

Kod et tabeltrænings-spil

Af Britt Bang Sørensen og Aviaja Borup Lynggaard

Info til læreren:

Formål:

At eleverne lærer at programmere samtidig med at de arbejder med matematiske færdigheder og begreber - fx. fagudtryk som faktor og produkt.

Før aktiviteten:

Introducer eleverne til Hopspots og Scratch/CodingLab. Opret en lærerkonto i Scratch, hvis I vælger dette fremfor CodingLab i skoleTube (beregnetid til dette - den skal først godkendes). Lav grupper af max 3 elever.

Under aktiviteten:

Lad evt. eleverne starte med den foreslåede kode. Lad dem selv lave sprites og baggrunde, hvis de har interesse i det. Start med version 1 og fortsæt til version 2. Snak undervejs om der var andre og bedre måder at løse kodningen på.

Har eleverne lyst, kan de jo selv arbejde med spilflowet - skal der for eksempel være nedtælling - eller skal det være på tid - hvor lang tid er du om at løse 10 gange stykker?

Efter aktiviteten:

Variationer: Tilføj også 0 og 10 som kostumer. Prøv også med division. Lad eleverne prøve spillet med en mindre årgang og tilpas spillet til dem, f.eks. med plus og minus. Peer to Peer.

Evaluerings:

Afrund forløbet med at lade eleverne selv reflektere over deres spil samt at give feedback på de andre gruppers spil. Sammenlign evt. hvor forskellige de er blevet.

Kod et tabeltrænings-spil

Lav dit eget spil, hvor man træner tabeller samtidig med at man bevæger sig.
Her er et forslag til kodning - men du kan helt sikkert forfine koden og grafikken.

Varighed: 3-4 lektioner

Fag: Teknologiforståelse, matematik

Teknologi: Hopspots med Key:connector og Scratch

Kodning: 10-15 år

Spillet: Elever som arbejder med gange (fra ca. 4. klasse)

Materialer:

En computer (og en account til Scratch eller CodingLab i SkoleTube)

11 hopspotsbrikker: en hopspots-brik til hver af tallene fra 0 til 9 samt en "Enter" eller vi kan kalde det "send" hopspots-brik.

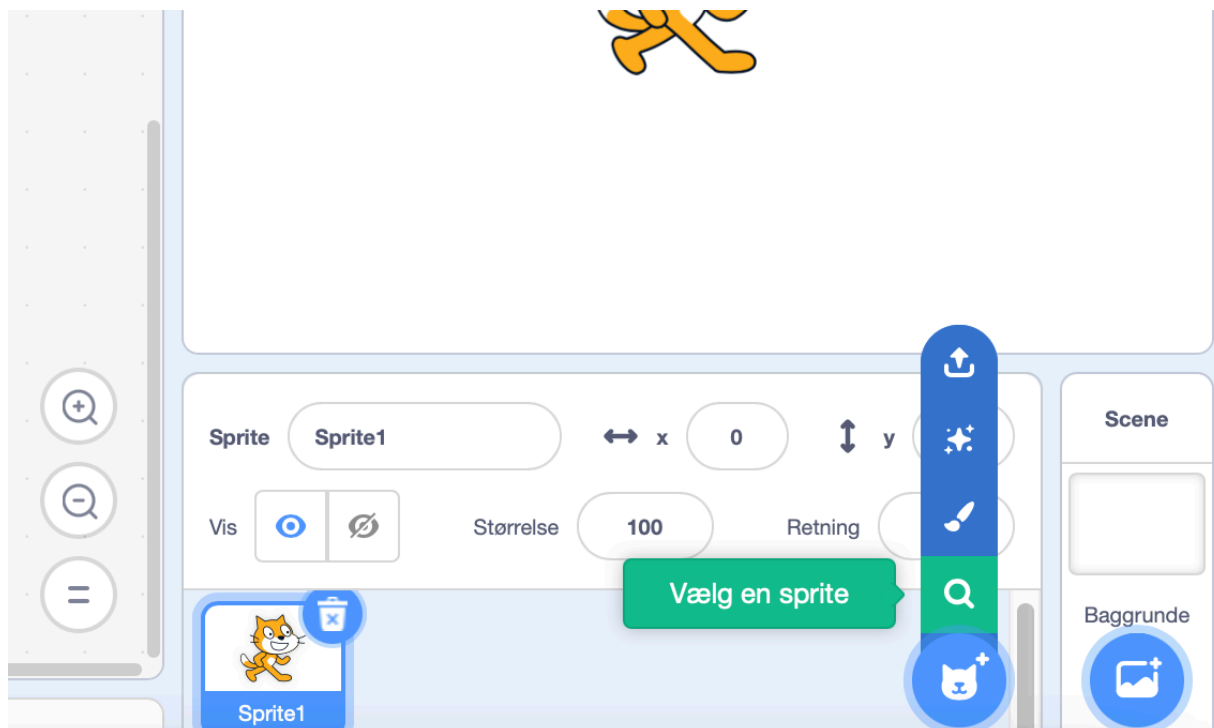
En Key:connector, som kan koble hver hopspotsbrik til et tal på tastaturet.

