**Eksempel på princip #5 i et undervisningsforløb – en regnemaskine**

Eksemplet er en smule kompliceret, så det egner sig bedst til slutningen af mellemtrinnet eller overbygningen.

#### **Konkretisering:**

Start med at finde den færdige version af regnemaskinen på <https://scratch.mit.edu/projects/135976136/> og demonstrér, hvordan den virker. Lad eleverne tænke over, hvilke elementer der indgår, **uden** at de ser koden bag.

Svar:

* 2 forskellige baggrunde
* en instruktion på startbaggrunden
* 1 sprite med 4 forskellige udtryk
* talebobler
* tekstfelt til svar
* 6 forskellige muligheder for udregning

Lad dem derefter reflektere over, hvilke elementer, der er de allervigtigste for programmets funktionalitet.

Svar:

1: Programmets skal kunne fortage beregninger ud fra brugerens input; gerne plus, minus, gange, dividere, kvadratrod og afrunding, men mindre kan gøre det.

(2: Der skal helst være en sprite. Dette er ikke nødvendigt, da programmet kan genereres af baggrunden, men da Scratch altid starter op med katten, er det mest logisk at lade denne eller en selvvalgt sprite generere koden).

Lad nu eleverne eksperimentere med at udforme en lille kodesekvens, som får programmet til at lægge tal sammen. Du kan evt. vise dem de kommandoer, som indgår i figur 39.

