

Verslag project

Pong – Computersystemen

El Yachouti Abdoullah, Boustani Daniel

23/12/2018

Inhoudstafel

1. Inleiding.....	2
2. Beschrijving van het project.....	2
3. Werking van het spel.....	3
4. Problemen en oplossingen.....	4

1. Inleiding

In deze document wordt onze project beschreven. Hierin leest u een beschrijving van onze project, de werking ervan, enkele ondervonden problemen en hun oplossingen, en uiteindelijk mogelijke verbeteringen.

2. Beschrijving van het project

Onze project is het beroemde spel *Pong*. Deze bestaat uit twee *source*-bestanden en één *include*-bestand.

De eerste *source*-bestand is het eigenlijke spel. Daarin worden alle procedures, die gedefinieerd zijn in het tweede *source* bestand, aangeroepen.

Naast het definiëren van procedures, bevat de tweede *source*-bestand alle globale variabelen van het spel.

In het spel, speelt de speler tegen de computer aan de hand van een palet. De eerste die aan een score van vijf geraakt is de winnaar.

3. Werking van het spel

Het spel begint bij de label *main*, waarin de gebruiker begroet wordt, en gevraagd wordt om op eender welke toets te drukken om het spel te starten. Na dit te hebben gedaan, zal het spel telkens bezig zitten in het *main-loop* van het spel totdat de speler het spel stopt, aan de hand van het *escape-toets*, of hij wint/verliest.

De *main-loop* begint door eerst elke spelelement (bal, paletten, scoren) te tekenen. Daarna wordt er gecheckt of men een toets heeft ingedrukt. Indien dat het geval is, zal dit eerst afgehandeld worden, vooraleer het palet van de computer en de bal ter beweging gebracht worden. Anders, kunnen de bal en het palet van de computer in beweging gebracht worden.

Het handelen van de ingedrukte toets gebeurt op een heel simpele manier: er wordt bekeken dat de waarde van *al* correspondeert tot de bovenste of onderste pijl, en er wordt gesprongen naar de gepaste label, waar de juiste procedure wordt opgeroepen.

De beweging van het palet van de computer is steeds hetzelfde: het palet beweegt steeds van beneden naar boven, en wisselt van richting wanneer het zich aan de bovenste of onderste grens bevindt. Ook, verandert het van richting wanneer de bal

van richting verandert, in andere worden volgt het palet van de computer ook de richting van de bal.

De bal beweegt, in een eerste tijd, van links naar rechts. Bij deze, wordt er steeds gecheckt of het tegen het palet van de speler/computer gebootst is, of het de linker/rechter limiet overschreden heeft, of geen van beide. In het eerste geval, verandert de bal van richting. In het tweede geval wordt er een punt aan de computer/speler gegeven en start het spel opnieuw. In het laatste geval beweegt de bal gewoon verder.

In een tweede tijd, beweegt de bal van beneden naar boven. Hier, wordt er steeds gecheckt dat de bal niet aan de boven of onderste grens geraakt is. Indien dat wel het geval is, verandert de bal van richting.

Er wordt uit de main-loop gesprongen wanneer, zoals al reeds gezegd, de speler de escape-toets ingedrukt of de computer/speler vijf punten heeft gescoord.

4. Problemen en oplossingen

Hierbij geven we een beschrijving van enkele problemen die we ondervonden hebben, en hun oplossingen.

Beweging van het palet van de computer: het probleem die we hierbij hadden is hoe we aan het palet moesten aangeven om van richting te veranderen. Bij het palet van de speler was dit vanzelfsprekend, het is de speler dat beslist, maar bij die van de computer was het moeilijker aangezien een computer niet zelfstandig kan denken.

De oplossing hiervoor was om een variabele te hebben (*hasToGoUp*) dat aangeeft in welke richting het palet moet gaan. Wanneer deze variabelen op één staat, betekent dit dat het palet naar omhoog moet, en wanneer die op nul staat moet het naar beneden. *hasToGoUp* verandert van waarde wanneer de palet aan de bovenste of onderste grens is aangerakt.

Beweging van de bal: het oplossen van het vorige probleem liet ons toe om deze probleem op een zeer gelijkaardige manier op te lossen. Hier ook werden twee variabelen gebruikt, *ballHasToGoUp* en *switchSide*, die, respectievelijk, aan de bal aangeven wanneer ze verticaal of horizontaal van richting moet veranderen. Deze variabelen veranderen van waarde wanneer de bal, respectievelijk, aan de bovenste/onderste grens en linker/rechter grens aangerakt is.

JLE en JGE (duplicatie): onze grootste vijand tijdens deze project was duplicatie, en voornamelijk subroutines die uit JLE en JGE bestonden. Sommige subroutines waren identiek hetzelfde, op één lijn na, namelijk een JLE-instructie/JGE-instructie. Om dit op te lossen nemen de betrokken procedures (bv. *ballMoves*) een argument *direction* dat aangeeft vanuit welke richting het betrokken spelelement komt. Afhankelijk van de richting, wordt er beslist of de JLE-instructie gebruikt wordt, of de JRE-instructie.

5. Mogelijke Verbeteringen

Alhoewel we, in het algemeen, blij zijn over onze project, zijn er enkele dingen die volgens ons beter geïmplementeerd konden zijn.

Een eerste verbetering is het traject van de bal. Na een bepaalde tijd beseft men dat de bal altijd hetzelfde traject volgt en is het dus gemakkelijk om te winnen. Dit zou opgelost kunnen worden door een procedure aan te maken dat een willekeurig getal genereerd, waarbij het getal gebruikt wordt als nieuwe snelheid van de bal. Ook zou dit opgelost kunnen worden door elke deel van een palet voor te stellen in een array en bij elke botsing te zien op welke deel van het palet de bal gebootst is, en daarmee te beslissen met welke snelheid de bal terug moet weerkaatsen.

Een tweede verbetering is het moeilijkheidsgraad van het spel. Het spel is niet zo moeilijk te spelen, aangezien de computer "dom" is. Dit zou opgelost kunnen worden door de snelheid van het palet van de computer aan te passen aan de snelheid van de bal zodat de computer sneller bij de bal geraakt. Een andere manier om het spel wat moeilijker te maken is om ervoor te zorgen dat het palet van de speler altijd kleiner wordt telkens hij scoort. Zo kunnen andere features bedacht worden om het spel wat moeilijker te maken.