Det er nok også en god ide at laminere de 10 tal, som kan ligge ud for hver hopspots-brik.

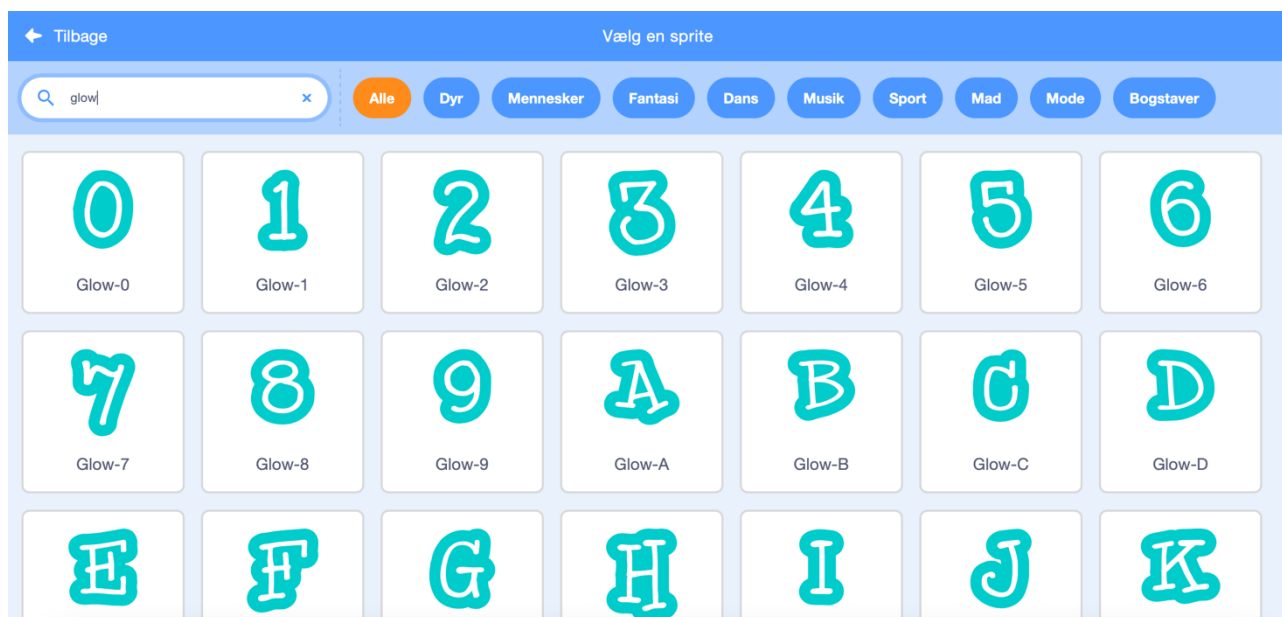
Kodning i Scratch

Sprites:

Du skal bruge 4 sprites - en sprite som ligesom styrer spillet (f.eks. katten), en sprite som viser hvert tal (faktor1 og 2) samt en sprite til gangetegnet (Gange). Start med "Vælg en sprite"

