

Programmera ett övergångsställe

I lektionen programmeras en algoritm för att styra trafikljus i en korsning. Eleverna får skapa en praktisk applikation och lära sig att skapa och modifiera algoritmer.



Lektionsförfattare: [Måns Jonasson](#)

Till läraren

1. Diskutera hur ett övergångsställe fungerar som tekniskt system
2. Programmera en algoritm som styr övergångsstället i Scratch
3. Avslutande reflektion
4. Extrauppgift: Programmera ett trafikljus med Javascript
5. Nästa lektion

LÄRARINSTRUKTIONER

Lektionens syfte

Syftet med lektionen är att låta eleverna programmera algoritmer för att lära sig styra tekniska system och öva på programmering, samt att ge dem förståelse för användningen av algoritmer i tekniska system i vår vardag.

Förberedelser

En förutsättning för att göra denna övning är att eleverna har grundläggande kunskaper i hur [blockprogrammeringsverktyget* Scratch](#) fungerar som programmeringsspråk. Du hittar fler lektioner där verktyget Scratch används om du tittar under [resurssidan för Scratch](#).

Denna lektion förutsätter att eleverna är bekanta med begreppet algoritmer, gärna genom att först ha genomfört lektionen "[Undersök och diskutera algoritmer](#)".

Tänk på att det är viktigt att förmedla att programmering handlar om att prova, testa och göra om. Misstag och fel är nödvändiga delar för att kunna lösa problem. Prata gärna om begreppet bugg, genom att använda material från lektionen "[Buggar eller fel vid programmering](#)".

*Blockprogrammeringsverktyg är ett verktyg där eleverna arbetar med programmering genom att använda block som sammanfattar kodsnuttar i "vanlig" kod. Scratch är utvecklat av [MIT – Massachusetts Institute of Technology](#).

Genomförande

Titta på [Scratchprojektet](#) tillsammans med eleverna och låt dem sedan remixa produktionen för att skapa en algoritm som löser uppgiften.

LÄROPLANSKOPPLING

Centralt innehåll i ämnet teknik (årskurs 7-9)

Arbetssätt för utveckling av tekniska lösningar

- Tekniska lösningar som utnyttjar elektronik och hur de kan programmeras.
- Teknikutvecklingsarbetets olika faser: identifiering av behov, undersökning, förslag till lösningar, konstruktion och utprovning. Hur faserna i arbetsprocessen samverkar.
- Hur digitala verktyg kan vara stöd i teknikutvecklingsarbete till exempel för att göra ritningar och simuleringar.
- Programmering i olika programmeringsmiljöer.

Centralt innehåll i ämnet matematik (årskurs 7-9)

Matematisk problemlösning

- Hur algoritmer kan skapas, testas och förbättras vid programmering för problemlösning.

VAD KRÄVS

Dator

Lektionsdel 1:

Diskutera hur ett övergångsställe fungerar som tekniskt system

Instruktioner

Läs och diskutera i grupper eller par.

Titta på filmen för att få en överblick av uppgiften

[Se media](#)

Ett övergångsställe med trafikljus för bilar och fotgängare är en del av ett tekniskt system för att reglera flöden av bilar, cyklar och människor samt att minska antalet skador.

- Vilka delar skulle kunna sägas ingå i systemet, utöver själva lamporna?
- Diskutera kopplingen till andra tekniska system i samhället som måste finnas för att ett övergångsställe ska fungera.

