# Lektion 3 (ca 60min)

**Syfte och lärandeobjekt**

Syftet med lektionen är att vidareutveckla elevers algoritmiska tänkande i programmering, genom att låta elever skapa egen kod och lösa problem i en öppen uppgift.

**Material**

Skapa studio i klassrummet som heter Lektion 3-4

Ta fram exempel på spel gjorda i scratch[[1]](#footnote-1)

Extra kopior av elevinstruktionen till Lektion 2

**Introduktion ”Vad är ett spel?” (15min)**

Vad är det som gör ett program till ett matematikspel snarare än en miniräknare? Testa två olika och diskutera tävlingsmomentet och att det är människan (användaren) som ska utmanas, inte programmet.

Visa programmen på de två följande länkarna. Är de exempel på matematikspel? Varför/varför inte? Låt eleverna diskutera.

<https://scratch.mit.edu/projects/264303123/#fullscreen>

<https://scratch.mit.edu/projects/264306258/#fullscreen>

I följande länk finns några spel som kan användas som exempel efter diskussionen på hur ett matematikspel skapat i scratch skulle kunna se ut1

Påminn eleverna om deras uppdrag som introducerades i Lektion 1 ”Ni ska nu göra ett MATEMATIKSPEL, som kan användas för att träna matematik.” Påminn även eleverna om stegen då de behöver hjälp (stega igenom koden osv)

**Aktivitet**

*Vad?*

Eleverna ska utveckla sin förståelse för hur man kan skapa egen kod och lösa problem i den processen.

*Hur?*

Eleverna skapar egen kod enligt en egen idé, eller enligt den stegvisa instruktionen i Lektion 2. Om en elev inte varit med i lektion 2, eller som inte kommit så långt, uppmuntras denne att fortsätta med instruktionen från lektion 2, annars uppmuntras de att börja från början med en egen idé.

* Om eleven väljer att fortsätta sitt spel från lektion 2, glöm inte att lägga till det i studion Lektion 3-4
* Om eleven väljer att göra ett helt nytt spel, då **ska** spelet delas och sedan läggas i studion Lektion 3-4 (finns i instruktionen till lektion 2)

*Varför?*

En viktig idé i programmering som del i matematikämnet är att det ska bygga på den enskildes kreativitet och problemlösningsstrategier. I det kreativa arbetet med att göra ett spel finns möjligheter att både utveckla förståelse för programmering såväl som för matematiska begrepp som variabel och slumphändelse.

Någon gång under lektionen, när det passar:

*Vad?*

Eleverna ska utveckla sin förståelse för hur variabler kan användas på olika sätt i en kod.

Hur?

Be en elev att visa hur hen har hanterat variabler. Välj ut en elev som t.ex. använder en variabel för att räkna poäng, eller för att benämna termerna i en räkneoperation, eller för att spara slumptal som används i spelets mekanik. Utnyttja klassrummet i scratch så att alla får tillgång till det utvalda projektet. Stega med fingret och förklara hur det fungerar.

Varför?

Variabelbegreppet har visats vara problematiskt i både matematik och programmering, genom att ge praktiska och konkreta exempel på hur de kan användas får eleverna en möjlighet att utveckla sin förståelse för dem även på ett generellt plan.

**Avslutning**

Om man vill ha en avslutningsaktivitet på lektionen så är det lämpligt att be eleverna att utforska klassrummet. Finns det redan något projekt som är spelbart? Vad har andra för idéer? Var kan man finna inspiration för att lösa ett problem man själv stött på i nästa lektion?

1. <https://scratch.mit.edu/studios/5566034/> [↑](#footnote-ref-1)