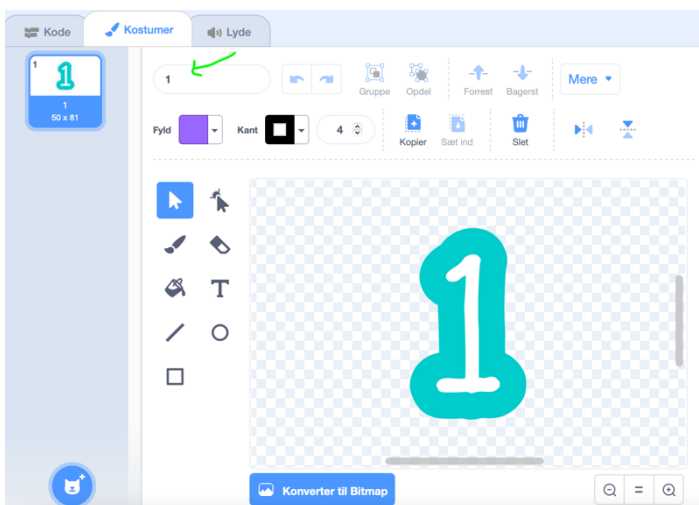
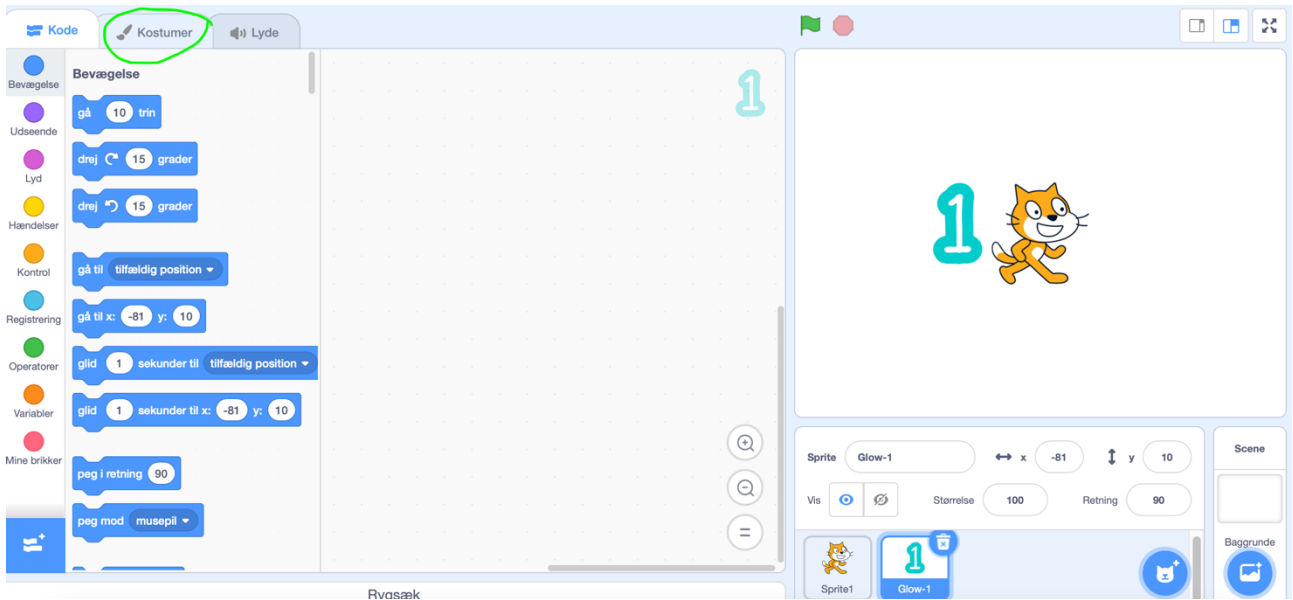


Der er allerede en sprite, du kan anvende til tallene - når du opretter din sprite, så søg på GLOW:



Du skal lave en sprite med 9 forskellige kostumer – nemlig tallene 1-9. Start med at vælge Glow-1.

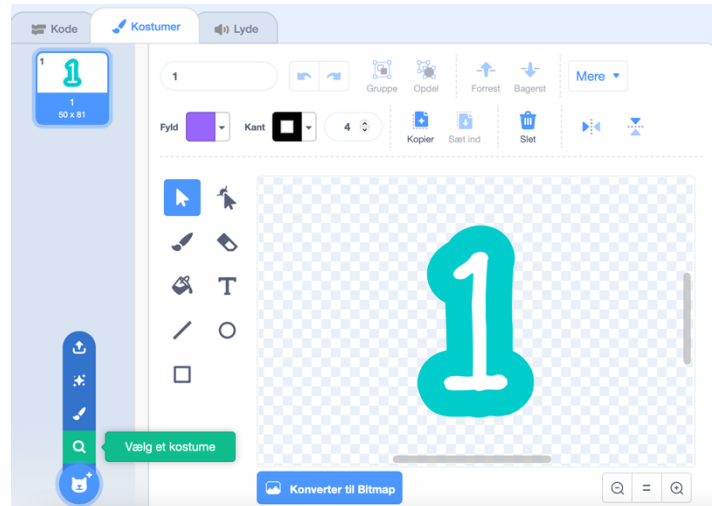
Når du står på spriten, kan du gå ind under fanen Kostumer i øverste venstre hjørne.



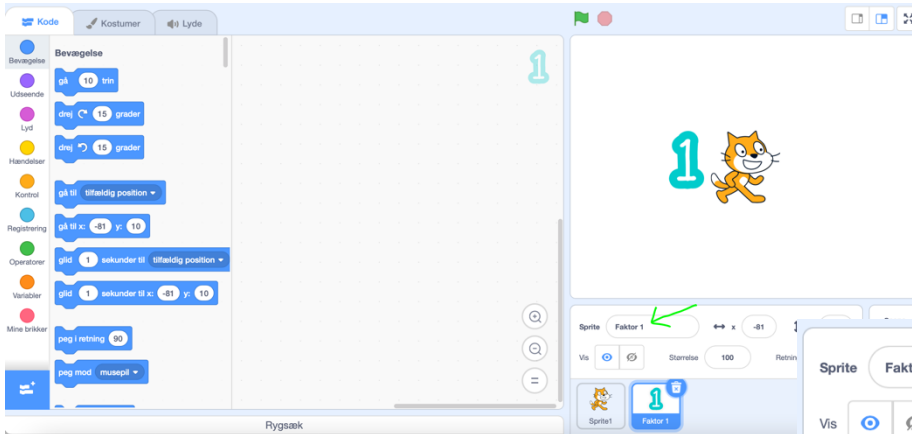
Du skal kalde kostumet det samme som tallet -altså kostume1 - som viser tallet 1, hedder også '1'.