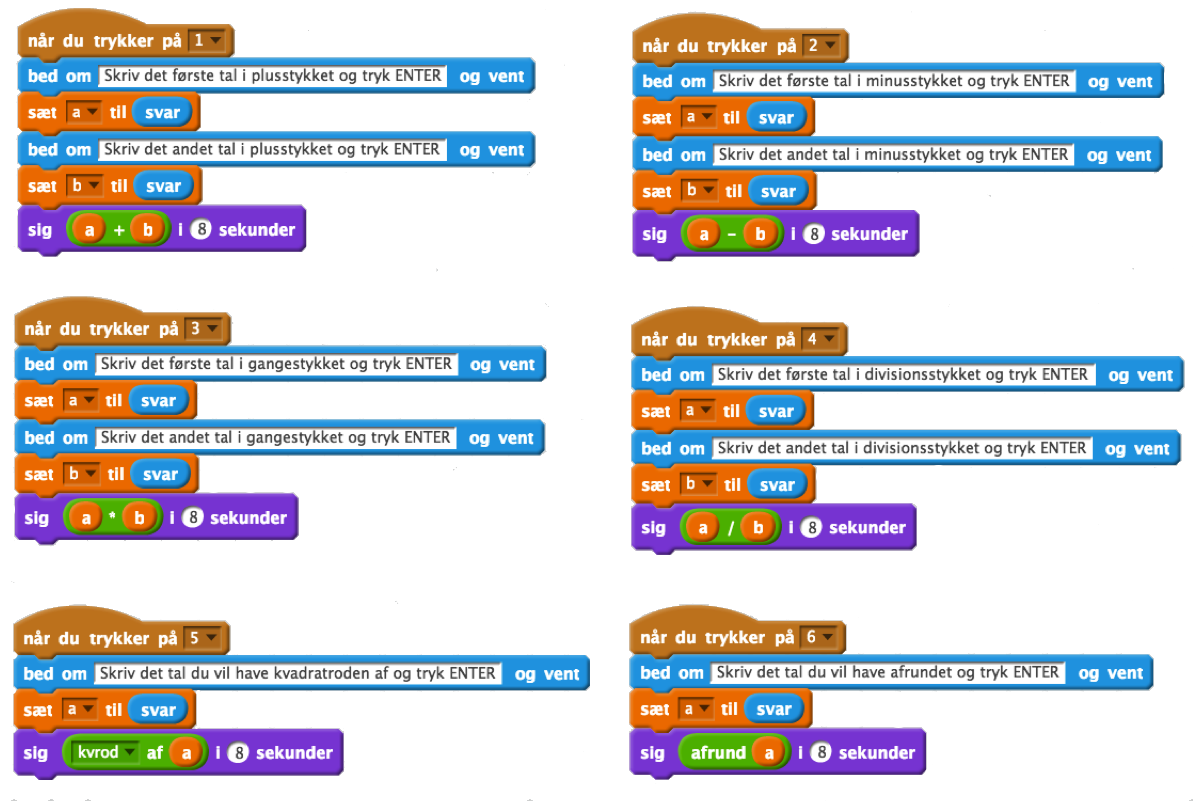


*Figur 39. Lille program i Scratch, som kan lægge to tal sammen*

Hvis ikke eleverne allerede har lært det, skal du også vise dem, hvordan de opretter variablerne a og b i den orange kategori, så programmet genererer de ekstra blokke, som de har brug for. Sørg for, at eleverne forstår, hvad det er, koden gør.

#### **Udvidelse:**

Eleverne udformer derefter lignende programmer for de øvrige regnefunktioner. Det kunne se således ud:

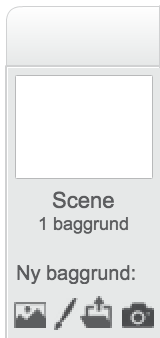


*Figur 40 viser første version af regnemaskinen med små programmer til plus, minus, gange, dele, kvadratrod og afrunding. Du kan finde denne version på:* [*https://scratch.mit.edu/projects/135970269/*](https://scratch.mit.edu/projects/135970269/)

Eleverne har nu en regnemaskine, som kan udføre de 6 typer udregninger, men der er ingen forklaring på, hvad man skal i programmet. Her i eksemplet skal man fx vælge mellem 1 og 6 for at programmet ved, hvilken type udregning, den skal lave, men det fremgår ingen steder.

Eleverne kan løse dette ved at lave en startbaggrund med instruktioner og en anden baggrund, som aktiveres, når der trykkes på en af de 6 taster, der aktiverer de små regneprogrammer. Her er der samtidig mulighed for at gøre spriten lidt mere levende ved at lade den skifte udseende samtidig, hvis de har valgt en sprite, som har flere kostumer. Flere kostumer betyder, at spriten findes med forskellige ansigtsudtryk og/eller positurer.

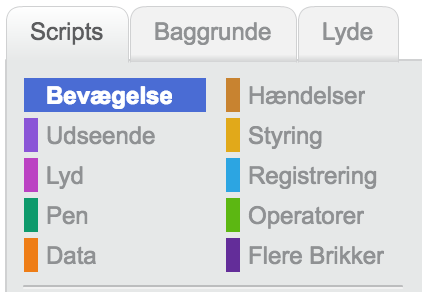
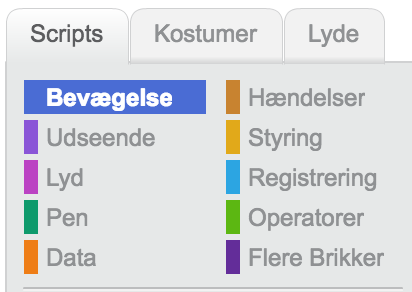
Eleverne skal starte med at udforme startbaggrunden. De kan fx vælge en fra biblioteket og så indsætte en instruerende tekst, eller de kan tegne en selv fra bunden, uploade et billede efter eget valg eller bruge computerens kamera til at tage et billede. De tilgår mulighederne ved at klikke på ikonerne nederst i venstre hjørne:



*Figur 41. Udsnit af brugerfladen i Scratch, som findes i nederste venstre hjørne*

Når eleverne arbejder med baggrunde, skifter en af fanerne i midterpanelet fra ”kostumer” til ”Baggrunde” automatisk. Grafikeditoren åbnes ved at klikke på denne fane. Nu kan baggrunden redigeres, og de kan fx indsætte en instruktion til regnemaskinen.

Grafikeditoren åbner også, hvis der klikkes på ”kostumer”, men så er det spriten og ikke baggrunden, der kan redigeres. Det kan eleverne fx bruge, hvis de har valgt en sprite, som ikke i forvejen har flere kostumer.



