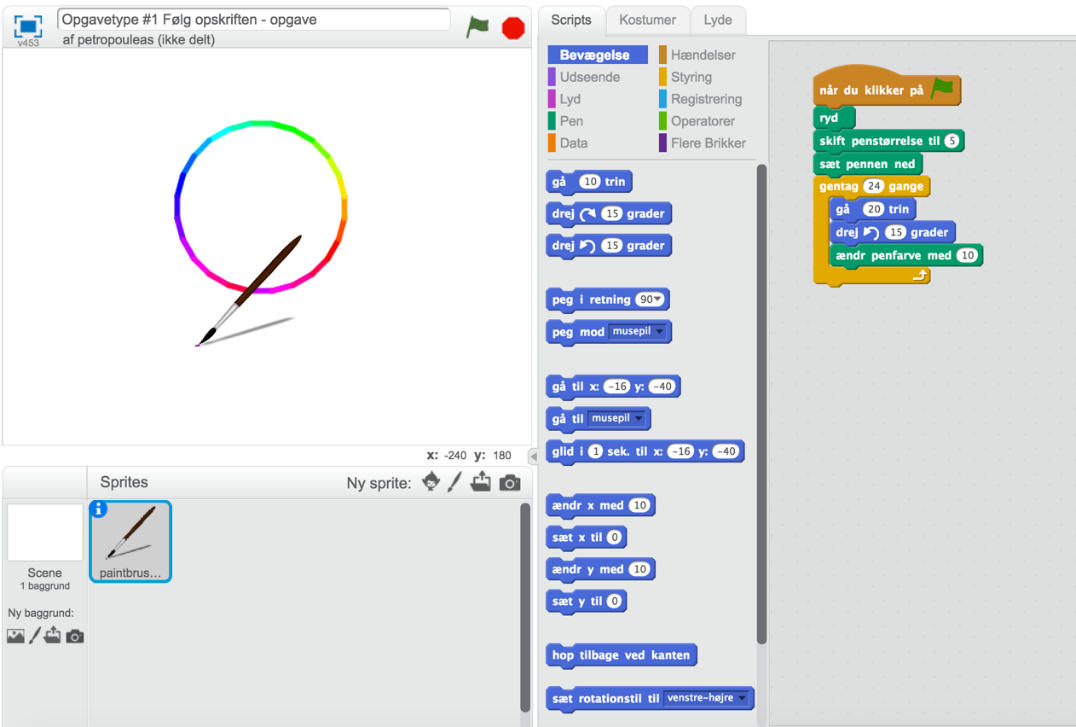
**Opgavetype #1: Følg opskriften**

Elev-handout

Start med at åbne Scratch, og skift katten ud med en sprite efter eget valg. Find de forskellige kodeblokke, og sæt sammen som vist på billedet, så dit program tegner cirklen. Prøv at forklare programmet – hvad gør de enkelte blokke?





**Opgave**

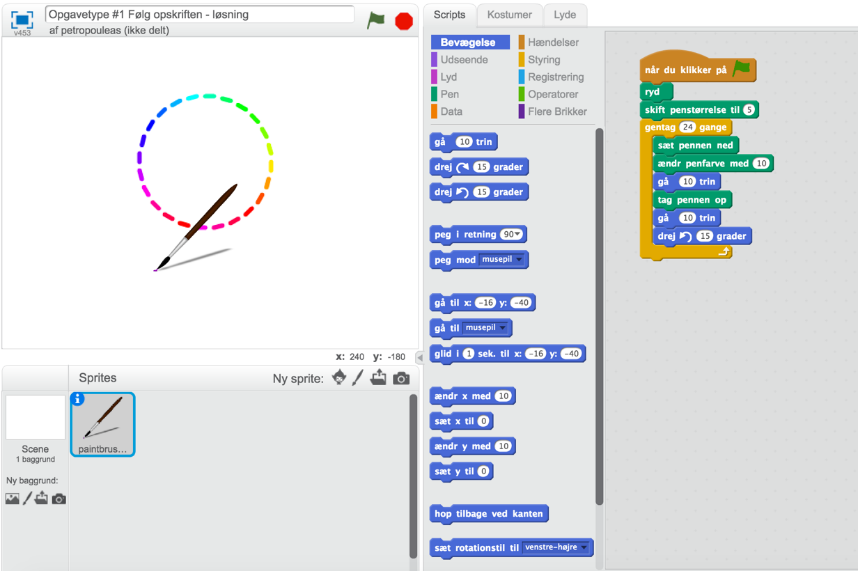
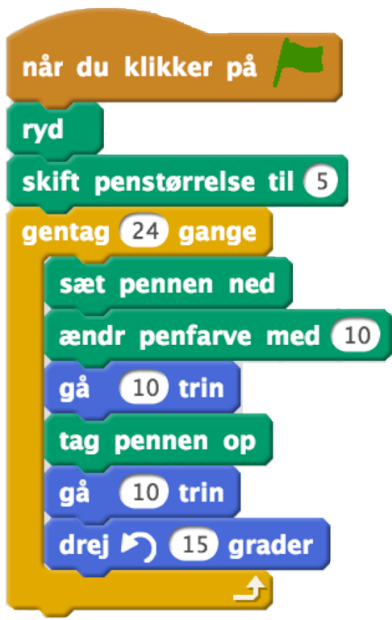
Kan du udvide og ændre koden, så cirklen i stedet tegnes med en stiplet linje, som du ser her? Prøv dig frem, indtil du har fundet løsningen

**Udvidelsesopgave**

Kan du lave et nyt program, som tegner en anden geometrisk figur?

**Til underviseren**

Opgavetype #1: Følg opskriften

**Løsning**

**Hvad eleven bør vide på forhånd**

Eleven skal kunne åbne Scratch og udskifte sprites, samt vide, hvordan de forskellige kodeblokke er organiseret, hvordan de sættes sammen og fjernes igen. De skal også vide, at de selv kan ændre på tallene i de hvide cirkler, som indgår i nogle af kodeblokkene, og hvor det grønne flag, som aktiverer programmet, er. De behøver ikke kende penne-værktøjet forinden, da de gennem løsning af den konkrete opgave automatisk vil opdage, hvordan det bruges.

**Beskrivelse af opgavetype #1: ”Følg opskriften”**

Eleverne får et færdigt program, som de skal kopiere sætning for sætning. Altså et program, der virker. Lad eleverne prøve at regne ud, hvad programmet gør, før det kopieres og afprøves. Derefter stilles eleverne en udfordringsopgave, hvor de skal modificere koden: “Kan du ændre i koden, så…?”

Til slut kan forløbet suppleres med en udvidelsesopgave, hvor eleverne skal lave et lignende program fra bunden. En meget brugt øvelse i Scratch er at tegne geometriske former. Du kan finde eksempler på Scratch.mit.edu. Hvis det første program, eleverne præsenteres for, tegner et kvadrat, kan du eksempelvis bede eleverne om at lave et lignende program, der tegner et rektangel, en trekant eller en cirkel.

***Didaktiske principper:*** *#2: Der tages udgangspunkt i et program, der virker. #3: Programmet afprøves, analyseres og modificeres, inden der konstrueres et lignende program*

***Computational Thinking:*** *logisk ræsonnement, da eleverne først skal forudsige, hvad programmet gør, algoritmisk tænkning og eksperimentering gennem udfordringsopgaven samt abstraktion og udformning i den sidste del, hvor de skal lave et lignende program.*

***SOLO-taksonomi:*** *bevægelse fra præstrukturelt til relationelt niveau.*