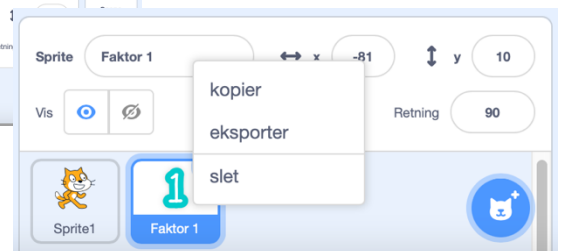
Herefter går du ind under 'Vælg et kostume' og finder de andre glow tal. Sørg for at navngive alle kostumerne, så de hedder det samme som det tal de viser.



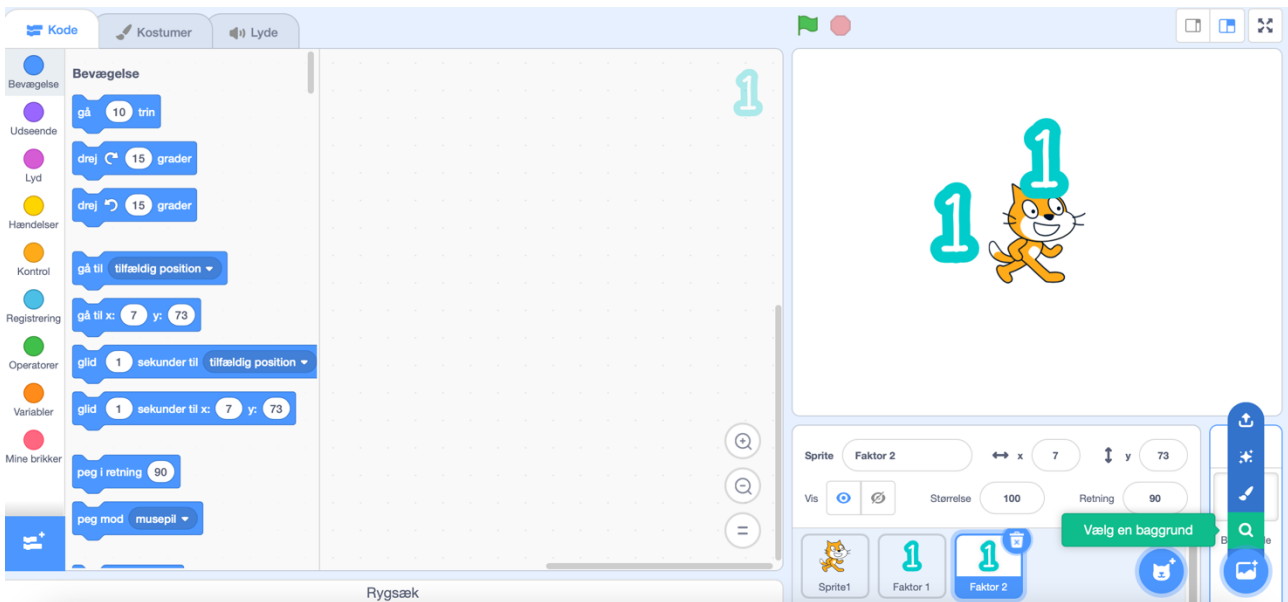
Kald Spriten for faktor 1 og kopier den til en ny sprite, som skal hedde faktor 2. Det gør man ude under 'kode' fanen.



Her højreklikker du på spriten og vælger kopier.



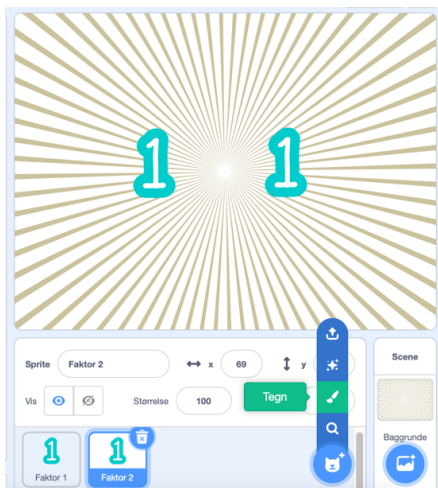
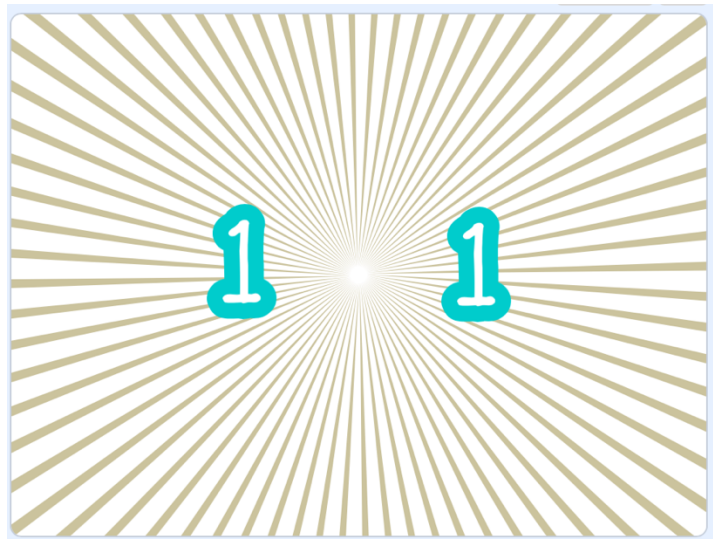
Du kan også vælge en baggrund, som du kan lide til din scene.



