

Singleplayer Snake spel

Av Evan Saboo (950115-4893) och Perttu Jääskeläinen (951216-9658)

2016-03-04

Objektiv och Krav.

Projektet gick ut på att göra en version av det klassiska spelet Snake. Spelet går ut på att man med hjälp av två knappar ska styra en orm som äter upp bitar på skärmen tills man kommer till en kant eller på sig själv vilket leder till förlust. Ormen kontrolleras med hjälp av två knappar som går åt motsatta håll när tryckt. Detta har vi lyckats med i projektet:

- Enkel snake spel där ormen blir större för varje gång han äter upp ”äpplet”.
- Poäng visare vars motiv är att visa antal gånger ormen har ätit upp äpplet.
- En switch som pausar spelet.
- Game over display när man förlorar.

Nedan är lista har vi inte lyckats med eftersom tiden var för kort:

- Det ska finnas olika nivåer där hastigheten är snabbare desto högre nivå man väljer (t.ex. Easy till Hard).
- Det ska finnas en topplista för högsta poäng nådda (en punkt = ett poäng).
- Ormen ska kontrolleras med knapparna på ChipKIT modulen.
- Ljud vid olika tillfällen, t.ex. vinst, förlust, nytt rekord samt i huvudmenyn när man t.ex. väljer bana eller hastighet.
- Olika varianter av spelbanan.

Lösning.

Detta projekt utvecklades med hjälp av ChipKIT Uno32 brädan tillsammans med Basic I/O skölden. Spelet visas i displayen på Basic I/O skölden. Både knapparna och växlar används för att navigera i programmet. All utveckling av programmet utfördes med programspråket C.

Bidrag till projektet.

Det blev annorlunda än vad vi hade planerat. Vi jobbade nästan alltid tillsammans där vi tog tur om vem som skulle skriva koden.

Verifikation. Programmet kördes alltid i ChipKit brädan för att kontrollera om programmet fungerade som det ska.

Reflektion. I början var det lite svårt för oss att hitta information på hur skärmen på I/O Basic Shield fungerade. Vi kollade sedan igenom några av kursen exemplar och fick hjälp i lunch office hour för att förstå hur skärmen fungerade. Sedan var det bara skapa logik för hur ormen ska röra sig och vart hans position var. Positionen av ormens huvud kontrollerades av en variabel som ändrades då ormen flyttade på sig. Vi använde oss av interrupt flag istället för en delay funktion för att bestämma ormens hastighet vilket gör att knapparna på I/O Skölden kan läsas av hela tiden utan någon fördröjning.

Om vi hade lite mer tid skulle vi ha kunnat implementera flera funktioner från planeringen.