

Gui Obejekt orienterat

Objekt orienterat pong



den 24 mars 2017

Emil.H / peter

# Abstract

Denna rapport innehåller beskrivning för hur vi skapade ett Pong som i slutändan funkar men som inte riktigt nådde den standard vi hade i början. Vi ville se hur man skapar ett objektorienterat lätt spel och i detta fall så valde vi pong eftersom det är ett av de enklaste spelen man kan skapa.  
Spelet blev i slutändan spelbart men fick inget poängsystem och plattorna funkar inte riktigt som vi vill att dom ska.

Innehåll

[Abstract 1](#_Toc482953694)

[Syfte 2](#_Toc482953695)

[Teori/bakgrund 2](#_Toc482953696)

[Lösningen 2](#_Toc482953697)

[Användarhandledning 2](#_Toc482953698)

[Javadoc 2](#_Toc482953699)

[Konceptdesign 2](#_Toc482953700)

[Grund ide 2](#_Toc482953701)

[Utveckling 2](#_Toc482953702)

[Avancerat nivå 2](#_Toc482953703)

[Spelplan 2](#_Toc482953704)

[Platta 2](#_Toc482953705)

[Boll 2](#_Toc482953706)

[Points 2](#_Toc482953707)

[Klassdiagram 2](#_Toc482953708)

[Systembeskrivning 3](#_Toc482953709)

[Spelplan 3](#_Toc482953710)

[Spel 3](#_Toc482953711)

[World 3](#_Toc482953712)

[Platta 3](#_Toc482953713)

[Boll 3](#_Toc482953714)

[Testkörningar 4](#_Toc482953715)

[Diskussion 4](#_Toc482953716)

# Syfte

Syftet med detta projekt är att skapa ett objekt orienterat grafiskt gränssnitt som är interaktivt i detta fall skapade vi ett spel (pong). Alltså hur skapar man ett objektorienterad Pong?

# Teori/bakgrund

Vi ville se om man kan skapa ett simpelt pong där man då också använder sig av objekt orienterad programmering och där vi också använder oss av tiles för att rita upp spelplanen. Vi valde just pong eftersom att vi hade idén till att skapa ett enkelt 2d spel och vi ville skapa ett spel som är väldigt simpelt.

# Lösningen

## Användarhandledning

Programmet är enkelt att enkelt att använda. Två spelare kör samtidigt. Spelare 1 använder sig av W och S för att röra sig upp och ned och spelare 2 använder sig av O och L.

# Javadoc

## Konceptdesign

Detta var de tankar och idéer vi hade innan vi började skapa spelet.

### Grund ide

Pong

Två plattor rör sig upp och ned. En boll studsar fram och tillbaka om bollen skulle ta sig förbi en platta så får den plattan på andra sidan ett poäng.

### Utveckling

Plattorna och bollen får olika storlek och olika hastighet varje gång man startar spelet

### Avancerat nivå

Man kan plocka upp saker som ändrar på hur spelet spelas. Till exempel bollen blir tre gånger så stor eller plattorna rör sig långsammare.

### Spelplan

drawRuta(); skapa rutan

Canvas(); skapar plats att rita ut alla saker på.

spelPlan(); Ritar ut spelplanen

DrawPlatta(); Ritar ut plattan på spelplanen

DrawBoll(); Ritar ut bollen på spelplanen

DrawPoints(); Ritar ut poängsystemet på spelplanen

### Platta

Move(); Det som rör plattan upp och ned

### Boll

Move(); Det som bestämmer bollens rörelse

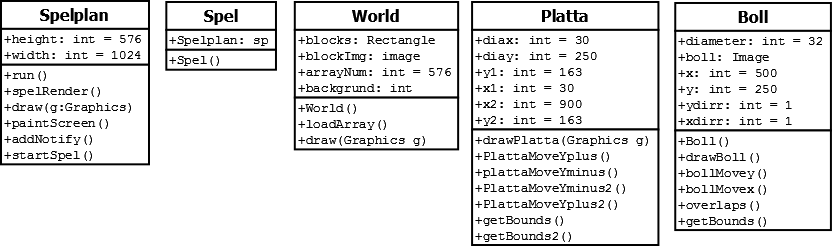
Collide y(); Det som bestämmer vad bollen ska göra när den träffar botten eller toppen av spelplanen

Collide x(); Det som bestämmer vad bollen ska göra då den träffar en an plattorna.