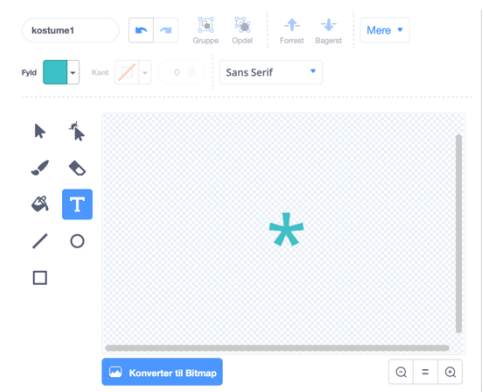
Jeg vælger 'Rays'.

Flyt tallene, så de står ud for hinanden.

Vi skal nu tegne en sprite for at få et gangetegn imellem tallene. Vælg 'Tegn' ud for en ny sprite. Du kommer nu ind under Kostume og her vælger jeg 'T' for tekst. Jeg kan nu skrive et gange-tegn: *



Ved at trække i hjørnet, kan jeg gøre den større. Jeg kan ændre farven og skrifttypen. Prøv dig frem til det er et flot gange-tegn.



Selve kodningen

Først skal du bruge nogle variabler. Du skal bruge en variabel til de to tal, du skal gange med hinanden – vi kan kalde dem faktor1 og faktor2. Så skal du også bruge en variabel, hvor du opbevarer produktet af de to tal -altså resultatet - vi kalder den resultat. Sidst skal du have en tæller, som viser antal rigtige - vi kalder den point. Klik variabernes visning væk, ved at fjerne det blå tjek udfor alle undtagen Point. Nu kan man kun se den ene variabel på skærmen og dermed fælg med i hvor mange point man har. Flyt Point-boksen hen, hvor du synes den står bedst. Jeg sætter min i øverste højre hjørne.

Variabler

Lav en variabel

faktor1

faktor2

my variable

Point

resultat

Senere kan vi bygge en timer og en highscore på - det er version 2. Alle skal først gennemføre version 1 og version 2 er så en udvidelse, man kan lave, hvis man har tid.

VERSION 1 – uden timer og highscore:

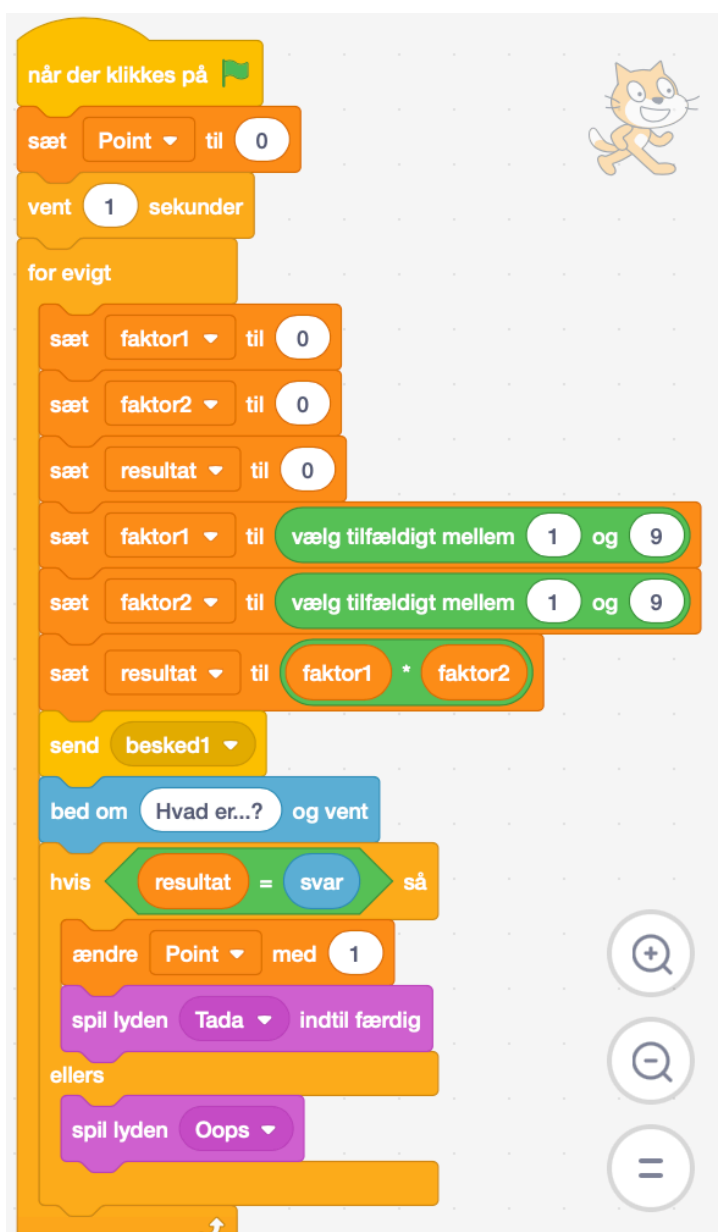
Start med at oprette de fire variabler: faktor1, faktor2, resultat og Point. Vi starter spillet ved at klikke på det grønne flag.

