen.

När en tågväg ska låsas, går metoden igenom alla listor och kontrollerar att alla bangårdsobjekt faktiskt har de lägen de ska ha. Till exempel ligger alla växlar som ska ligga i (+) i sin egen lista. När metoden LockRoute() anropas kollar den bland annat växlar som ligger i listan straightSwitches, och ser till att de ligger i (+). Ligger alla växlar i (+) fortsätter metoden till nästa lista och gör samma sak. När alla sex listorna har gåtts igenom utan problem, returnerar metoden true till ställverket, som återigen får en kvittens på att låsningen utfördes korrekt. Skulle något gå fel, returnerar metoden false. Ställverket kan sedan tända en lampa eller liknande på en kontrollpanel, så att även tågklareraren får en indikation på att låsningen utfördes. Även variabeln isLocked ändras till true när alla bangårdsobjekt har kontrollerats. isLocked finns för att ställverket vid behov ska kunna kolla om någon tågväg är låst, utan att behöva ”låsa” den igen. Koden för tågvägslåsning finns, men går i nuläget inte att anropa på något sätt.

Beroende på ställverkstyp ställs infartssignalen i ”kör” direkt när tågklareraren låser en tågväg, eller så måste detta göras som en separat åtgärd när tågvägen är låst. I simuleringen kommer detta att ska automatiskt eftersom simuleringen försöker simulera ett elektromekaniskt ställverk. I ett mekaniskt ställverk krävs att tågklareraren gör detta som en separat åtgärd.

### Ställverkslogik – Interlocking.cs

I klassen Interlocking.cs skapas alla bangårdsobjekt, d.v.s. alla signaler, växlar, och spårspärrar. När alla objekt finns skapas tågvägarna, och relevanta bangårdsobjekt kopplas till dem. För tågväg a2 måste 12 olika objekt kopplas till tågvägen. De är alla bangårdsobjekt som tågvägen beror av. Konstruktorn för varje tågväg anger i vilken ordning bangårdsobjekten ska anges i. I regel gäller, när det är applicerbart, ordningen: tågvägsvariant, försignal, infartssignal, växlar som ingår i tågvägen, växlar som måste låsas i avvisande läge, förreglade signaler, spårspärrar som måste kontrolleras, V-försignal, V-signal, (K-lås). Koden för K-lås finns inte än. Tågvägsvariant finns för vissa tågvägar som går igenom exakt samma växlar. Det går då inte att skilja på tågvägarna. Beroende på vilken tågvägsvariant som anges, läggs växlarna i den lista och det läge som tågvägen kräver.

### Användargränssnitt

Simuleringen ska eftersträva att efterlikna en manöverpanel som kan finnas på ett riktigt ställverk. För enkelhetens skull valdes att göra det med tryckknappar, eftersom det är lättast att skapa kod för. En bild på en lämplig knapp hämtades från Internet och beskars så att den skulle passa för ändamålet. Koden för knappens funktion är huvudsakligen tagen från Internet (Monogame Community), (<https://stackoverflow.com>), och sedan anpassad för att fungera korrekt i programmet.

För att se ut som en riktig manöverpanel krävs också att lampor och färgade fält kan visas. Detta är tänkt att lösas med Indicator.cs-klassen. Funktionen är tänkt att ge två texturer till det skapade Indicator-objektet, en (+) textur, och en (-) textur. Objektet tittar mot en del av ställverkskoden, och byter textur beroende på om ett visst krav är uppfyllt eller inte. Till exempel ska tågvägslås visas med rött eller vitt fönster, och andra lampor ska visa vilka signalbilder som finns i infartssignalerna. Koden för indikatorer finns, men är varken implementerad eller testad.

# Resultat

Resultatet uppnår inte hela kravspecifikationen. I nuvarande form fungerar inte programmet som en simulering som visar hur ett ställverk till fullo kunde ha fungerat. Programmet kan således ännu inte användas till något av de föreslagna utbildningssyftena.

Det finns inga hinder för att programmet vidareutvecklas och i framtiden uppnår samtliga krav i kravspecifikationen. Möjligheten finns också att ta bort vissa av de avgränsningar som är satta och lägga till fler funktioner i programmet så att simuleringen ska bli så verklighetstrogen som möjligt. Projektet har bidragit till att mina kunskaper inom både järnvägsteknik och programmering har fördjupats.

## Uppfyllda krav

Programmet kan manövrera växlar och spårspärrar, samt känna av att dessa inte är blockerade av fordon. Det kan också känna av vilket läge en växel eller spårspärr redan ligger i, och förhindra att den läggs om åt ”samma håll” två gånger i rad. Programmet har samtliga tågvägar som anges i kravspecifikationen. Även om inte a1o fungerar exakt som angivet, finns en motsvarande lösning.

