

# TDDD78 projekt: Brädspel

## 1 Varning

Digitala versioner av bräd- och kort-spel är något som har varit ett vanligt förekommande projekt i kursen under åren som har gått. Det finns ohöört många olika typer av brädspel och kortspel vilket gör det näst intill omöjligt att skriva en generell projektspecifikation. På grund av det är den här specifikationen en av de med minst innehåll vilket betyder att ni själva kommer behöva fylla ut den för att passa just ert projekt. Det påverkar dock inte hur mycket som förväntas av er.

Ytterligare en sak som har observerats under årens gång är att bräd- och kort-spel har en tendens att ofta bli mycket fokus på att skriva kod för varje möjlig pjäs. Ta till exempel schack, skriver man femtioelva if-satser för att hantera de olika pjäserna i alla situationer har man högst troligt misslyckats att visa på att man behärskar Java och objektorienterad programmering i den utsträckning som förväntas i projektet.

Hur intressant artificiell intelligens (härefter kallat AI) än må vara i diverse bräd- och kort-spel är det inte fokus i den här kursen. Därför är inte en imponerande AI, såvida den inte visar på era kunskaper i Java och objektorienterad programmering, vanligen vad man bör lägga fokus på i projektet. Med det sagt, AI kan vara ett ypperligt sätt att visa på ovanstående färdigheter men det är långt ifrån ett säkert kort.

Till slut, många spel (framförallt kortspel) har ohöört mycket innehåll i form av variationer (för kortspel olika kort). Det här tillhör vad man brukar kalla "content" på engelska och är en essentiell del i säljbara spel. Det är dock i det flesta spelen inte speciellt intressant för den här kursen. Se därför till att börja med att plocka fram essensen i spelet och applicera därefter Ockhams rakkniv på essensen för att skala bort så mycket som möjligt. Det ni har kvar är förhoppningsvis det som är intressant för kursen i Java och objektorienterad programmering. Tar vi till exempel Magic the Gathering behöver man enbart ett par kort för att testa så att allt fungerar: Ett land-kort, ett monster-kort som har en tappningseffekt, ett instant-kort, ett sorcery-kort och ett enchantement-kort. Det är inte ens en uttömmande lista på alla korttyper men det är tillräckligt för att se att själva spelmekaniken fungerar utan att ni har spenderat för mycket tid på alla detaljer. Har ni tid och känner att ni har lyckats visa på era färdigheter kan ni därefter utöka spelet med mer innehåll.

## 2 Introduktion

Implementationer av bräd- och kort-spel har funnits väldigt länge. Till exempel kom digitala versioner av schack under 70-talet. Sedan dess har det kommit fler och fler implementationer av spelen som introducerar ny alternativ. Att spela över nätverk var en tidig utveckling (så fort Internet uppstod det vill säga) men även AI-motståndarna utvecklades snabbt för spelen<sup>1</sup>. På senare tid har kommit fler och fler spel på mark-

---

<sup>1</sup> En av anledningarna till det här är att många brädspel, likt schack och go, är komplexa. Det gör dem utmanade problem för forskare och företag för att driva utvecklingen framåt.

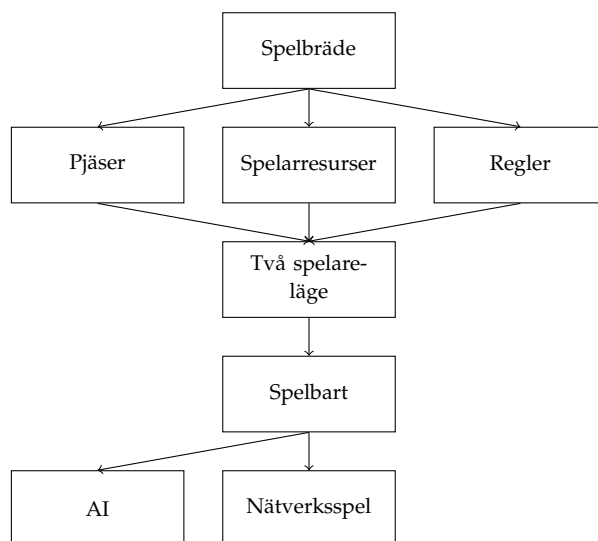


Fig. 1: Milstolparna för ett bräd- eller kort-spel.

naden som är implementationer av bräd- och kort-spel. Det är inte längre enbart en digitalisering av analoga spel som sker utan man skapar helt nya spel på samma form fast i datorn. Några exempel på kända bräd och kort-spel som har gjorts till datorspel är schack, go, Magic the Gathering och Ticket to Ride. Ett par exempel på spel som har gjorts för datorer utan en fysisk version är Hearthstone och Amarello.