Koden ser således ud for den sprite, som fortæller hvad der sker i spillet:

Når der er klikket på det grønne flag, så kører spillet for evigt - vi har ikke indbygget en timer. Vi nulstiller Point og venter 1 sekund. Så nulstiller vi vores faktor 1 +2 og resultat, så vi er sikre på at vi har en ren tavle.

Der vælges nu et tilfældigt tal mellem 1 og 9 for de to faktorer. Vi udregner resultatet (produktet) af de to tal og gemmer det i vores variabel – **Resultat**

Herefter sender vi en besked (som modtages af vores 3 andre sprites) og beder derefter om et svar fra spilleren med kommandoen - bed om. Spillerens svar returneres i en slags variable **svar** - som vi ikke selv har lavet, men som Scratch laver til os. Vi sammenligner vores resultat med spillerens svar - og giver derefter en lyd tilbage, som viser spilleren, om der er regnet rigtigt eller forkert. Lyden kan du selv optage :-)



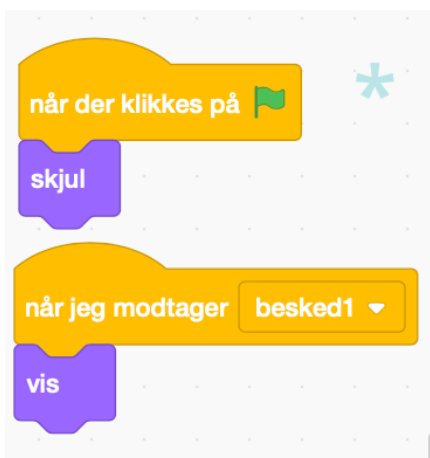
Er der regnet rigtigt, så lægger vi 1 til vores variabel **Point**

Men hvad sker der egentlig med vores sprites, når vi sender besked1? Og hvordan vises regnestykket til spilleren? Det er faktisk meget enkelt – vi viser spriten med det kostumenavn, som står i faktor1 og faktor2 - derfor skal dine kostumer have navn som tallet.

Stil dig først på spriten, som viser Faktor 1 og sæt den til at skjule sig fra start og vise det kostume, som faktor1 variabelen vælger. Gør det samme ved at kode udfor Faktor 2 spriten:



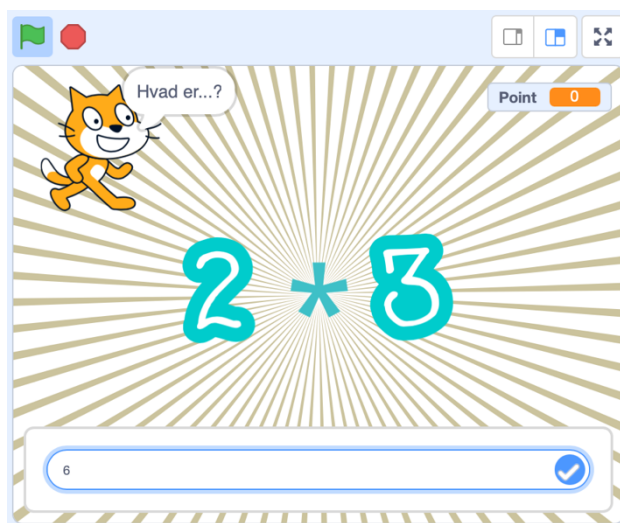
Gangetegnet vises bare samtidig - det har jo kun 1 kostume:



Stil dig på spriten, der viser gangetegnet, når du laver koden for dette. Her skal du have den til at skjule sig ved start og vise sig, når der sendes besked1.

Nu kan spillet spilles. Start ved at trykke på det grønne flag. Tast svaret og send afsted på 'Enter'.

Min skærm ser således ud:



Nu kan Hopspots-brikkerne sættes til. Skriv et tal til hver brik fra 0-9 eller print sidste side ud i dette hæfte, klip tallene ud og læg dem på en Hopspots-brik. Læg brikkerne på en lang række, med et tal udfor hver brik. Læg også en Enter-brik ud.



