

T2 L13: Robotarmen

(Termin 2 Lektion 13)



Nu i denna lektion så ska vi göra en robotarm som man kan styra med 8 tangenter! Armen kommer bestå av tre delar och med dem så kommer man kunna plocka upp X och mata roboten.

Koncept i fokus: "lager", avancerade kontroller och funktioner.

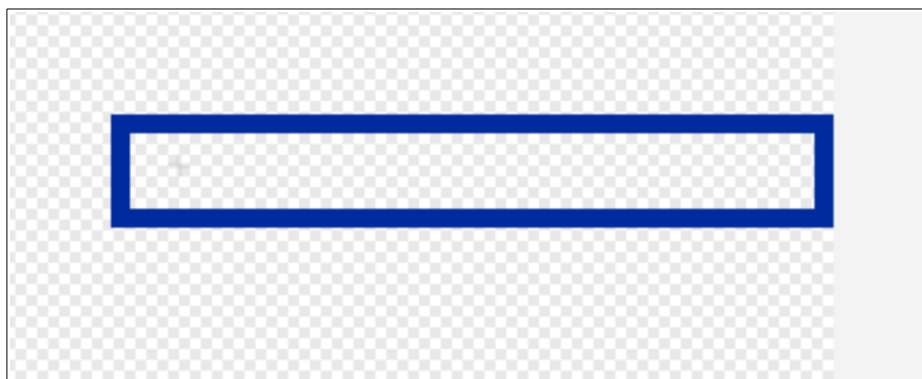
Delmoment 1: Rita armdelar

Vi börjar med att rita en bakgrund, ett robotansikte och sprajtarna som gör upp armen.

1. Radera katten.

2. Rita en ny sprajt, det kommer vara första delen av armen av roboten. Använd rektangelvertyget för att göra en rektangel i någon färg.

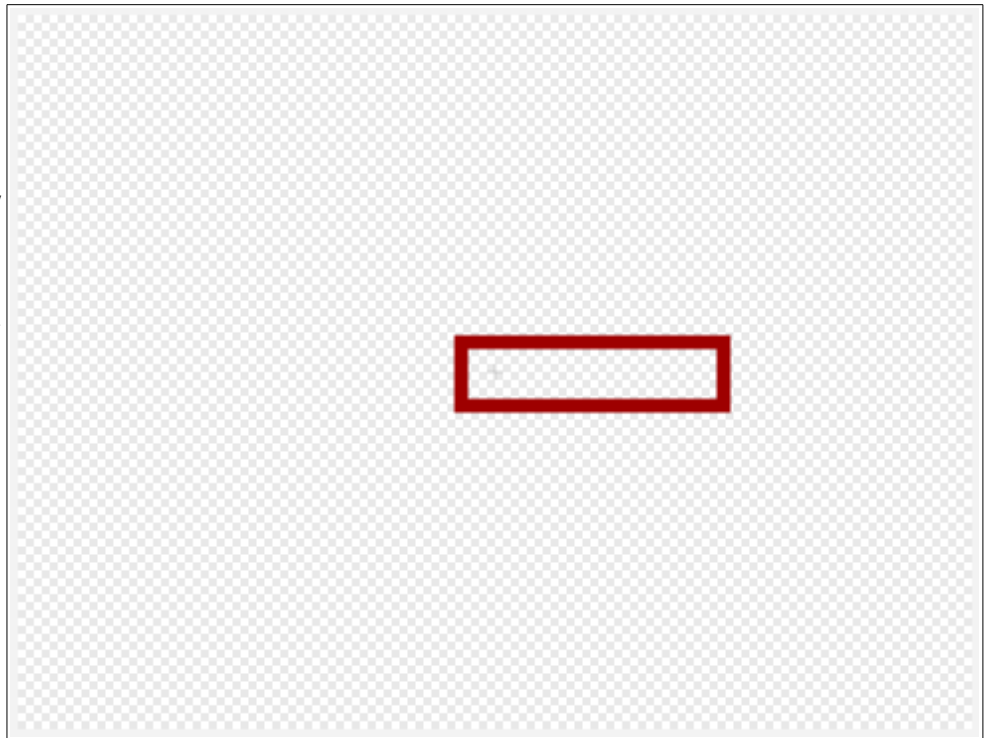
Den ska gå från mitten ända till höger kant.



Sen kan du fylla i den eller ändra den. Men behåll storleken på den.

3. Rita en ny sprajt, det kommer vara andra delen av robotarmen.

Den armen ska vara kortare så låt den bara vara hälften så lång som den förra.