### 3 Bakgrund

Se följande wikipedia-artikel:

[https://en.wikipedia.org/wiki/Board\\_game#Board\\_games\\_and\\_other\\_media](https://en.wikipedia.org/wiki/Board_game#Board_games_and_other_media)

### 4 Milstolpar

Milstolparna som presenteras för det här projektet är ganska diffusa eftersom varje spel kan skilja sig ohört mycket från varandra. En följd av det är att det enbart finns ett fåtal milstolpar men många av dem är istället ganska komplexa.

### 5 Milstolpe beskrivning

- **Spelbräde:** Nästan alla spel har ett spelbräde. I brädspel tar det oftast formen av ett bräde som pjäser kan placeras på. Kortspelens bräde brukar vara olika designade platser för kort. Beroende på kortens placering har de olika betydelser. Till exempel kan det vara en "kasthö" (en. discard pile) och ett resursfält.
- **Pjäser:** I de flesta spelen finns det även pjäser av ett eller flera slag. I kortspel är det oftast korten som är "pjäserna" på spelplanen.

- **Spelarresurser:** I många bräd- och kort-spel finns det olika mängder resurser som spelaren använder sig av för att utföra sina drag.
- **Regler:** Här kommer en av de största delarna i spelet och även en av de större fallgroparna: Hur man spelar spelet. Se till att tänka igenom spelets struktur och tänk på hur ni skulle skriva detaljerade instruktioner för varje tur innan ni implementerar spelet. Läs därefter varningssektionen igen och se till att ni skalar ner spelet så att bara grunden i spelet finns kvar. Har ni gjort en bra implementation kan ni utöka det i ett senare skede. Här är även risken att man fastnar när man skriver regler för hur olika pjäser fungerar, försök komma på enkla generella algoritmer (inte nödvändigtvis effektiva) för att lösa problemet istället för en massa if-satser. Till exempel kan vi lösa en springares placering i schack med följande pseudokod:

```
moves ← { (±1, ±3), (±3, ±1) }  
moves.filter((x, y) → {  
  if this.myX + x < 0 || this.myX + x ≥ board.getWidth() ||  
    this.myY + y < 0 || this.myY + y ≥ board.getHeight() then  
    return false  
  else if !board.isFree(this.myX + x, this.myY + y) &&  
    board.getPiece(this.myX + x, this.myY + y).getOwner() ==  
    this.getOwner() then  
    return false  
  else  
    return true  
  end if  
});  
return moves.map((x, y) → (this.myX + x, this.myY + y))
```

- **Två spelar-läge:** Ett tvåspelar läge är ett simpelt enkelt sätt att få ett spel (som ska ha flera spelare) att bli spelbart snabbt. Man behöver varken AI eller något nätverk utan låter helt enkelt spelarna att byta plats framför datorn.
- **Spelbart:** Vid det här laget bör spelet vara i ett någorlunda spelbart skick. Testa spelet lite mer noggrant och se till att alla buggar är lösta.
- **AI:** Den här milstolpen kan bli väldigt stor eller väldigt liten beroende på vilken spel man gör och hur komplex man vill göra sin AI. Tänk på att komplexiteten i AI inte är fokus för den här kursen.
- **Nätverksspel:** Att kunna spela över ett nätverk gör det betydligt lättare för många typer av spel (där de andra spelarna inte får se till exempel ens hand i ett kortspel). Liket AI är nätverk inte något som är ett av kursens mål så utöver de programmeringsfärdigheter och förståelse för objektorientering koden visar på ger det inget extra (för examinationen).

## **6 Designbeslut att tänka på**

Se sektion 1.