Pong

Pong is het eerste spel ter wereld dat voor de computer uitkwam. De programmeurs Julian Gommers en ik, Sjoerd Furth, leek het leuk om dit spel opnieuw te maken. Alleen dan veel uitgebreider.

# Het doel

Het doel is om eerst een basis spel te maken en deze uit te breiden. Later in de opleiding als wij met Arduino’s gaan werken, dan is tevens ons doel om een controller te maken met de arduino’s om de batjes te besturen.

# Het begin

We zijn begonnen met het maken van een GIT repository. Hierin hebben we als eerste de basis code geladen met de library SLICK2D, een library in Java speciaal voor 2D spellen. Hierna is razend snel de basis gelegd voor het spel.

# Algoritmes

Het spel bestaat uit diverse algoritmes. Ik zal ze hieronder toelichten.

1. Hoeken van terug kaatsen; Op het moment dat de bal het batje van de speler raakt wordt deze in een bepaalde hoek terug geschoten. Hier zitten wiskundige berekeningen achter die dit berekenen. Iedere bounce gaat de bal ook steeds iets sneller.
2. Voorspellen waar de bal terecht komt; Op het moment dat deze optie is geactiveerd via de toets 8 of 9 wordt er weergegeven waar de bal ongeveer terecht gaat komen. Hier zitten diverse wiskundige formules achter die in de klasse InfoText staan gedefinieerd.
3. AI in de single player en insane modus; Er zit daadwerkelijk een vorm van Kunstmatige Intelligentie (AI) in dit programma. De AI zorgt in single modus Easy en gemiddeld dat hij het midden van zijn batje gelijk probeert te houden aan de Y coördinaat van de bal. In Expert is deze nog intelligenter en doet deze onder andere het volgende:
   1. Na iedere bounce zet hij het batje terug naar het midden om zo min mogelijk responsetijd te hebben om te reageren op de bal.
   2. De AI mikt de bal op het grootste gat van de andere speler.

# De klassen

De code is opgebouwd uit meerdere klassen. Deze zijn als volgt.

1. InfoText.java; Hierin staan alle tekstuele delen, zoals de menu’s, van het spel, tevens staat hier de code voor de voorspellingen.
2. Ball.java; Hierin staat de complete code voor de bewegingen van de bal.
3. NativeLoader.java; De library vereist het inladen van een aantal systeemuitbreidingen als een Audio Library. Deze worden ingelezen vanuit een JAR File die deze bestanden bevat. Door het gebruik van deze klasse is het uitvoeren van de code voor de ontwikkelaars en testers een stuk eenvoudiger geworden.
4. Scores.java; Hierin staat alle code met betrekking tot het bepalen van de scores.
5. Player.java; Hierin staat alle code die betrekking heeft met het tekenen, positionering en beweging van de speler batjes.
6. StartGame.java; Hierin staat alle basis code om het spel te starten. Hierin lopen onder andere twee loops te gelijk die de updates en renderen van het spel verzorgen. In deze code wordt ook de gamecontainer gemaakt en de toets gecombineerd met de actie.