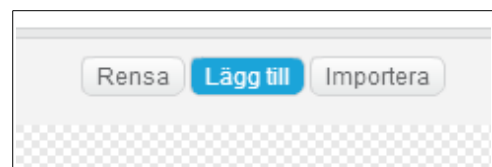


4. Klicka på "Konvertera till vektor" för att kunna ändra på bilden lättare.

5. Kopiera den korta sprajten så att du har två sprajtar som är lika långa. Kopiera klädseln på den nya sprajten så att den har 3 totalt. Ge klädslarna namnen:

"öppen", "stängd" och "fångad".

5. Välj klädseln fångad. Klicka nu på "Lägg till" högst upp för att lägga till någon bild från biblioteket. Välj nånting som roboten ska plocka upp och äta senare i spelet.

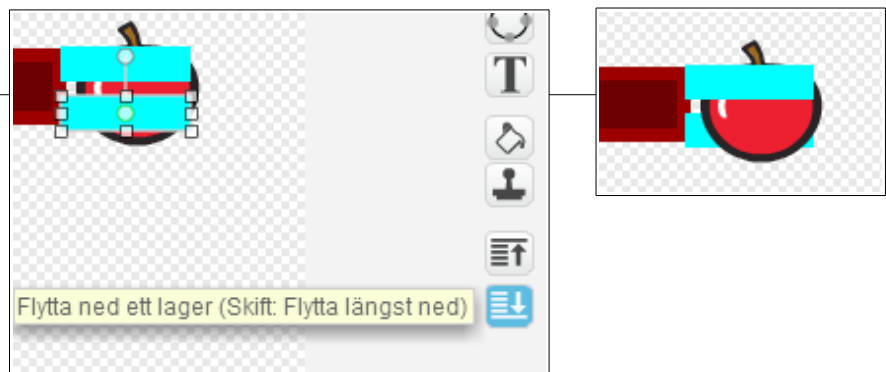


Jag valde ett äpple.

6. Rita sen en griphand som är enfärgad. Den ska bara ha en färg. Jag ritade en robothand som var turkos.

Eftersom vår bild är konverterad till vektor, så kan vi välja en del av bilden och klicka på "flytta ner ett lager". Jag valde den undre delen av handen och klickade på Flytta ned ett lager. Då ser det ut som att handen håller i äpplet.

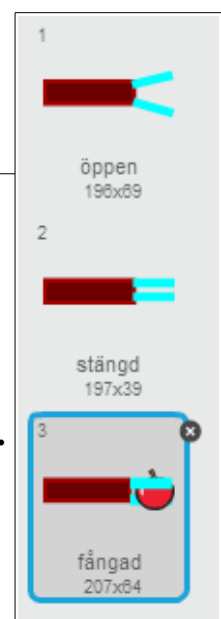
Fint!



7. Rita nu en öppen hand på klädseln "öppen" och rita en stängd hand som inte håller i något på klädseln "stängd".

Snyggt!

8. Namnge den första sprajten till "arm1" och så vidare.



9. Lägg till en sprajt, välj sprajten som griphanden tar tag i så att

den också finns i scenen. I mitt fall blir det äpplet.

Delmoment 2: Koda armarna

Nu ska vi koda hur man styr arm-delarna. Man kommer styra den blåa med piltangenterna upp och ner. Den röda med piltangenterna höger och vänster. Den sista med gripklon kommer att styras med w,a,s,d och mellanslag.

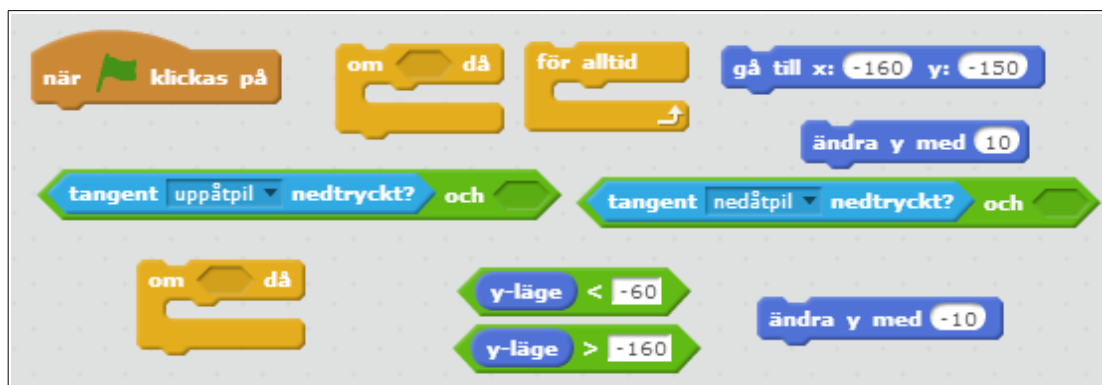
Vi ska nu skapa ett skript för arm1 som gör att man kan styra den upp och ner, men inte hela vägen.

