

---

# PROGRAMUTVECKLINGSMETODIK

---

## Laboration 3



DEN 12 DECEMBER 2017

KARLSTADS UNIVERSITET

## Innehållsförteckning

1. Inledning.....	2
2. Analys .....	2
2.1 Specifikationer.....	2
2.2 Klasser, egenskaper och förmågor .....	3
2.2.1 Mainklass.....	3
2.2.2 Mainpanel.....	3
2.2.3 Fönstret .....	3
2.2.4 Bollen.....	3
2.2.5 Brickor .....	3
2.2.6 Slagträet .....	3
3. Design .....	4
3.1 Färgdesign .....	4
3.2 Designen för slagträet .....	4
3.3 Designen för bollen .....	4
3.4 Designen för brickorna .....	4
4. Implementering & testning .....	5
5. Problem .....	5
5. Sammanfattning.....	5

# 1. Inledning

I denna laboration ska man skapa ett program som efterliknar spelet Breakout. Spelet går ut på att man ska i princip styra ett slagträ för att slå iväg en boll, som i sin tur ska träffa ett antal brickor. Spelet ska bestå utav en spelplan (GameBoard), där den övre delen av spelplanen ska brickorna utplaceras. Brickorna ska även bestå utav flera olika färger. Runt spelplanen ska det finnas tre väggar för att bollen ska kunna studsas på dem. Vid spelplanens botten ska det inte finnas någon vägg, eftersom när bollen åker ner så ska ett av liven förbrukas. Ett slagträ ska finnas längst ner på spelplanen som bollen kan studsas på, därtill ska slagträet styras med hjälp av tangentbord. Vissa brickor ska försvinna vid första träff, medan andra ska byta färg innan de försvinner. Brickorna som försvinner vid första träff ska ge en poäng, samtidigt som brickorna med unika egenskaper ska ge extra poäng. Det är spelarens jobb att försöka träffa bollen med slagträet. Ifall spelaren missar bollen och den hamnar nedanför slagträet, så kommer spelaren få en boll mindre (Ett liv förbrukas). När spelaren har förbrukat tre bollar, det vill säga tre liv, då har spelaren förlorat samt att spelarens resultat ska visas tydligt på skärmen. Å andra sidan om alla brickor försvinner då har spelaren vunnit och spelarens resultat ska också visas på skärmen.

## 2. Analys

### 2.1 Specifikationer

*För att specifikationerna för programmet ska uppnås måste dessa delar vara med:*

- Spelplanen ska ha tre kanter där bollen kan studsas mot dem.
- Spelplanen ska ha ett slagträ som styrs utav tangentbord och förflyttas endast i sidled.
- Spelplanen ska ha brickor som försvinner när bollen studsar på dem.
- Brickor i spelplanen ska ha olika färger.
- Det ska finnas brickor som tar flera olika träffar innan de försvinner, samt ska det även visas tydligt genom att byta färg på brickor.
- Brickor ska ge poäng när spelaren träffar bollen, vissa brickor ska ge mer poäng än andra.
- Bollen ska kunna studsas på slagträ och brickor.
- Spelarens nuvarande poäng ska synas i spelet.
- Om bollen hamnar nedanför slagträet, så ska en ny boll tilldelas till spelaren. När spelaren har förlorat alla bollar så ska spelet vara över.
- Spelaren har tre bollar (liv) på sig för att få så många poäng som möjligt.
- Spelet ska startas genom att trycka på space-knappen.
- Spelet ska vara spelbart.

## 2.2 Klasser, egenskaper och förmågor

*Genom att analysera kravspecifikationen så kunde jag komma fram till att dessa klasser behövdes:*

- En mainklass för att skapa fönstret (Frame).
- En klass för mainpanel (Spelplanen).
- En klass för fönstret.
- En klass för bollen.
- En klass för Brickor.
- En klass för slagträet.

### 2.2.1 Mainklass

Mainklass är en klass som endast har förmågan att skapa ett nytt "Frame" fönster och det är där programmet exekveras först.

