

# TDDD78 projekt: Grafiskt programmeringspel

## 1 Introduktion

Det finns en kategori av spel som går ut på att spelaren programmerar beteendet för sin karaktär (eller lag). Ett exempel är Code Combat (<https://codecombat.com>) där spelaren är en äventyrare som ska programmera sin hjälte genom olika banor. Det finns även tappningar där spelaren ska programmera en robot och strida mot andra spelares programmerade robotar. Till exempel RoboWar från 1992. Kombinerar man den här typen av spel med grafisk programmering får man det här projektet. Målet är att spelaren har en bana som den ska ta sig igenom. För att göra det behöver spelaren programmera sin karaktär med hjälp av ett grafiskt programmeringsgränssnitt.

## 2 Bakgrund

Se till följande länkar:

<https://codecombat.com>

<https://en.wikipedia.org/wiki/RoboWar>

[https://en.wikipedia.org/wiki/Visual\\_programming\\_language](https://en.wikipedia.org/wiki/Visual_programming_language)

## 3 Milstolpar

Spelet har två olika delar som båda behövs för att man ska kunna spela det. Den ena hur man programmerar sin karaktär och den andra är visualisering av banan och när spelarens kod körs. De mest grundläggande delarna är beskrivna i noderna fram till och med noden **Spelbart** i figur 1. Resterande noder i figuren beskriver olika utökning som man kan göra.

## 4 Milstolpe beskrivning

Följande är mer detaljerade beskrivningar av milstolparna.

- **Första banan:** Den första saken som måste göras är den första banan som spelaren ska ta sig igenom. För det här spelet rekommenderas det starkt att man låter banan vara ett rutnät som spelaren rör sig igenom.
- **Programyta:** Parallellt med att spelaren ser banan som hen ska få sin karaktär att gå igenom behöver hen ha ett utrymme att skapa sitt program på. Det här milstolpen syftar till att skapa ytan där programmet skapas i. Utöver det ska det även finnas en startbox som spelaren senare kan använda för att koppla den första boxen i programmet till.

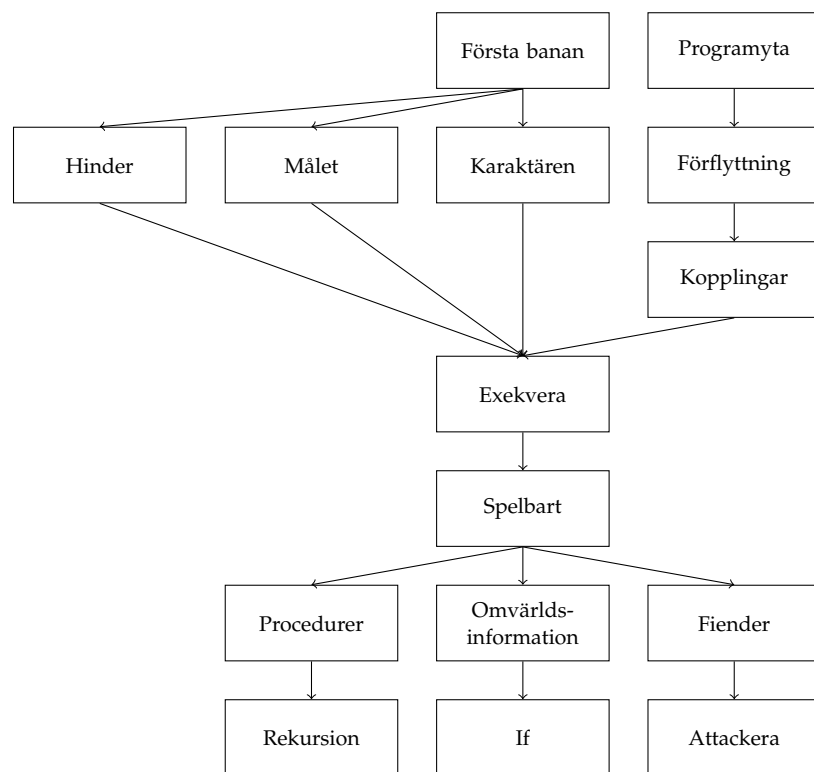


Fig. 1: Milstolparna för ett grafiskt programmerings spel.

- **Hinder:** En bana som enbart består av ett rutnät utan några begränsningar utöver kanterna på rutnätet är inte speciellt intressant. Om man istället lägger till hinder, till exempel väggar, blir det en större utmaning för den som programmerar.
- **Målet:** En bana behöver ett mål som karaktären ska nå. Den här milstolpen syftar till att målet ska implementeras och visualiseras på banan.
- **Karaktären:** I den här milstolpen ska spelarens karaktär skapas. Det innebär att den ska finnas med i modellen men även synas på banan.
- **Förflyttning:** Det här är de första handlingarna som spelaren ska kunna använda för att skapa sitt program. De är enkla handlingar som får karaktären att röra sig.
- **Kopplingar:** Har man bara en massa block placerade på en yta säger det inte mycket om hur programmet fungerar. Låter man spelaren koppla ihop olika block med varandra är det plötsligt möjligt för hen att skapa ett helt beteende.
- **Exekvera:** När man kommer hit har man alla byggstenar färdiga för att kunna köra spelarens program i banan. I den här milstolpen går ut på att exekvera programmet som spelaren har skapat i den aktuella banan.
- **Spelbart:** När ni har kommit hit bör spelet vara fungerande i dess mista form. Testa ert program lite extra för att se till så att allt fungerar bra och är buggfritt.

- **Procedurer:** Som ni själva förmodligen har märkt är det bra att kunna skapa funktioner och metoder som utför delar av en uppgift så att de kan användas till att lösa mer komplexa uppgifter. I den här milstolpen ska ni ge spelaren möjligheten att skapa procedurer vilket kan liknas med en funktion eller metod. De procedurer som spelaren har skapat ska sedan kunna användas i programmet de har skapat.
- **Omvärldsinformation:** Spelet blir betydligt mer intressant om spelaren har möjlighet att läsa av hur omvärlden ser ut runt omkring dess karaktär. Är det en vägg till vänster? Höger? Det är även nödvändigt att den här möjligheten finns för spelaren om hen ska kunna använda sig av meningsfulla if-block.
- **Fiender:** Har vi ett äventyrsspel kan vi naturligtvis även ha med fiender som är ute efter spelarens karaktär. De kan stå still och anfalla karaktären om den kommer i närheten eller patrullera runt på banan. Deras exakta beteende är helt upp till er.
- **Rekursion:** Rekursion är ett av de mest grundläggande begreppen när det kommer till programmering. Den här milstolpen syftar till att ni ska ordna så att spelarens procedurer kan vara rekursiva.
- **If:** När man har fått information om omvärlden är det intressant för spelaren att kunna agera utifrån den. I den här milstolpen är tanken att man implementerar ett if-block för spelaren. Ni får själva bestämma hur det ska se ut men tänk på att det bör vara meningsfullt för spelaren att använda det.
- **Attackera:** När fiender har tillkommit är det möjligt att ge spelaren möjligheten att slå tillbaka mot fienderna. Den här milstolpen syftar till att ge spelaren ett attack-block som hen kan använda för att attackera fienderna.

## 5 Designbeslut att tänka på

Det finns inga speciella designbeslut som har uppmärksammats ännu.