Tænd for Key:connectoren og forbind den til computeren, som var det et trådløst tastatur, ved at gå ind i Bluetooth indstillinger og tilføje et tastatur (det hedder Key:connector og det tal, der står bagpå jeres key:connector). Læg Key:connectoren på et bord, så du kan trykke på to taster på samme tid.



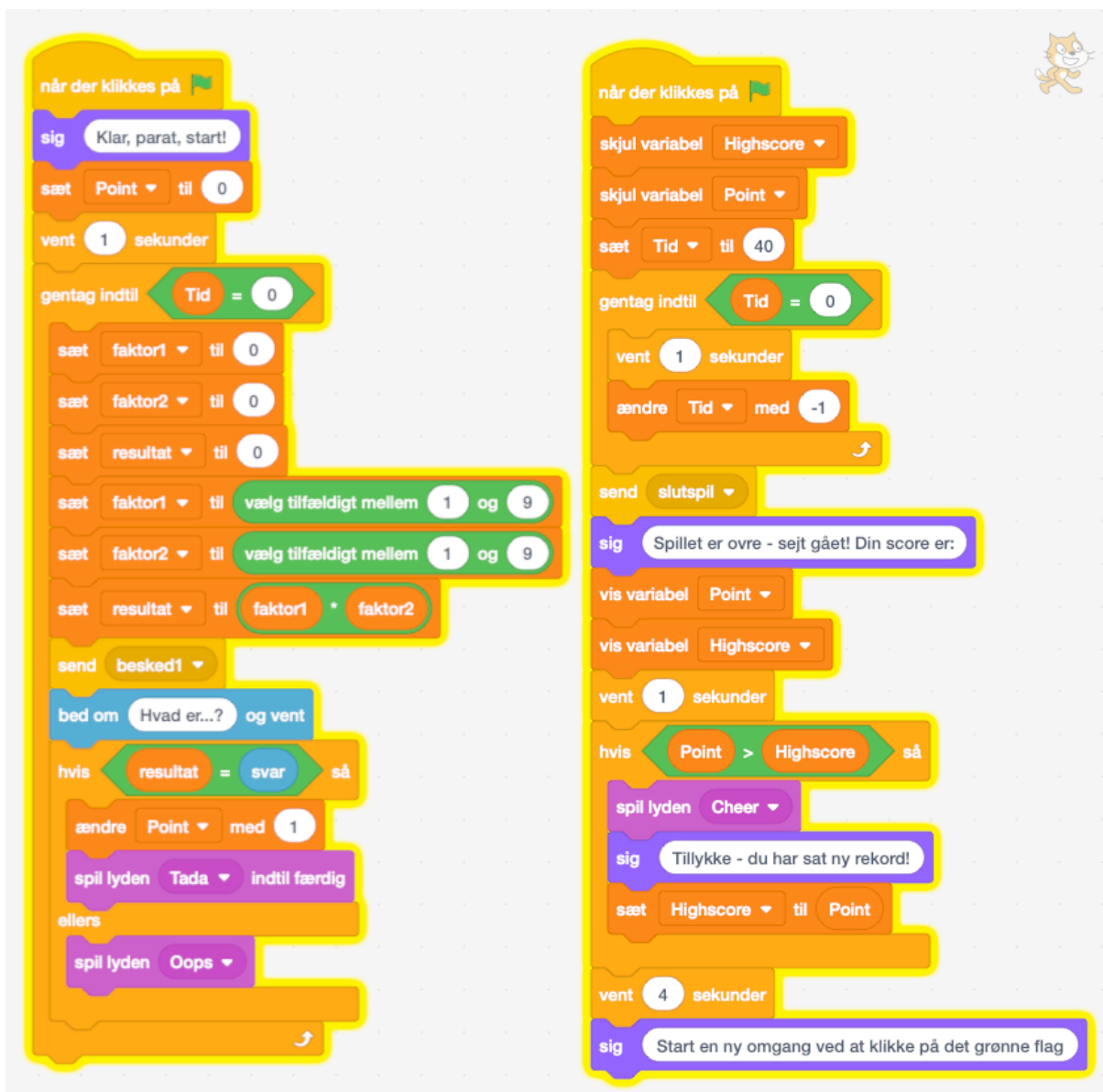
Tryk nu på "Connect"+"0" indtil 0 blinker på Key:connectoren. Træd på den brik, som skal være 0. Denne sender nu et 0 til computeren, hver gang der trædes på den. Afprøv det. Tildel alle tallene på samme måde, så de giver det tal som der står skrevet udfor dem. Lav ligeledes forbindelsen til "Enter" knappen. Prøv at spille spillet med fødderne!

VERSION 2– med timer og highscore (forlængelse af VERSION 1):

Når du skal have en timer på og en highscore, får vi brug for 2 nye variable: **Tid** og **Highscore**.

Du får også brug for endnu en blok med 'Når der klikkes på grønt flag', som jo kører parallelt med den gamle blok, så den skal laves ud for katten, som afvikler spillet.