### 2.2.2 Mainpanel

Mainpanel är en klass som beskriver spelplanen och har båda egenskaper och förmågor. Egenskaperna hos Mainpanel är bland annat att spelarens nuvarande poäng ska finnas på skärmen samt hur många liv spelaren har. Därtill har Mainpanel en egenskap som bestämmer hur snabbt bollen och slagträet ska förflyttas. För att allt ska synas på skärmen så har Mainpanel en "PaintComponent" funktion som är en förmåga hos klassen. Klassen har även förmågan att kunna kontrollera kollisionen mellan brickor, slagträet och bollen, dessutom har Mainpanel förmågan att kunna ta bort brickor, slagträet, bollen och kunna styra slagträet med tangentbordet. Klassen har också förmågan att minska spelarens liv med ett när bollen hamnar nedanför slagträet.

### 2.2.3 Fönstret

Syftet med klassen till fönstret är att framställa fönstrets egenskaper och visa spelplanen med hjälp av förmågan "Boll\_window". D.v.s. det är i fönstret som spelaren kan se spelplanen.

### 2.2.4 Bollen

Boll-klassen är klassen som beskriver egenskaperna och förmågorna hos bollen. Bollen är beskrivet som ett objekt, därför kan man använda den som en klass. Bollens egenskaper är att den har en storlek, färg, hastighet i x-led, hastighet i y-led och en viss position i x-led samt y-led. Förmågorna hos bollen är att den ska kunna förflyttas (ge en hastighet), försvinna, styras med tangentbordet som är några exempel.

### 2.2.5 Brickor

Klassen Brick, precis som klassen Boll, skapades för brickorna. Det betyder också att brickorna anses som ett objekt i programmet. Brickorna har egenskaper som till exempel storlek, position och färg. Medan förmågorna hos klassen brick är att kunna ta bort, ge dessa egenskaper, samt byta färg på brickorna.

### 2.2.6 Slagträet

Klassen för slagträet beskriver egenskaperna och förmågorna hos slagträet. Slagträet är beskrivet som ett objekt, därför kan man använda den som en klass. Slagträets egenskaper är att den har en storlek, en hastighet samt en position i x-led och y-led. Slagträet har förmågan att kunna försvinna, röra på sig, visas på skärmen och ge en hastighet för slagträet.

## 3. Design

### 3.1 Färgdesign

När jag skapade färgdesignen tog jag inspiration ifrån andra breakout-spel. När det kommer till färgdesignen så använde jag mina egna idéer helt och hållet. Jag bestämde mig för att använda en svart bakgrund, vilket funkar bra i kontrast med det flesta andra färger. Färgen för brickorna tog jag ett flertal olika färger, så att spelplanen inte skulle se så tråkig ut. Slagträets färg var bara ett spontant färgval utan någon tanke bakom sig.

### 3.2 Designen för slagträet

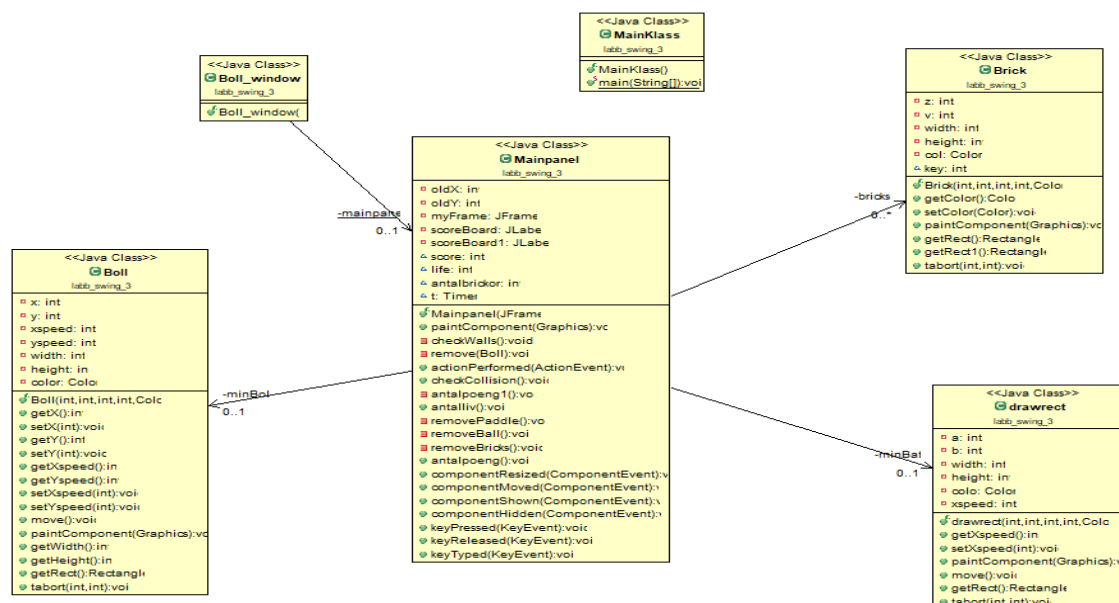
När jag skapade slagträet hade jag tagit några idéer ifrån nätet, saker som bredden och höjden. tanken var att bredden inte får vara för lång, då det kan bli för enkelt för spelaren, men bredden får inte vara för liten heller. Höjden på slagträet har inte så stor betydelse för svårighetsgraden, mest för synskull. Hastigheten var också en faktor som har med hur spelets svårighetsgrad ska avgöras. Jag tog en hastighet som passade för bollens hastighet.