Programmet använder sig av inkapsling, klassarv och polymorfism i signal- och växelklasserna. Växlar använder sig av inkapsling för att ”gömma” information som ställverkslogiken inte behöver känna till, till exempel om en växel är i rörelse. Signaler använder sig av klassarv och polymorfism för att ärva basklassens egenskap att alla signaler ska kunna visa ”stopp” eller ”vänta stopp”.

## Ej uppfyllda krav

Teoretiskt går det att låsa tågvägar, men eftersom funktionen inte är prövad kan det inte garanteras att den fungerar korrekt. Tågvägar som medger samtidig infart eller utfart med andra tågvägar fungerar också i teorin, men eftersom funktionen inte är prövad går det inte att anta att den uppfyller kravspecifikationen.

Programmet klarar heller inte att hantera K-nycklar. Koden för vägskyddsanläggningen finns inte, således kan inte programmet simulera den. Eftersom funktion för varken K-nycklar eller vägskyddsanläggning finns, går det (teoretiskt) att låsa tågvägar utan att kontrollera att K-lås och vägskydd är i sina rätta lägen. Detta är enligt kravspecifikationen inte tillåtet.

Programmet kan inte spara ställverkets inställningar, till exempel växlars lägen. Det kan inte heller läsa och ställa in de inställningar som skulle kunna läsas från en sparad fil.

# Diskussion

I sin helhet tycker jag att gymnasiearbetet inte blev särskilt lyckat. Det framgår tydligt under rubriken Resultat att många delar av den tänkta slutprodukten fattas. Det beror på bland annat misstag i både genomförande och planering. Framför allt beror det dock på kraftig tidsbrist; hade projektet kunnat starta på utsatt startdatum, och inte 4 månader efteråt, skulle förmodligen flera, om inte samtliga, delar av kravspecifikationen vara uppfyllda.

## Förbättringar

### Konsten att börja i tid

Det enskilt största misstaget med detta projekt var att inte börja i tid. Det berodde på att jag inte insåg möjligheten till detta projekt förrän flera månader efter att jag hade påbörjat mitt första gymnasiearbete. Teknikprogrammet på Rinmangymnasiet erbjuder i första hand elever att kombinera gymnasiearbetet med kursen Entreprenörskap 1. Det innebär att elever starkt uppmuntras till att starta ett eget Ungt Företagande-företag. Eleverna får knapphändig information om att det teoretiskt går att välja bort Entreprenörskap och göra gymnasiearbetet i ett, för eleven, mer intressant ämne. Så även i detta fall, där jag plågades av de fruktansvärt tråkiga uppgifterna i Entreprenörskap. Skulle det inte ha varit möjligt att göra ett ställverk som gymnasiearbete, hade jag med högsta sannolikhet istället levererat en *mycket* utförlig rapport om varför Entreprenörskap och Gymnasiearbete är en högst opassande kombination.

Det är svårt att se hur detta kunde ha undvikits, men det visar tydligt att programmering kan ta lång tid, även om programmet inte verkar särskilt avancerat. Det visar också att det är värt att byta till ett ämne som är intressant, istället för att genomlida ett arbete inom ett ämne som man inte alls tycker är intressant.

### Bättre logik för växelomläggning

Som programmet fungerar nu finns det en fördröjning på en sekund för att simulera omläggningstid. Detta har åstadkommits genom att processen pausas helt i en sekund. En bieffekt av detta blir att hela programmet pausas, och det går inte att interagera med några andra element i programmet under denna sekund. I verkligheten är omläggningstiden sannolikt längre för centralt elmanövrerade växlar, vilket innebär att pausen skulle behöva bli längre för att simuleringen ska vara verklighetstrogen. Ytterligare ett fel med knapparna är att de inte ”fastnar” i intryckt läge när de trycks in. De trycks istället in under en bråkdels sekund, *efter* att programmet har pausats i en sekund.

Lösningarna på båda problem är relativt enkel. Istället för att pausa hela programmets process, borde det gå att skapa en process enbart för knapptryckningen. Knyts knapptryckningen till en egen process, borde det gå att pausa enbart denna istället för hela programmet. En bättre, däremot svårare, metod skulle vara att tillverka en fördröjningsfunktion som kan anropas för ett visst objekt istället för att pausa processen. För att knapparna ska ”fastna” när man trycker på dem borde det inte vara svårare än att flytta intryckningen av knappen till precis innan processen pausas. Då byter knappen textur till en intryckt knapp ögonblickligen innan processen pausas. Så fort processen återupptas så återgår knappen till sitt normala utsläppta läge, eftersom den inte längre är klickad på.