9. Skapa ett skript för arm1 som:

- Startar när spelet startar
- gå till x:(-160) y:(-60)
- för alltid:

```
om( ( tangent( uppåtpil )nedtryckt? )och( ( y-läge ) < ( -60 ) ) ):  
    ändra y med( 10 )
```

```
om( ( tangent( nedåtpil )nedtryckt? )och( ( y-läge ) > ( -160 ) ) ):  
    ändra y med( 10 )
```



Pröva att spela spelet och trycka på piltangenterna för att se att man kan styra arm1 upp och ner.

Nu ska vi göra så att arm2 alltid följer med arm1, men att man kan styra den till höger och vänster med piltangenterna. Det kommer se ungefär ut som det förra skriptet fast det ändrar på **x-läge** istället!

10. Skapa ett skript i arm2 som:

- Startar när spelet startar
- sätt x till(50)
- peka i(0)riktning
- för alltid:

```
om( ( tangent( högerpil )nedtryckt? )och( ( x-läge ) < ( 50 ) ) ):  
    ändra x med( 10 )
```

```
om( ( tangent( vänsterpil )nedtryckt? )och( ( x-läge ) > ( -160 ) ) ):  
    ändra x med( 10 )
```

```
sätt y till( ( y-läge )av( arm1 ) )
```



Kan man nu styra arm 1 och 2 med piltangenterna? Ser det ut som att dom sitter ihop? Om arm2 far utanför arm1 någon gång, så kan du ändra på skriptet i arm2 så att den sätter andra gränser på **x-läge**

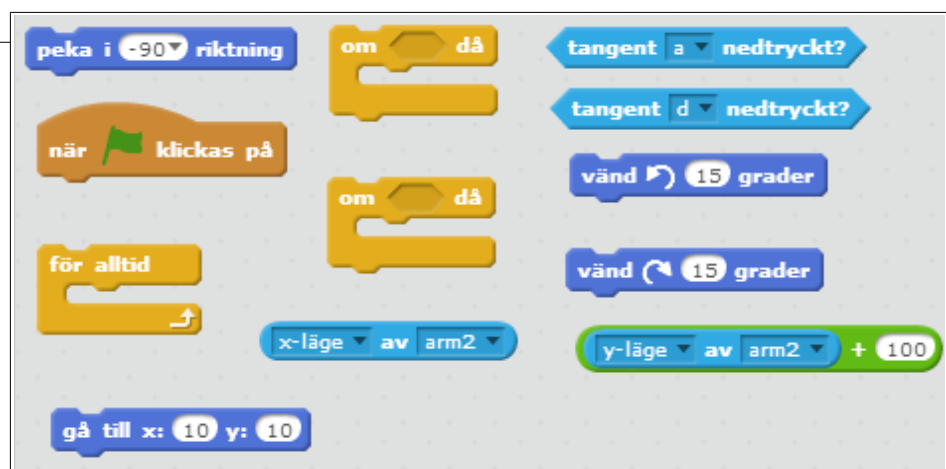
Kanske 50 och -160 är för höga eller låga tal? Om delarna alltid sitter ihop så är dem ganska bra i alla fall. Prova tills du är nöjd.

Nu ska vi koda den sista armen med griphanden. Vi ska göra så att man kan vrida armen med tangenterna a och s. Med mellanslag ska man kunna öppna och stänga gripklon. Om gripklon stängs när den nuddar äpplet i mitt fall så ska den ta tag i äpplet genom att byta klädsel till rätt klädsel.

12. Skapa ett skript i arm3 som:

- Startar när spelet startar
- peka i (-90) riktning
- för alltid:
 - om(tangent(a) nedtryckt?):
 - vänd åt vänster(15)grader
 - om(tangent(d) nedtryckt?):
 - vänd åt höger(15)grader
 - gå till x:((x-läge)av(arm2)) y:((y-läge)av(arm2) + (100))

Pröva att spela och se om du kan styra hela armen och nå med klon över nästan hela scenen.



Bra!