### 3.3 Designen för bollen

Designen för bollen var snarlik med slagträet. Fast i detta fall skulle bollen färdas i två olika riktningar. Jag bestämde hastigheten för bollen med relation till hastigheten på slagträet, så att hastigheten passar så bra som möjligt. När jag bestämde bollens bredd och höjd tog jag hänsyn till mellanrummet emellan brickorna. Jag ville att spelaren skulle kunna lyckas få bollen emellan brickorna, men inte så att det blev för lätt.

### 3.4 Designen för brickorna

Brickornas design var tills mestadels inspirerad av breakout-projekt jag har sett på nätet. Brickornas uppdelning var fyra rader med åtta kolumner. Varje kolumn är uppdelad, så att det blir ett visst mellanrum. Jag ansåg att den designen såg snyggare ut än den med ihopsatta kolumner. För att jag skulle få plats med 8x4 nät fick jag lov att anpassa bredden och höjden på brickorna så att dom fick plats. *Här nedan kan du se UML-diagrammet:*



## 4. Implementering & testning

Tankegången när jag skapade klasserna var att utgå ifrån dom klasser jag visste skulle vara med. Dom klasser jag visste skulle vara med var baserat på det som stod i specifikationerna. Som första steg gjorde jag en mainklass. Sedan gjorde jag ett fönster där spelplanen låg, alltså mainpaneln. I spelplanen, lade jag sedan det olika objektklasserna som boll, slagträet och brickorna.

Strategin under projektets gång var att testa programmet kontinuerligt så att jag var säkra på att det jag gjorde var korrekt. Jag testade koden oftast vid tillfällen där jag hade gjort någon ändring eller när jag hade skapat något nytt.

## 5. Problem

En av dom svåraste problemen jag stötte på under projektets gång var delen där jag försökte ta bort brickorna vid en träff. Problemet var att brickorna inte kunde försvinna när bollen studsade emot dom. Jag kunde lösa problemet genom att använda en slags sorteringsteknik. Eftersom brickorna låg i olika positioner i en Array, så flyttade jag på platsen för just den brickan vid den specifika positionen. Det gjorde så att brickan kunde försvinna vid en träff.

Ett annat problem jag stötte på var att jag försökte få bollen att starta sin rörelse när man tryckte "space". Detta var ett av kraven för programmet. Jag lyckades lösa det igenom att flytta timern till mainpanel istället.

Ett av problemen var hur jag skulle hantera kollisionen, lösningen till detta var att lägga bollen och alla andra objekt i varsitt objekt av rektangulärform. Anledningen till det var så att jag kunde använda intersect för objekten. Vilket gjorde att jag kunde sätta dom i if-satser så att jag kunde kolla kollisionen.

## 5. Sammanfattning

I detta projekt har jag stött på många motgångar (buggar mm). Men vid dem flesta fall har jag lyckats tagit oss igenom dom problemen igenom att felsöka och lite tålamod. Jag lyckades uppfylla dom krav laborationen hade satt ut, och dom krav jag hade satt för oss själva när jag börja med projektet. Det var många nya saker jag stötte på under projektets gång. Under projektets gång har jag fått större förståelse för objektrelationer genom att jag har jobbat med fler klasser än innan. Jag har också lärt använda oss av ett nytt bibliotek som kallas Swing istället för Wheels. Delarna som har tagit längst tid var rörelsen av bollen och slagträet. Det tog oss ungefär två arbetsdagar var för det två delarna. Att få fram brickorna var också en av dom krångligare momenten. Det tog oss drygt en arbetsdag för att bli klar med. Kollisionen emellan bollen och brickorna blev jag klara med på en arbetsdag, samt räknaren för poäng och liv.