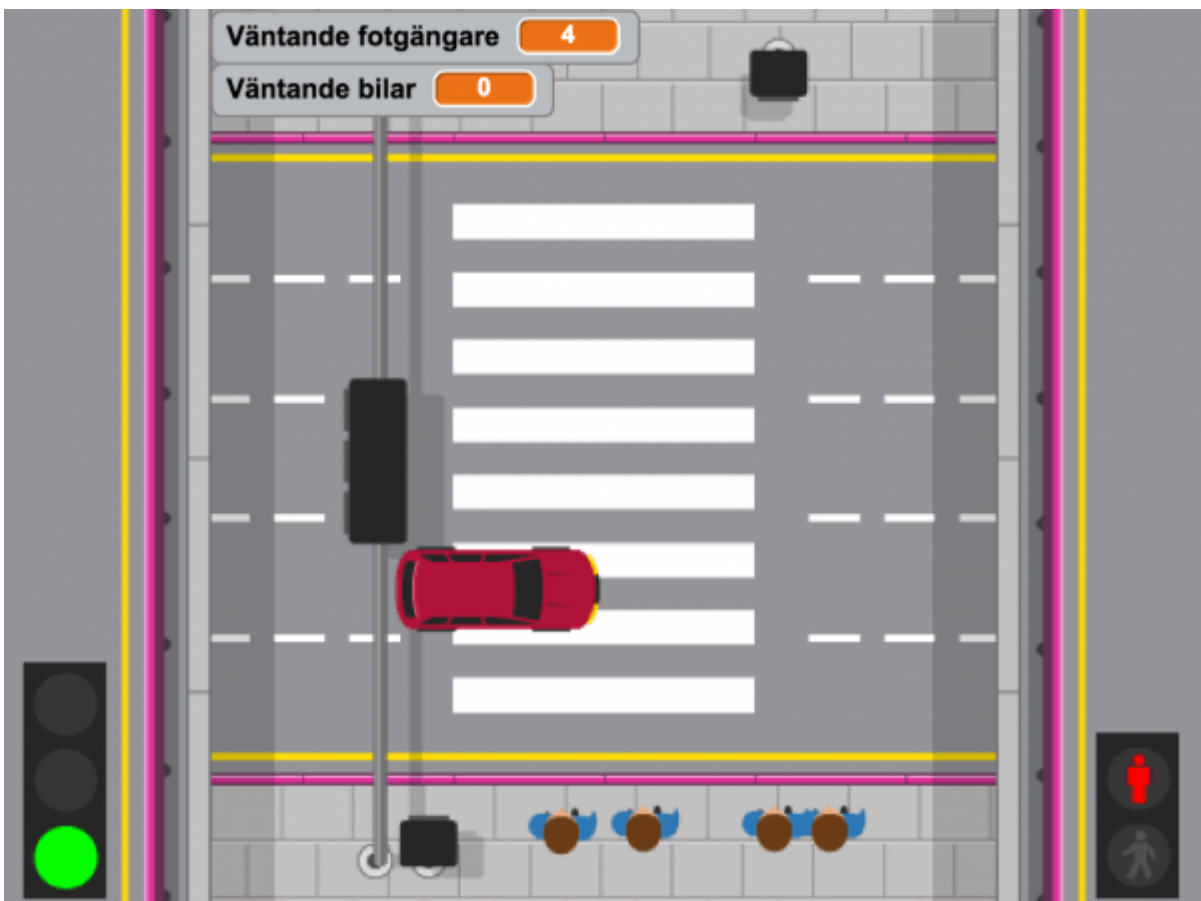
Lektionsdel 2:

Programmera en algoritm som styr övergångsstället i Scratch

Instruktioner

Jobba i par eller lös uppgifterna individuellt.

Surfa in på Scratch-projektet (<https://scratch.mit.edu/projects/214211923/>) och jobba självständigt för att lösa uppgiften.



I [övningsprojektet på Scratch](#) kommer fotgängare gående mot ett övergångsställe som

visar rött. Bilar kör förbi, för de har grönt ljus. Du ska programmera algoritmen som växlar trafikljusen så att flödet mellan fotgängare och bilister blir optimalt.

När antalet väntande fotgängare blir för stort får du ett varningsljud. Samma sak händer om det blir för många bilar i kö. Målet är att automatisera trafikljusen så att de växlar vid vissa intervaller, eller när antalet väntande bilar och fotgängare håller på att bli för högt.

1. Börja med att fundera på den bästa metoden. Skriv pseudokod* på papper eller i en textredigerare.
2. I Scratchprojektet finns redan ganska mycket kod. Gå igenom den och försök att förstå vad varje del utför.
3. Testa att köra igång projektet och se vad som händer. Analysera varför fotgängarna blir påkörda.
4. Leta reda på var i projektet algoritmen som styr trafikljusen ligger.
5. Utvärdera koden för att förstå hur den fungerar. Testa att ändra några av värdena för att ändra funktionen.
6. Testa! Utvärdera. Förändra koden. Testa igen!

I Scratch-projektet finns två variabler som automatiskt styr om trafikljusen. Genom att ändra innehållet i dessa variabler kan du styra ljusen, och därmed stoppa bilar eller fotgängare.

Du kommer att behöva använda följande block:



* Pseudokod är ett sätt att börja tänka i programmering utan att behöva skriva riktig kod. Pseudokoden kan vara uttryckt på vanlig svenska. Ett exempel för denna övning skulle kunna börja så här:

```
Om antalet fotgängare är högre än 10 så  
växla till grön gubbe  
växla till rödljus för bilarna
```

Lektionsdel 3:

Avslutande reflektion

Instruktioner

Presentera era lösningar inför klassen och berätta hur ni tänkt när ni löst uppgiften. Jämför de olika lösningar som tagits fram under lektionen med varandra. Diskutera vilken/vilka som löst uppgiften på ett effektivt sätt.

Reflektera över att det går att lösa samma uppgift på många olika sätt med algoritmer och programmering. Diskutera gärna hur olika programmerare kan lösa uppgifter på olika sätt och hur det kan påverka vårt samhälle.

Lektionsdel 4:

Extrauppgift: Programmera ett trafikljus med Javascript

Instruktioner

Scratch är en utmärkt miljö för att lära sig programmeringslogik. För en mer avancerad nivå kan den här övningen vara ett bra alternativ.

Javascript-övningen [trafikljus](#) går att köra i en dator med webbläsare. Övningen är framtagen av Mikael Tylmad, författare av läromedel i Javascript.

[Se video](#)

Lektionsdel 5:

Nästa lektion

Instruktioner

Arbeta vidare med programmering och matematik i lektionen: [Matematiska lägesmått med en micro:bit](#)



Klar? Klicka här