Nu ska vi göra ett litet skript så att det ser ut som att man plockar upp äpplet eller det roboten ska plocka upp i ditt spel. Vi ska göra det genom att skicka ett meddelande till äpplet när klon stänger sig. Sen ska äpplet kolla om den rör färgen som är vid klon. Om äpplet rör

vid den färgen så får klon byta till klädseln som har äpplet. Och äpplet kan då försvinna så det ser ut som att armen plockar upp det.

Vi börjar då med ett skript för mellanslag i arm3.

13. skapa ett skript i arm3 som:

- Startar när mellanslag trycks ner
- om((klädselnummer)=(1)) då:
spela ljudet(Slagverk/hand clap)
byt klädsel till(stängd)
skicka(stängd)
- annars:
byt klädsel till(öppen)



Nu skickas det ett meddelande. Prova att spela för att se att du kan stänga och öppna gripklon. Man kan ännu inte fånga äpplet. Men snart!

Vi ska börja med att göra en funktion som gör att äpplet kommer fram på en slumpmässig plats.

14. Skapa en funktion "nyPosition" i äpplet som:

- göm
- vänta(2)sekunder
- gå till x:(slumpstal(-80)till(200)) y:(slumpstal(-170)till(170))
- visa

15. Skapa ett skript i äpplet som börjar när spelet börjar och direkt kör funktionen "nyPosition".

Så, nu kan vi använda den funktionen när vi tar emot meddelandet "stängd".

16. Skapa ett skript i äpplet som:

- Startar när den tar emot "stängd"
- om(rör färgen(färgen på gripklon))då:
 skicka("fångad")
 nyPosition

Så! Nu behöver vi bara göra en sista sak så att gripklon kan byta klädsel till när den lyckats fånga äpplet eller saken i ditt spel.

17. Gå in på arm3 och skapa ett skript som:

- Startar när den tar emot "fångad"
- byt klädsel till(fångad)

Testa spela nu! Kan du fånga äpplet med gripklon? Kommer äpplet fram igen så att du kan fånga det igen?

Bra då går vi vidare!

Delmoment 3: Robotens kropp, mun och ögon.

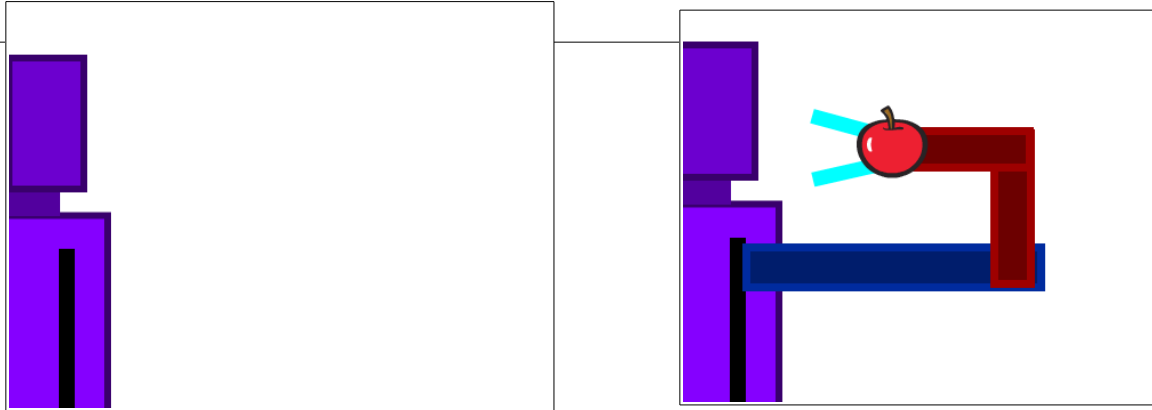
Vi ska nu rita 2 nya sprajtar. Ansikte och mun för roboten. Vi ska även rita bakgrunden där robotens kropp kommer vara.

Först ska vi rita själva kroppen. Den ska ritas på bakgrunden för kroppen behöver inte röra på sig. Spela spelet så att arm1 hamnar så högt upp som spelaren kan låta den åka. Sen så kan vi börja rita roboten.

18. Rita nu en robot eller något med en kropp och ett huvud. Rita

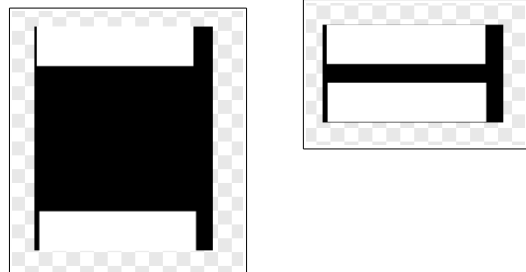
inte något ansikte. Det ska vi låta vara två egna sprajtar. Jag ritade en lila robot. Den fick ett svart spår där armen kan åka ner och upp.

