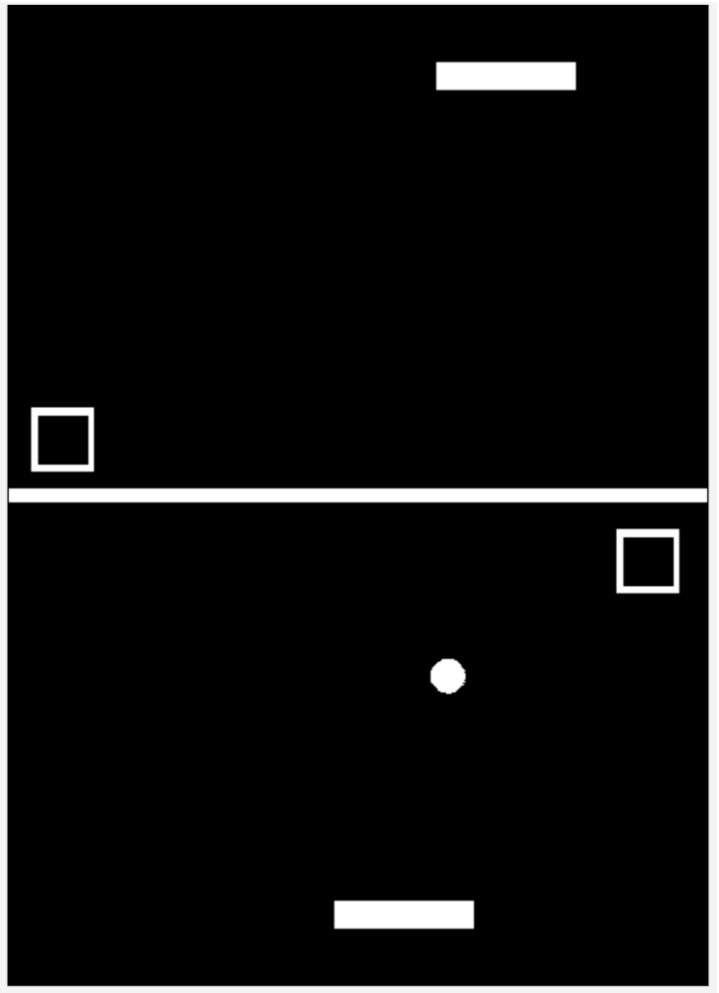
BUSINESS ACADEMY SOUTHWEST

# Ping - Compulsory Assignment #1



## Handed-in by

1. Marc
2. Maximilian
3. Christian
4. Nijas

# Date: 2/10-2018
# Table of Content

TABLE OF CONTENT ..... 2
1. INTRODUCTION ..... 3
2. FUNCTIONALITY ..... 3
3. INTERNAL DESIGN ..... 4
4. EVALUATION ..... 5

# 1. Introduction

I dette projekt vil vi efterligne det kendte ”Pong” spil som blev lanceret af Atari. Spillet går ud på at man har en bold, og to paddles. Man styrer selv den ene paddle og den anden bevæger sig af sig selv. Bolden skydes til en side og spillet begynder. Man skal ramme bolden med sin paddle og prøve at ramme forbi modstanderens paddle, derved score man et point.

# 2. Functionality

Da vi først fik udleveret spillet, var der kun en bold som spawnede og en paddle som bevægede sig fra side til side. Vi fik samtidig udleveret en række opgaver som skulle laves for at spillet kunne køre optimalt. Først opgave skulle vi lave en paddle som kunne bevæge sig når spilleren trykkede på nogle taster. Vi flyttede først den paddle som kunne bevæge sig af sig selv, dernest kopierede vi dens kode og slettede dens kode så den bare stod stille. Samtidig tilføjede vi koden ”isKeyDown” til både højre og venstre så vi kunne bevæge vores egen paddle, så der nu var to paddles både for oven og for neden.

Opgave 2 gik ud på at lave en paddle som kunne bevæge sig af sig selv, det havde vi allerede fra da vi fik udleveret opgaven. Vi tilføjede en kode så den paddle som bevægede sig af sig selv ville spawn på et nyt sted når den ramte væggen, dog ikke for tæt på spillerens paddle.

I opgave 3 skulle der tilføjes en kode som gjorde at bolden ville bounce på de paddles som vi havde tilføjet til spillet, dette blev gjort ved hjælp af koden checkPaddle og checkPlayer (Ball.class), derved kunne vi sige at hver gang bolden rørte en paddle ville den bevæge sig i en anden retning.

Opgave 4 blev konstrueret af Nijas i Gimp og lagt ind i mappen ”Images” i vores ping mappe. Dernæst blev baggrunden ændret til den baggrund, inde i pingworld klassen, ved hjælp af koden setBackground.