*Figur 42. Udsnit af midterpanelet i Scratch, som viser ændring i den midterste fane*

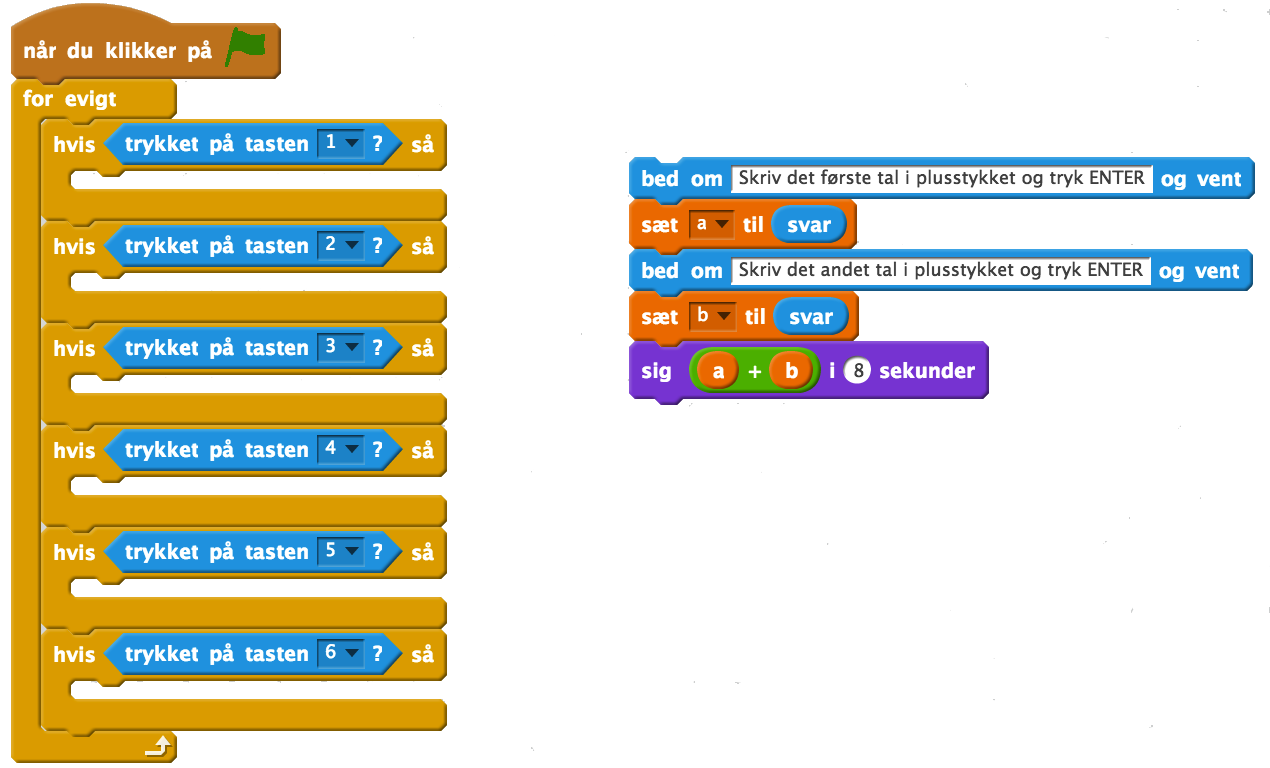
Når eleverne har udformet deres startbaggrund og måske også redigeret den baggrund, som der skal være, mens regnemaskinen er aktiveret, skal de lave de små programmer, som kan styre skiftet mellem baggrunde og evt. også spritens udseende. De små nye programmer kan fx se således ud:



*Figur 43. Små programmer, som tilføjer skift af baggrund og spritens udseende*

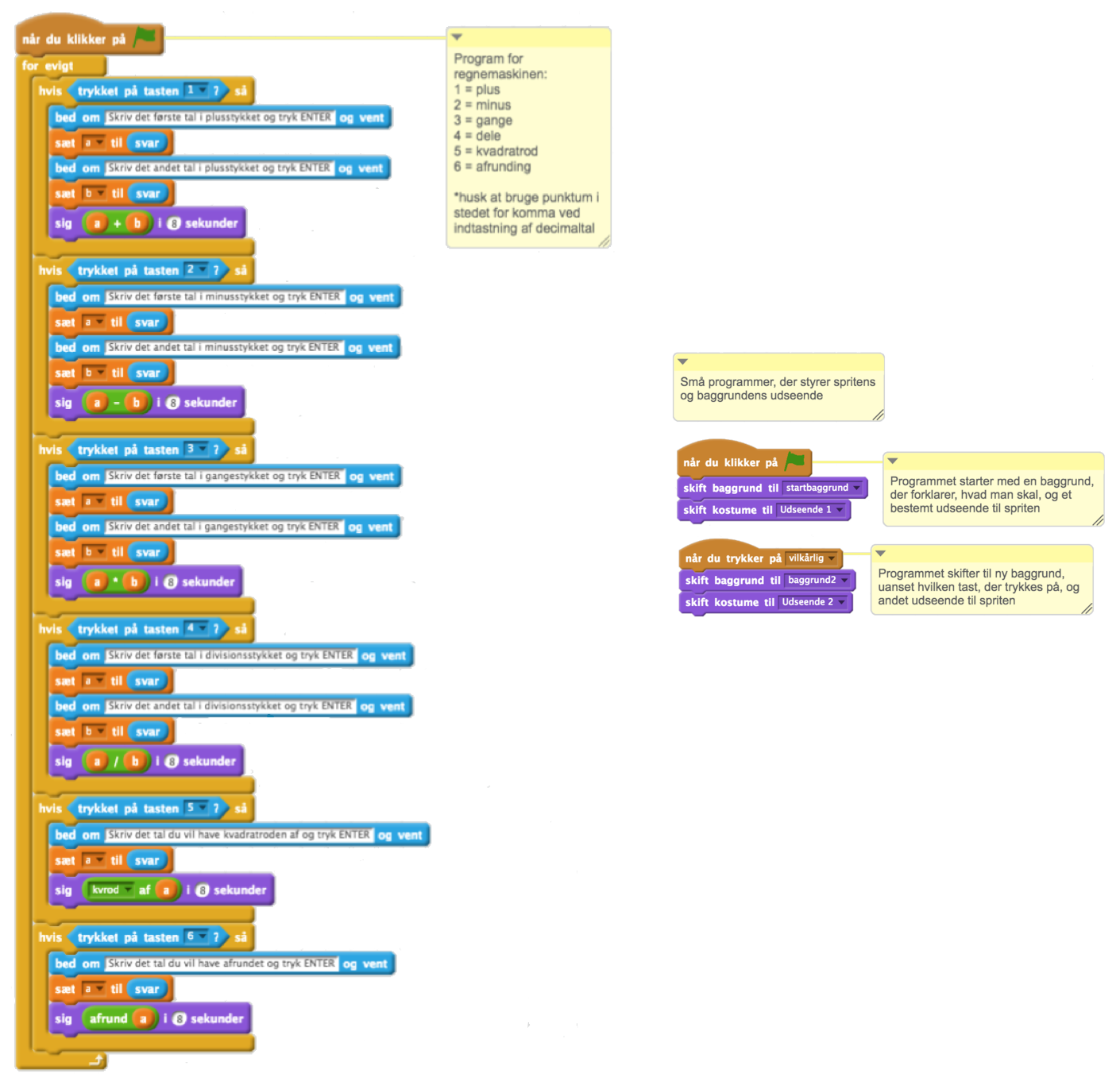
#### **Omstrukturering:**

Med udvidelsen af de 2 nye små programmer er editoren lidt uoverskuelig. Det kan være svært hurtigt at se, hvilken del der gør hvad. Derfor bør eleverne nu forsøge at sætte nogle af de forskellige programmer sammen til et samlet program:



*Figur 44. Starten på et samlet regneprogram*

I figur 44 er der påbegyndt et program, som kan samle de seks regneprogrammer. Programmet omkranses af en for-evigt-løkke, da de kommandoer, der er inde i denne, skal afvikles, **hver gang** de definerede betingelser er opfyldt, i dette eksempel tastetryk på enten 1, 2, 3, 4, 5 eller 6. Uden for-evigt-løkken vil de blot blive afviklet første gang, betingelserne opfyldes. Til højre ses kodesekvensen for additionsprogrammet. Denne skal indsættes i den første hvis-løkke. De resterende regneprogrammer indsættes på samme måde i de øvrige hvis-løkker. Så ser programmet således ud:

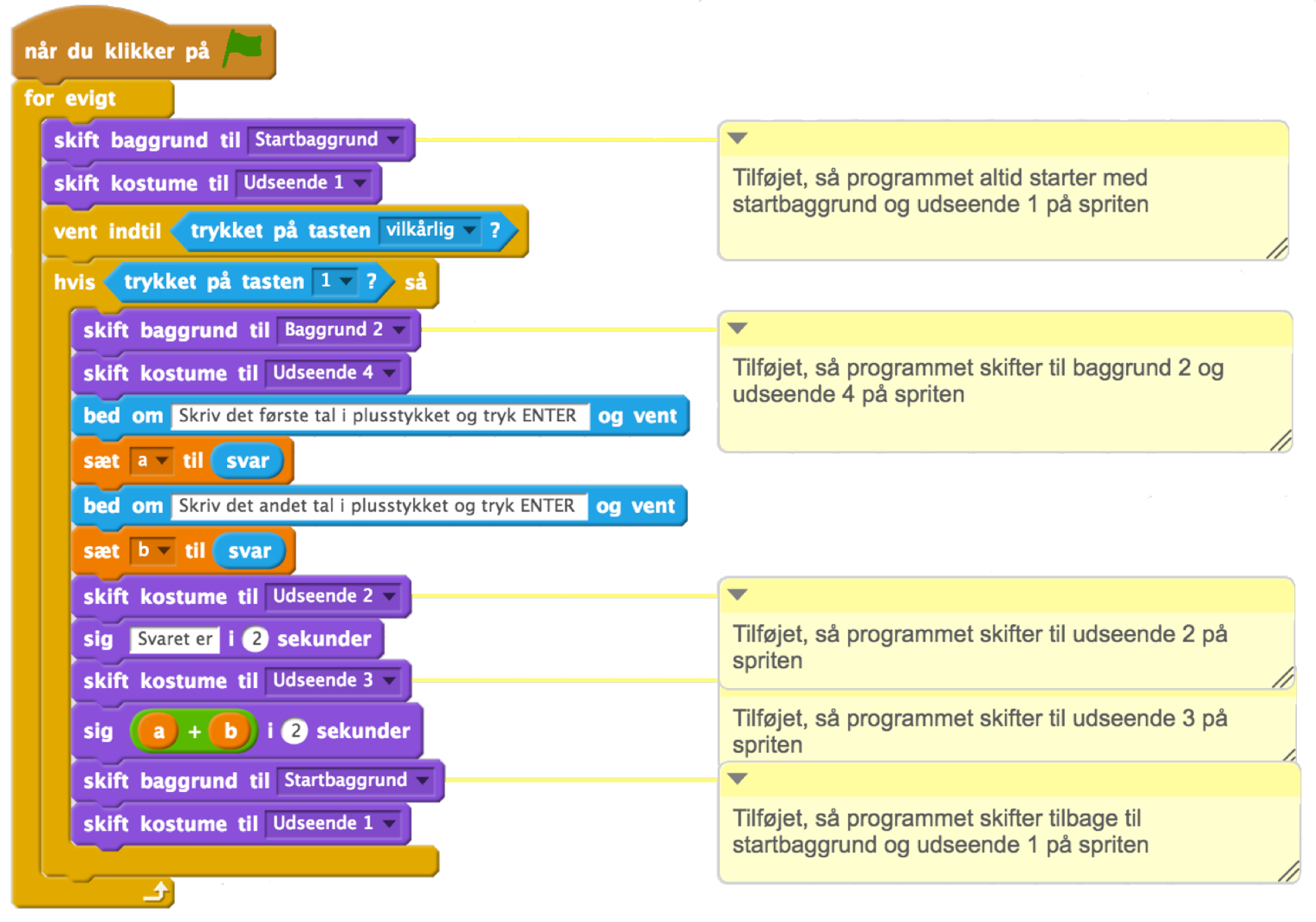


*Figur 45 viser anden version af regnemaskinen, hvor de små regneprogrammer er sat sammen til et samlet program. Du kan finde denne version på:* [*https://scratch.mit.edu/projects/135961698/*](https://scratch.mit.edu/projects/135961698/)

I Scratch kan blokke skilles ad ved at markere dem med musecursoren og trække dem nedad. De kan også kopieres ved at højreklikke og vælge ”kopier”. De overskydende blokke, som ikke længere skal bruges, kan fjernes ved enten at trække dem over i midterpanelet, hvor de så forsvinder, eller ved at højreklikke og vælge ”slet”.

#### **Ny udvidelse og omstrukturering:**

Til den sprite, vores eksempel bruger, findes der 4 forskellige kostumer, dvs. vi har altså mulighed for at lade den skifte mellem 4 forskellige udtryk for at gøre den endnu mere levende. I figur 46 er programmet endnu engang omstruktureret for at få plads til denne udvidelse. Samtidig er de små programmer flyttet ind i den lange kodesekvens, således at programmet altid starter med startbaggrunden med instruktionen og et bestemt kostume (udseende 1), skifter baggrund og kostume (udseende 4) ved tryk på tast, skifter kostume igen, når spriten ”tænker” (udseende 2) og når den svarer (udseende 3). Til sidst skifter både baggrund og kostume tilbage til startscenen, så programmet er klar til at udføre en ny udregning.