Till vänster ser du min bakgrund och till höger ser du hur hela scenen ser ut med alla sprajtar.



Nu ska vi rita ett ansikte och en mun! Vi börjar med munnen.

19. Rita en ny sprajt. Om du väljer "Konvertera till vektor" så blir det lättare att animera. Jag ritade en svart mun som har två klädselar. En öppen och en stängd. Då kan jag sen animera så att munnen öppnas och stängs när roboten äter.



Eftersom jag valde att arbeta med vektor så var det enda jag behövde göra att kopiera min första mun och sen göra det svarta lite mindre och flytta tänderna.

Ge den första klädseln namnet "öppen mun" och den andra "stängd mun"

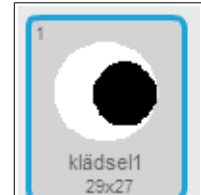
Flytta sedan munnen till där robotens mun ska vara och ge sprajten

namnet "mun".

Nu ska vi rita ansiktet!

20. Rita en ny sprajt. Det ska vara ett enda öga.

Rita en cirkel i mitten och en svart pupill i den så att ögat kollar till höger. Pupillen kan vara vilken färg du vill.

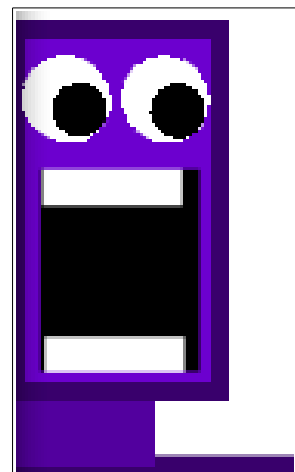


21. Skapa nu ett skript i ögat som:

- Startar när spelet startar
- för alltid:
 peka mot(arm3)

22. Kopiera nu sprajten som vi gjort (ögat) och placera det på robotens ansikte.

Pröva spela spelet och se om ögonen alltid kollar på armen!



Jag glömde byta till vektorläge innan jag ritade. Eftersom jag ritade mina ögon så små, så ser man kanten av pixlarna nu när dem är stora. Ögonen är lite knöliga.

Delmoment 4: Koda munnen och energi

Vi ska nu koda munnen så att den äter upp äpplet. Vi ska även göra en variabel som heter "energi" som ska öka när roboten äter äpple. Energin ska sakta minska hela tiden.

23. Skapa en variabel (för alla sprajtar) som heter "energi".

24. Skapa en funktion i mun som heter "tugga" som:

- * vänta(0.1)sekunder
- * nästa klädsel
- * spela ljudet(Effekter/chomp)
- * ändra(energi)med(1)
- * vänta(0.1)sekunder
- * nästa klädsel

Klicka på funktionen för att se att den fungerar, äter roboten och ökar "energi" ?

När man äter så tuggar man flera gånger, så nu ska vi göra en ät-funktion.

25. Skapa en funktion i mun som heter "ät" som:

- * repetera(6):
 tugga

Klicka på funktionen "ät" för att testa så att roboten tuggar flera gånger.

Nu ska vi använda funktionen "ät" i ett skript. Vi ska göra ett skript i munnen som testar om den rör armen som håller i äpplet samtidigt som armens klädsel är 3 och håller i äpplet. Om det är så så ska munnen animera att äta och skicka ett meddelande att den äter äpplet.

26. Skapa ett skript i mun som:

- Startar när spelet startar
- byt klädsel till(klädsel1)
- för alltid:
 - om((rör(arm3)) och((klädselnummer) av(arm3)) =(3)) då:
 - skicka("ät")
 - ät
 - byt klädsel till(klädsel1)



Nu ska vi se till att armen byter klädsel när roboten tuggar så att roboten inte kan äta för evigt på ett äpple!

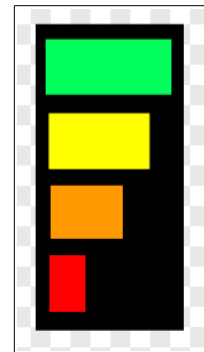
27. Skapa ett skript i arm3 som:

- Startar när den tar emot "ät"
- byt klädsel till(öppen)

Delmoment 5: Visualisera energi!