I den nye blok starter du med at slukke for de variable, som du ikke vil have vist hele tiden. Du sætter **Tid** til det antal sekunder, dit spil skal køre. Jeg sætter den til 40. Herefter laver du en gentagløkke, hvor du hvert sekund tæller en ned i timeren. - det skal du selvfølgelig kun gøre, indtil timeren er på 0 - så løkken hedder gentag indtil **Tid = 0**

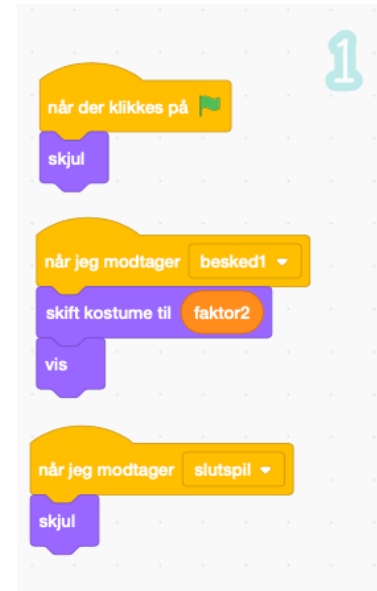
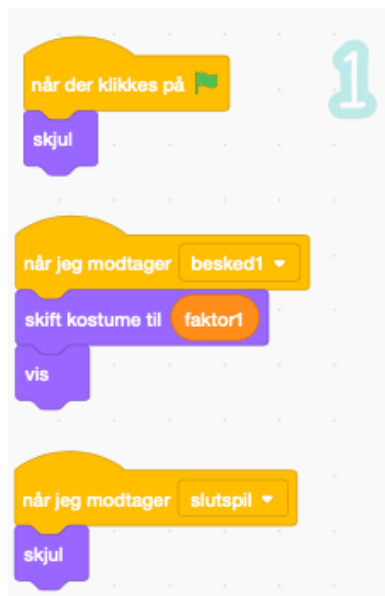
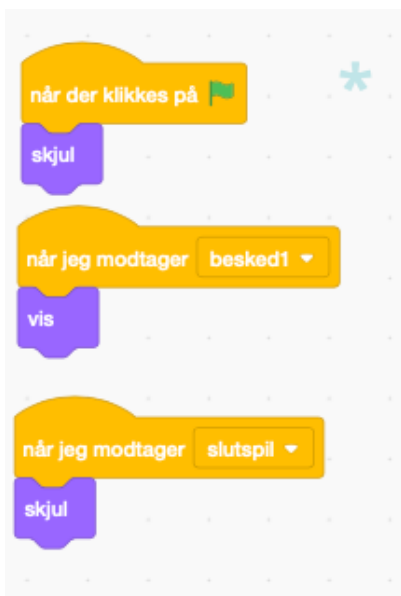


The image shows two Scratch scripts side-by-side. The left script is a 'when green flag clicked' event that initializes variables (Point to 0, faktor1 and faktor2 to 0, resultat to 0), waits 1 second, and enters a loop 'repeat until Tid = 0'. Inside the loop, it sets faktor1 and faktor2 to random numbers between 1 and 9, calculates resultat = faktor1 * faktor2, and sends a message 'besked1'. It then asks 'Hvad er...?' and waits. If the answer is correct, it increases Point by 1 and plays 'Tada' sound. Otherwise, it plays 'Oops' sound. The right script is another 'when green flag clicked' event that hides variables Highscore and Point, sets Tid to 40, and enters a loop 'repeat until Tid = 0'. Inside the loop, it waits 1 second and decreases Tid by 1. After the loop, it sends a 'slutspil' message, displays 'Spillet er ovre - sejt gået! Din score er:', shows variables Point and Highscore, waits 1 second, and checks if Point > Highscore. If true, it plays 'Cheer' sound, says 'Tillykke - du har sat ny rekord!', and sets Highscore to Point. Finally, it waits 4 seconds and says 'Start en ny omgang ved at klikke på det grønne flag'.

Når timeren er på 0 sender du en 'slutspil' til dine sprites, og du kan nu vise dine point og highscoren. Du kan lave en sammenligning, om highscore er slået - og så kan du jo opdatere den med en ny highscore, hvis den er.

I den gamle blok retter du kun to ting: - du sætter samme gentag løkke ind, som ender, når Tid = 0. Altså - Gentag indtil Tid=0 og Starter med at sige "Klar, parat, start!"

Og hvad sker der så med vores sprites, når vi sender slutspil? Igen meget enkelt - vi slukker simpelthen bare for visning af alle tre sprites, ved at skjule dem, når der modtages besked 'slutspil':



Prøv spillet med fødderne! Skift placeringer – læg dem tæt på hinanden eller langt fra hinanden. Hvem regner hurtigst? Skal nogle af de andre klasser prøve jeres nye spil? God fornøjelse!

Har du lyst til at prøve at spille mit spil?

Version 1: <https://scratch.mit.edu/projects/663824082>

Version 2: <https://scratch.mit.edu/projects/587921533>