### Points

addPoint(); Det som räknar ut när ett poäng ska läggas till och för vilken spelare det ska läggas till för.

## Klassdiagram



## Systembeskrivning

# Spelplan

* run(): Bestämmer vad som ska göras då programmet startar.
* spelRender(): Renderar det som ska vara på spelplanen.
* draw(Graphics g): Hämtar grafiken från de olika klasserna.
* paintScreen(): Ritar ut allt som ska visas på skärmen.
* addNotify(): Visar programmet när det ska starta.
* startSpel(): Startar spelet.

# Spel

* Spel(): Hämtar mallen för spelplanen och ritar upp den på skärmen.

# World

* World(): Visar vilket nummer som representerar vilken tile.
* loadArray(): Visar hur alla olika tiles ska ritas ut.
* draw(Graphics g): ritar ut alla tiles.

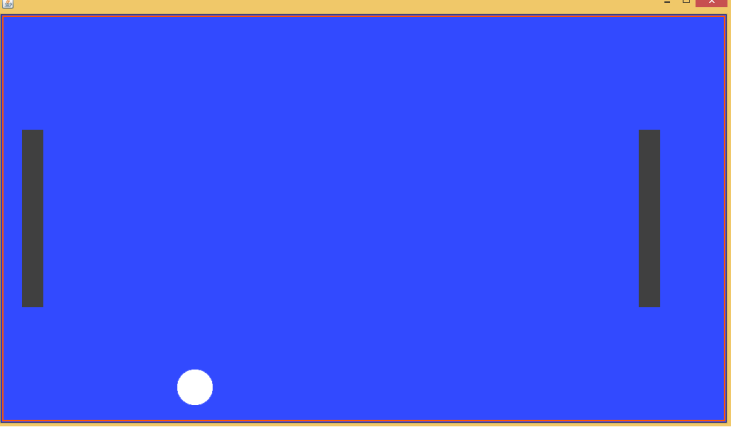
# Platta

* drawPlatta(): Ritar ut båda plattorna på skärmen.
* plattaMoveYplus(): Visar platta Y hur snabbt uppåt den ska röra sig.
* plattaMoveYminus(): Visar platta Y hur snabbt nedåt den ska röra sig.
* plattaMoveYminus2(): Visar platta Y2 hur snabbt nedåt den ska röra sig.
* plattaMoveYplus2(): Visar platta Y2 hur snabbt uppåt den ska röra sig.
* getBounds(): Skapar en rektangel runt platta Y som visar var den befinner sig.
* getBounds2(): Skapar en rektangel runt plattaY2 som visar var den befinner sig.

# Boll

* drawBoll(Graphics g): Ritar ut bollen på spelplanen.
* bollMovey(): Bestämmer hur bollen ska röra sig i y led och när den ska studsa mot botten och toppen av spelplanen.
* bollMovex(): Bestämmer hur bollen ska röra sig i x led
* overlaps(): Kollar om bollen överlappar någon av plattorna.
* getBounds(): Gör en rektangel runt bollen så man vet vart den befinner sig i x och y led.

# Testkörningar



Färdiga pong spelet.

# Diskussion

Vårat Pong blev väldigt simpelt allting funkar men det som gjordes bäst var nog hur spelplanen ritas upp i tiles. Själva plattorna och bollen är inte så bra. Plattorna funkar inte alltid då man startar spelet och bollen studsar bara i en vinkel. Men annars så är det ett fungerande Pong. Vi ville egentligen lägga till ett poäng system med det hände inte då vi inte hade tid nog att göra det. Och jag tror att vi i början hade lite högre mål än vad vi skapade i slutändan. Den konceptdesign vi skapade innan programmet hade 3 olika steg av svårighetsgrader för att skapa spelet. Vi hann inte med första svårighetsgraden som var att vi skulle skapa två plattor, en boll som studsar och ett poäng system.