

Programmera ett kärnkraftverk

I lektionen programmeras en algoritm för att styra processen i en reaktor i ett kärnkraftverk. Eleverna får skapa en praktisk applikation och lära sig att skapa och modifiera algoritmer.



Lektionsförfattare: [Måns Jonasson](#)

Till läraren

1. Hur verkar algoritmerna i tekniska system?
2. Hur fungerar ett kärnkraftverk?
3. Programmera en algoritm som maximerar effekten i ett kärnkraftverk
4. Avslutande reflektion
5. Nästa lektion

LÄRARINSTRUKTIONER

Lektionens syfte

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

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 produktionen i Scratch 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)

Arbetsätt för utveckling av tekniska lösningar

- Tekniska lösningar för styrning och reglering av system. Hur mekanisk och digital teknik samverkar, till exempel i värme- och ventilationssystem.
- 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

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

VAD KRÄVS

Dator

Lektionsdel 1:

Hur verkar algoritmerna i tekniska system?

Instruktioner

Läs igenom avsnittet och besvara frågorna.

Inom industrin skedde datoriseringen i stor skala från 1970-talet och framåt. Sedan dess har allt mer av styrningen blivit datoriserad, med en utveckling från enklare moment och industrirobotar till helt automatiserade fabriker.

Diskutera med eleverna hur programmeringen är central för styrningen av en fabrik eller ett kraftverk. Hur många exempel på datoriserade processer kan eleverna nämna?

Frågor att besvara

1. Varför tror du att antalet automatiserade moment inom industri fortsätter att öka?
2. Vilka industriarbeten kommer att skötas av människor om 50 år?
3. Hur kan innovationer inom programmering påverka tillverkningsindustrin?

Lektionsdel 2:

Hur fungerar ett kärnkraftverk?

Instruktioner

Titta på filmen och besvara frågorna.

[Titta tillsammans på filmen "Fatta fakta. Kärnkraft"](#) från UR för att få en inblick i hur ett kärnkraftverk fungerar och kan alstra elektricitet. Diskutera gemensamt hur kontrollmekanismerna i ett kärnkraftverk fungerar och vad som krävs av maskiner som styrs automatiskt.

Fundera på hur fel i programmeringskoden som kontrollerar ett kärnkraftverk kan påverka oss.

Exempel på frågor att besvara:

- Att programmera viktiga samhällsfunktioner är ett viktigt jobb. Vilka krav ställs på den som ska programmera maskiner som är så viktiga för samhället?
- Om något går fel med koden i ett kärnkraftverk kan det få väldigt allvarliga konsekvenser. Kan du komma på andra automatiserade funktioner i samhället som är programmerade och skulle skapa väldigt stor skada eller oreda om det blev fel i koden?

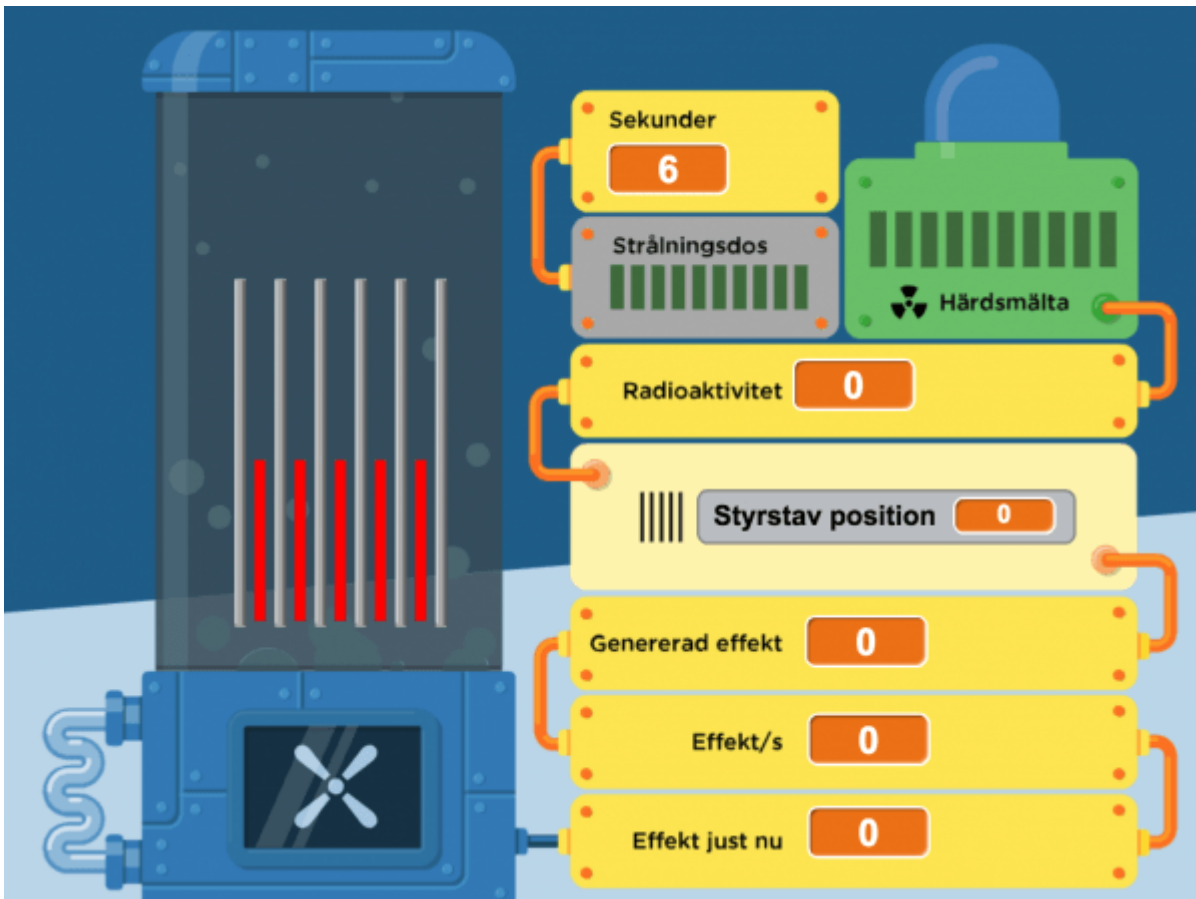
Lektionsdel 3:

Programmera en algoritm som maximerar effekten i ett kärnkraftverk

Instruktioner

Jobba i par eller lös uppgifterna individuellt. Surfa in på Scratchprojektet

(<https://scratch.mit.edu/projects/215177252/>) och jobba självständigt för att lösa uppgiften.