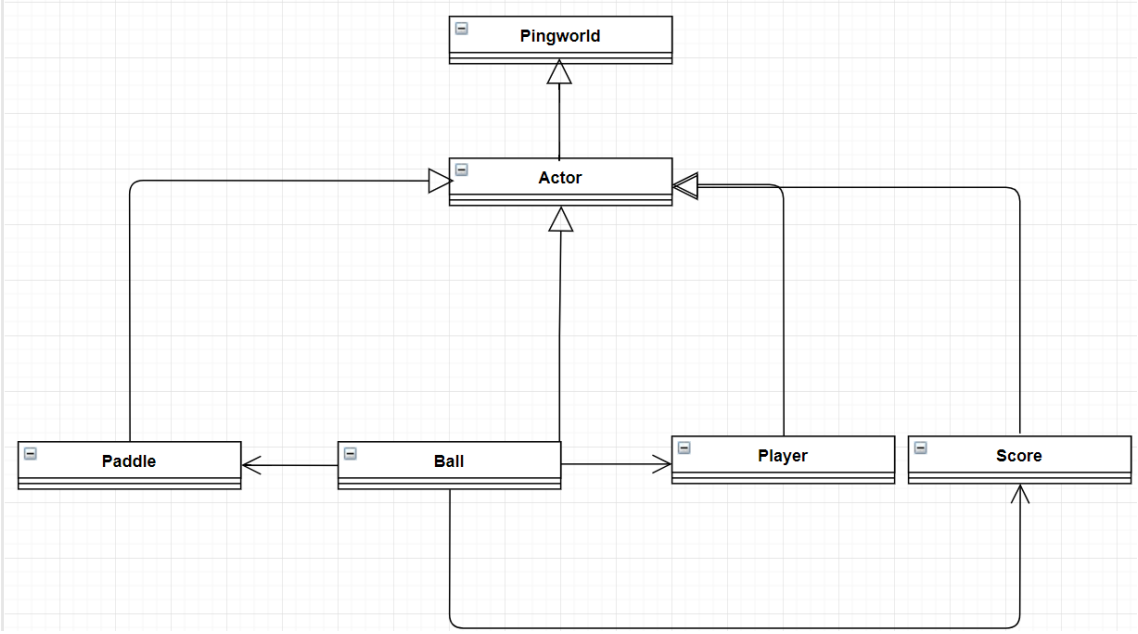
Opgave 5 blev fuldført ved hjælp af pong spillets originale lyde, som blev fundet på nettet. Opgaven gik ud på at tilføje lyde til bolden når den ramte vægge, paddles og når der scores. Lydene blev downloadet og sat ind i mappen sounds. Derefter blev koden skrevet ind i boldens class, når den fandt eller ”ramte” en væg. Koden hedder Greenfoot.playSound(”pongwallsound.wav”); hvis det var væggen som den skulle ramme.

Opgave 6 var en del sværere end vi regnede med, vi kunne ikke få tingene til at komme frem på vores kanvas, vi måtte få hjælp af en tutor, Kasper Siig, som fortalte os omkring heap og hvordan det fungerede, efter hans besøg kom tingene op på skærmen og vi fortsatte vores kodning.

Opgave 7, blev fuldført hvis bolden blev ramt af modstanderens paddle, så ville den bevæge sig hurtigere end før.

# 3. Internal design

Vores UML-diagram giver et blik over hvordan de forskellige classes benyttes.
Pingworld indeholder koordinatsættet til vores World, som vi kan benytte til flere forskellige objekter videre i projektet.
Paddle, Ball og Player er vores subclasses til Actor, hvorfra de arver funktionerne move(); og act. Disse classes er det visuelle vi ser på skærmen, henholdsvis bolden og de to paddles.
Paddlen har ingen funktioner afhængig af andre klasser. Den har arvet fra Actor, men køre derefter uafhængig af de andre. Det samme gør Player som er den class vi som spilleren kontrollere.
Vores Ball derimod har funktioner der associerer sig med både paddlen og playeren. Disse funktioner er nødvendige for at spillet vil kunne fungere. På både player og paddlen køres der en checkpaddle funktion der sørger for at bolden finder paddlen og kan få kontakt med den, i stedet for at gå igennem dem, bouncer den af dem.
Når vores ball går forbi de 2 paddles forsvinder den og respawner på dens sædvanlige position så spillet kan fortsætte. Når den gør dette er det fordi vores ball har scoret et point hos enten vores AI eller playeren. Dette point skal registreres og det gøres i vores score class. Derved forekommer der en association mellem vores ball og score class.



# 4. Evaluation

Projektet var et sjovt projekt, men greenfoot er ikke den bedste brugerplatform til at kode på, det hakkede og frys til tider. Trods det, var det en god og spændene opgave.

Vi har lært at arbejde med flere klasser på samme tid, vi har dermed fået mere forståelse for opbygning af programmer og software. Vi har lært at have overblik over tingene og lært hvordan samarbejde og kommunikation er en vigtig faktor for at kunne lave et godt program og for en arbejdsvenlig stemning.