*Figur 46. Et udsnit af programkoden for den endelige version af regnemaskinen, som du finder på:* [*https://scratch.mit.edu/projects/135976136/*](https://scratch.mit.edu/projects/135976136/)

I figur 46 er der kun vist starten af omstruktureringen. Først er det lille program, som fik regnemaskinen til altid at starte med startbaggrunden og udseende 1, samt en kommando om at vente, til der trykkes på en vilkårlig tast, sat ind foran hvis-løkken, som omkranser additions-programmet. Inde i dette program er de øvrige kommandoer, som styrer skift af baggrund og udseende indsat.

Derudover er det antal sekunder, som spriten udsiger svaret på regnestykket i, ændret fra 8 sekunder til 2 sekunder, da det faktisk er nok til at registrere svaret.

De øvrige fem regneprogrammer skal stadig komme i forlængelse af additionsprogrammet, som du så i figur 45. I hver af disse indsættes samme tilføjelser og ændringer som i additionsprogrammet. Hvis du åbner den færdige version af regnemaskinen på <https://scratch.mit.edu/projects/135976136/>, kan du se kodesekvensen i sin fulde længde.

Der er mange andre muligheder for udvidelser. Fx kan eleverne vælge at sætte lydeffekter ind undervejs eller tilføje bevægelser og andre effekter. De kan også lave forskellige baggrunde til hver af de seks former for udregning. Hver gang, eleverne tilføjer nye elementer, er princip #5 i spil: De konkretiserer en abstrakt ide, udvider programmet og omstrukturerer løbende for at gøre programmet så simpelt og logisk som muligt. Fx er det en overvejelse værd, om programmet efterhånden er blevet så langt, at det faktisk vil være smartere at opdele det i mindre programmer igen som i første version af regnemaskinen, eller om der måske er nogle mønstre på tværs af de 6 programmer til udregning, som med fordel kan samles i en enkelt kodesekvens, som aktiveres hver gang, et af udregningsprogrammerne skal bruge den.