I [övningsprojektet på Scratch](#) ser du en reaktor i ett kärnkraftverk i genomskärning. Processen i ett kärnkraftverk startar när styrstavarna höjs upp, vilket låter de radioaktiva partiklarna flöda mellan bränslestavarna. Ju längre upp du hissar styrstavarna, desto fler

reaktioner kan ske. Men om du flyttar stavarna för långt upp går det för snabbt och strålningen ökar över godkända nivåer.

Din uppgift blir att programmera en algoritm som automatiskt håller bränslestavarna på nivåer som dels maximerar mängden genererad energi, dels ser till att strålningen hålls inom acceptabla nivåer.

Håll koll på de två mätarna som visar total strålningsdos och risken för härdsmälta! Om de slår i taket är det game over.



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. Hitta var i projektet algoritmen som påverkar styrstavarnas position hör hemma.
4. Börja förändra algoritmen för att automatstyra bränslestavarna utan att orsaka katastrof.
5. Testa! Utvärdera. Förändra koden. Testa igen!

Du kommer att behöva använda dessa block, bland andra:



* 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 effekten är lägre än X
växla höj bränslestavarna med Y

Lektionsdel 4:

Avslutande reflektion

Instruktioner

Jämför olika lösningar som tagits fram under lektionen med varandra. Diskutera vilken/vilka som löst uppgiften på ett effektivt sätt.

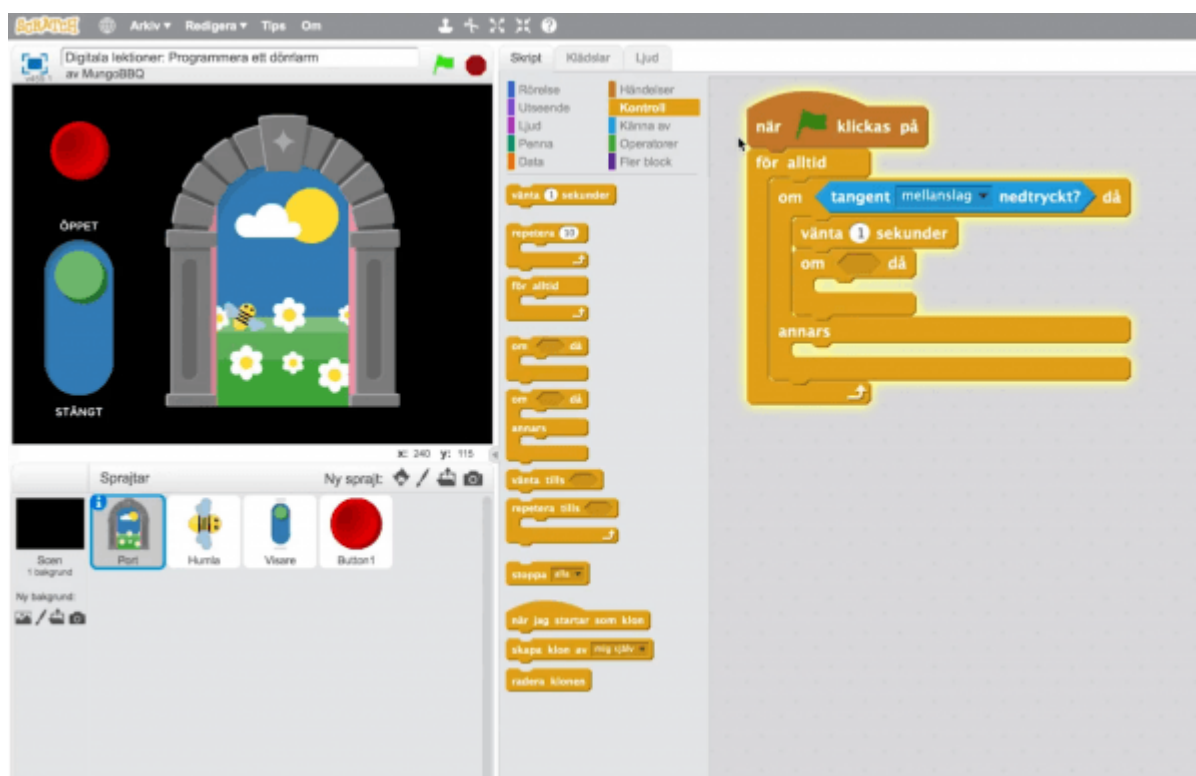
Presentera era lösningar och hur ni tänkt när ni löst uppgiften. Reflektera över att det går att lösa samma uppgift på många olika sätt med algoritmer och programmering. Diskutera gärna hur olika programmerare kan lösa uppgifter på olika sätt och hur det kan påverka vårt samhälle.

Lektionsdel 5:

Nästa lektion

Instruktioner

Utveckla dina kunskaper i att programmera tekniska system i lektionen: [Programmera ett dörrlarm | 1 av 2](#).



Klar? Klicka här