### Mer detaljarbete med basklasser för huvud- respektive försignaler

Det är inte optimalt att en försignal, som inte kan ”beordra” ett fordon till ”stopp” på samma sätt som en huvudsignal, fungerar på samma sätt som om den hade varit en huvudsignal. För att tydligare skilja på en huvudsignal och försignal (även V-signal, V-försignal) bör en särskild klass för försignaler skapas. Basklassen har bara en egenskap, att titta på vad signalen framför ger för signalbesked. Om försignalen ska kunna signalera kommande hastighetsnivåer eller bara ”stopp” eller ”kör” är upp till den riktiga klassen att avgöra.

### Fler listor som kontrolleras vid tågvägslåsning

En mindre brist, men som trots allt gör att vissa krav inte uppfylls, är avsaknaden av kontroll av vägskyddsanläggning och K-lås. Eftersom det inte alls kontrolleras om till exempel vägskyddet är aktiverat, går det (teoretiskt) att låsa tågväg även fast det inte är tillåtet enligt kravspecifikationen. Lösningen är väldigt enkel, lägg till ytterligare två listor som kontrolleras vid tågvägslåsning. Eftersom det inte finns någonting som kan uppfylla kraven på upplåsta K-lås och vägskyddsanläggningar, blir det omöjligt att låsa en tågväg som kräver något av de åtgärderna.

### Bättre planering

Projektet hade varit lättare att genomföra om jag verkligen var helt säker på vad jag ville att simuleringen skulle åstadkomma. Jag insåg snabbt värdet av att ha en kravspecifikation, men med facit i hand borde den ha varit mycket mer utförlig. Att definiera om ställverket är mekaniskt, elektromekaniskt, eller helelektriskt skulle ha underlättat planeringen samt skrivande av rapport. Det hade dessutom varit bra att bestämma under vilken tidsperiod som simuleringen utspelar sig. Då skulle jag kunna ha gjort bättre efterforskningar över vilka standarder som gällde på den tiden, och eventuellt konsulterat en specialist inom ämnet. Det hade möjligvis också inneburit att någon annan kunde granska rapporten, framför allt begreppsförklaringarna, för att säkerställa att de används på ett korrekt sätt.

Det är också enligt min åsikt högst olämpligt att blanda föreskrifter, normer, och konventioner från olika tidsperioder. Det är särskilt olämpligt att blanda sådant material från statens järnvägar med privatägda banor med annan spårvidd, eftersom regelverken många gånger kunde skilja sig på grund av annan typ av trafik eller kraftigt sänkta hastigheter.

Att hela tiden pendla mellan olika typer av ställverk har lett till att projekteringen och kodningen ibland har använt lösningar som är realistiska och relevanta i en viss typ av ställverk, men helt orealistiska i ett annat. I Bilaga 3, Kontroll av Växel, går det att se att jag ursprungligen hade tänkt att växlar i tågvägen automatiskt skulle läggas om när en tågväg låstes. Denna funktion finns i moderna helelektriska ställverk, men i mekaniska och elektromekaniska ställverk var omläggning av växlar en manuell åtgärd som behövde göras innan tågvägen låstes och växlarna förreglades.

Till mitt försvar hade sådana efterforskningar bidragit till att programmet blivit mer försenat och ännu mindre färdigt. Det känns bättre att ha något att visa upp, än att ha en exakt plan på hur det skulle ha sett ut.

## Saker som gjordes på ett bra sätt

Trots viss negativ ton är jag väldigt glad att jag gjorde en del saker tidigt. Att göra en förreglingstabell och att skapa versionshantering med GitHub är exempel på sådana saker.

### Förreglingstabellen

Bilaga 2, Förreglingstabell, bistod mig när jag arbetade med att ange vilka bangårdsobjekt som skulle finnas i vilka lägen under kodningen av tågvägarna. Sannolikt skulle avsaknaden av en sådan tabell innebära att jag skulle ha hoppat mellan programmeringstänk och säkerhetstänk hela tiden.

### Versionshantering med GitHub

Fördelarna med GitHub är väldigt många. Dels går det att gå tillbaks i en fil och titta när en viss ändring har blivit gjord. Det går också att ”spola tillbaka” tiden, och börja om vid ett tidigare skede om man upptäcker ett så pass allvarligt fel med koden att man är tvungen att börja om. Lyckligtvis behövde jag inte använda den funktionen. GitHub har också ett inbyggt ärendehanteringssystem som jag i framtiden sannolikt kommer att använda för att hålla koll på fel och fixar i koden. Anledningen till att jag inte har använt mig av detta än är att koden inte är klar, och det finns inga särskilda fel i en kod som inte är färdig. GitHub har även ett förenklat ärendehanteringssystem likt Trello, med ”ärendebrickor” som kan flyttas runt på olika tavlor för att markera deras status.