Vi ska göra en sprite som håller koll på hur mycket energi roboten har. Den ska visa med bilder hur mycket energi så att man slipper kolla på ett nummer.

28. Rita en ny sprajt. Konvertera den till vektor och rita först en svart avlång rektangel. Sen ritar du små rektanglar i den som på ett batteri på datorn.



29. Kopiera sen klädseln 4 gånger så att vi har 5 klädslar totalt.

Sen tar du bort färgerna så att vår energimätare minskar eftersom.

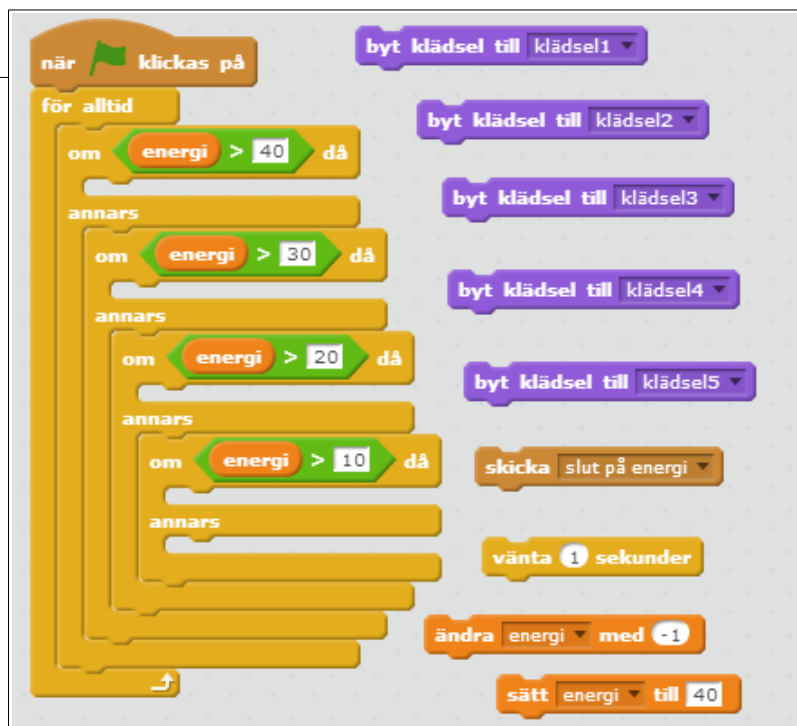


Snyggt! Ge sprajten namnet "energimätare"

Nu ska vi skapa ett skript i vår energimätare så att den visar olika mycket energi beroende på hur högt tal variabeln energi har.

30. Skapa ett skript i energimätare som:

- Startar när spelet startar
- sätt(energi)till(60)
- för alltid:
 - ändra(energi)med(-1)
 - om((energi) > (40)):
 - byt klädsel till(klädsel1)
 - annars:
 - om((energi) > (30)):
 - byt klädsel till(klädsel1)
 - annars:
 - om((energi) > (20)):
 - byt klädsel till(klädsel1)
 - annars:
 - om((energi) > (10)):
 - byt klädsel till(klädsel1)
 - annars:
 - byt klädsel till(klädsel5)
 - skicka("slut på energi")



Testa spelet och se om energimätaren ändras med tiden. Fylls den på när man äter?

När roboten får slut på energi så skickas meddelandet ”slut på energi”. Vad som händer då är helt upp till dig! Du kan göra så att roboten kör igång en animation så att den sprängs! Eller att roboten säger något roligt!

Delmoment 6: Fantastiska uppgraderingar!!!

Nu har vi kommit till uppgraderingarna. Vad tror du skulle göra spelet mer intressant? Här kommer förslag, men du får koda dina egna idéer såklart!

- Gör så att en annan spelare kan styra äpplet med musen! Då blir det en tävling!
- Gör en timer så att man bara har 60 sekunder på sig att äta. Mät antalet äpplen som äts upp!
- Gör så att roboten är kräsen och ibland vill äta ett äpple och ibland en apelsin! Så måste man plocka rätt sak!
- Gör en mus som roboten ska äta, gör så att musen vänder sig ifrån armen och försöker springa bort!

43. Sätt spelet i fullskärm och visa det för en vuxen!

Berätta vad för uppgraderingar du gjort!

Berätta hur man gör ögon som följer någonting!

Spelet ska heta **T2 L13** så att det är lätt att hitta! Spara!

Nu är du klar med hela lektion T2 L13 ! Nu kan du börja på lektion 14 (T2 L14) !