Den enligt min åsikt bästa fördelen med GitHub är insticksprogrammet Gource, som kan skapa visualiseringar av ett projekts utveckling. Det innebär att jag i efterhand kan spåra projektets utveckling genom att titta på en relativt kort film, istället för att läsa mig igenom cirka 130 kommentarer att varierande kvalitet.

Det som inte går att se i Gource, är de kommentarer som skrivs när koden ändras. Kommentarerna jag har skrivit har oftast varit bra, med konkreta beskrivningar av det jag har gjort. Andra gånger har jag varit på dåligt humör och fyllt kommentaren till hälften med svordomar. Sådana kommentarer är inget som förenklar för den som försöker sätta sig in i programmets funktion.

## Framtida utveckling

Målet med projektet är ”att skapa en datorsimulering som simulerar hur främst ställverkslogiken och i andra hand tågklarerartjänsten kunde ha fungerat på en något större mötesstation under mitten av 1900-talet”. Det målet är inte uppfyllt. Koden är långt ifrån bra, och jag är nybörjare inom programmering. Jag uppmuntrar intresserande människor att söka kontakt med undertecknad via GitHub (<https://github.com/TNTPata>) för att få tillgång till koden så att vi tillsammans kan förbättra den.

Jag anser att projektet är klart när:

* Manöverpanelen är identisk med ett riktigt elektromekaniskt ställverk.
* Funktionen är identisk med ett riktigt ställverk.
* MRO Säo eller TTJ (eller andra regler) simuleras fullt ut. Vid regelbrott straffas spelaren genom poängavdrag eller liknande. Poäng erhålles genom att utföra åtgärder korrekt, eller tåg som avsänds i rätt tid.
  + Tåganmälan till och från grannstationer fungerar som det gör på riktigt.
  + Ordergivning med avsedda blanketter finns.
  + Det går att ta in tåg på stationen genom andra förfaringssätt än genom att ställa infartssignalen i ”kör”.
* Det går att simulera blockapparater eller linjeblockering.
* Koden går att expandera
  + Och med relativ lätthet göras om för andra stationer, både mindre i och större skala.
  + Och kan simulera andra stationer (med mellanliggande linje) för antingen en ensam spelare eller flera spelare över internet.
* Programmet kan användas i utbildningssyfte, både för allmänheten och för tågklarerar-elever.
* Det finns en Virtual Reality-version för att korrekt kunna ge handsignaler.

# Litteraturförteckning

Banverket. (den 1 Juli 1999). BVF 544.94005. *Kontrollås och kontrollåsnycklar*.

Kungliga Järnvägsstyrelsen. (1916). *Banlära - Järnvägars byggnad och underhåll* (Vol. II). (O. Hemming, Red.) Stockholm: Kungliga Järnvägsstyrelsen.

Larsson, A. (den 1 Januari 2011). ÖSlJF 319. *Signalteknik - ÖSlJ signalsäkerhetsanläggningar(1)*. Östra Södermanlands Järnväg.

Larsson, A. (den 1 Maj 2013). ÖSlJH 31. *Signalteknik – utdrag ur signalteknisk handbok(1)*. Östra Södermanlands Järnväg.

Larsson, A. (den 1 Maj 2013). ÖSlJH 32. *Signalteknik – teckenförklaringar till växel- och signalanläggningar(1)*. Östra Södermanlands Järnväg.

Microsoft Corporation. (u.d.). *.NET Documentation*. Hämtat från Microsoft Docs: https://docs.microsoft.com/en-us/dotnet/#pivot=docs&panel=getstarted

*Monogame Community*. (u.d.). Hämtat från Monogame.net: http://community.monogame.net/

Monogame. (u.d.). *Monogame Documentation*. Hämtat från Monogame.net: http://www.monogame.net/documentation/?page=main

Sundberg, B. (den 1 Januari 2011). ÖSlJF 310. *Signal- och ställverksanvisningar(2)*. Östra Södermanlands Järnväg.

Trafikverket. (den 1 Juni 2019). *Regelmoduler i Trafikbestämmelser för järnväg: Modul 3 HMS: Signaler - System H, M, S.* Hämtat från Trafikverket.se: https://www.trafikverket.se/contentassets/26967a3da3b9441daafc5959743c5161/3hms\_signaler\_hms\_20190601.pdf den 5 April 2019

Trangius, K. (2013). *Programmering 2 C# - Spelprogrammering med XNA och MonoGame.* Lidköping: Thelin Läromedel.

# Bilageförteckning

Bilaga 1, Signalplanritning

Bilaga 2, Förreglingstabell

Bilaga 3, Kontroll av växel

Bilaga 4, BVF 544.94005: Kontrollås och Kontrollåsnycklar

Bilaga 5, ÖSlJH 31: Utdrag ur signalteknisk handbok

Bilaga 6, ÖSlJH 32: Teckenförklaringat till växel- och signalanläggningar

Bilaga 7, Banlära, 2:a bandet