

i2c Snake med online highscore

Av Jonathan Koitsalu, Ahmed Alhasani, Daniel Johansson, William Hansen

Table of Contents

i2c Snake med online highscore

Table of Contents	
Introduktion	
Idé	
System	
Implementation	
Hårdvara	
Arduino	
ESP32	
Knappar	
Display	
Övrigt	
Mjukvara och services	
MongoDB	
Heroku	
Node.js	
Arduino IDE	
Bibliotek	
Tidsestimering	
Generella tidsförhållanden	
Kronologisk tidsestimering	
Iteration 1:	55h + 10h extra marginal
Iteration 2:	60h + 10h extra marginal
Iteration 3:	45h + 10h extra marginal
Iteration 4:	20h
Iteration 5:	80h
Avgränsningar	

Introduktion

Första rapporten för projektarbetet i kursen "inbyggda system: arkitektur och design" under tritechkonsult Oscar Wahlbergs handledning.

Gruppens idé förklaras och presenteras i en systemskiss. En plan för implementation redogörs sedan med en närmare titt i verktygen, mjuk- och hårdvara, som kommer användas.

Projektets omfång tydliggörs i rapporten under punkten avgränsningar i kombination med tidsestimeringen av projektets deluppgifter.

Idé

Efter dagar av hjärnstormande runt exoskelett, voice-recognition system, och androider med självbevarelsedrift kom vi äntligen fram till en nyskapande idé:

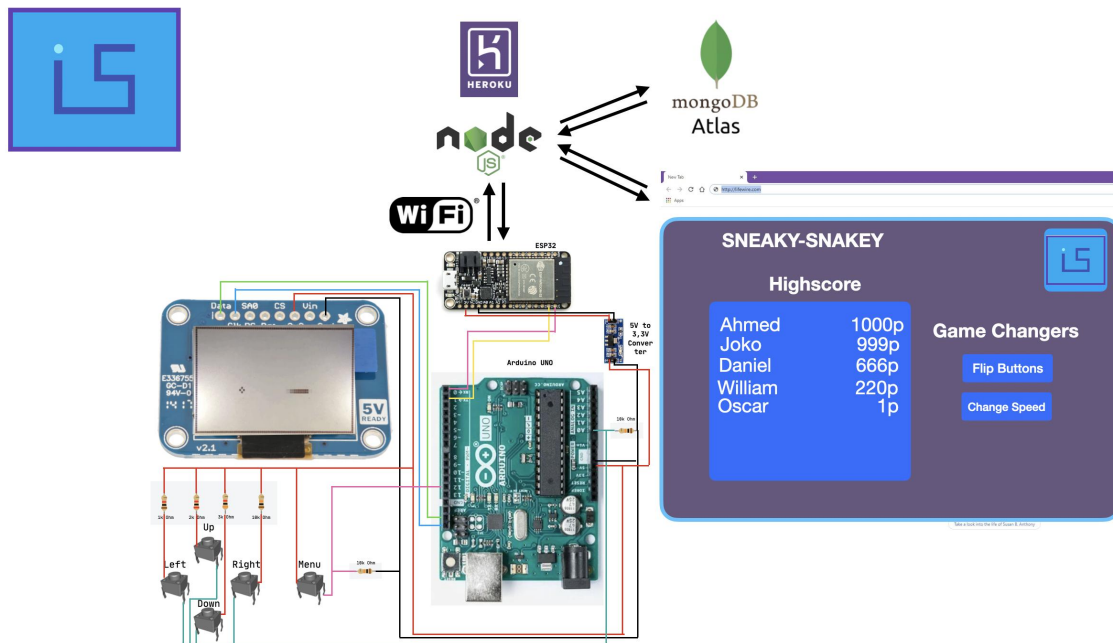
Snake.

Spelet ska köras lokalt på en minkrokontrollerenhet med mekaniska retrostil knappar. För maximal immersion presenteras grafiken på en liten display.

Spelare ska ha en arena att mäta sig mot i form av en online highscore-lista som uppdateras efter varje spelade match.

För att göra matcherna mer intressanta kommer spelparametrar såsom hastighet att kunna ändras från hemsidan.

System



Implementation

Hårdvara

- ESP32
- Arduino Uno
- OLED LCD Display
- Knappar
- Breadboard
- Kablar

Arduino

Arduinon är tänkt att fungera som navet i vår produkt, det är där spel-logik, skärm och knappar kommer hanteras. Fördelen med detta är att vi kan ta från vår tidigare erfarenhet att programmera mot Atmega chipet. Nackdelen är att vi behöver två kort då Arduinon inte har någon inbyggd Wi-Fi funktion.

ESP32

ESP32 är ett utvecklingskort med både Wi-Fi och Bluetooth egenskaper och flera digitala och analoga in/utgångar. Den kommer sköta Wi-Fi kommunikation till systemets webserver. Tanken är att Arduino:n skickar över data, med exempelvis UART, till ESP32 som skickar detta vidare via Wi-Fi till webserverns endpoint. Wi-Fi kommunikationslösningen kommer med två alternativ. Det ena är att använda bara Wi-Fi delen på ESP32 som en Wi-Fi shield det vill säga att den styrs helt av Arduinon. Risken med det är att det tar för lång tid att implementera då erfarenhet saknas. [Läs mer här.](#)

Det andra är att skriva kod som läser av UART meddelanden och sedan skickar dessa vidare med Wi-Fi. Fördelen med detta är att det går snabbt att implementera då både Arduinos library och Espressifs library finns tillgängligt för att hantera UART och Wi-Fi.

Knappar

Till Arduino:n planeras att koppla fem knappar som input, fyra av knapparna kommer att vara för styrning till själva spelet och en knapp för att komma till meny systemet. Planen är att skriva en state machine vilket innebär att styr knapparna kommer att kunna användas som styrning till menyn när man är i meny-state och styrning till spel när man är i spel-state.

Display

Spelet kommer att presenteras på en OLED LCD Display från Adafruit med en upplösning på 128x64. Det kan tolkas som ganska litet men upplösningen är bättre än en nokia 3310-diplay. En större skärm hade dessutom inneburit en större investering för medlemmarna i gruppen. En risk med detta är att spelet blir lite begränsat på grund av utrymmet på skärmen. Men vi tror att det kommer fungera bra som prototyp och vill man så går det att skala upp. Skärmen är monokrom (svartvit) vilket begränsar kreativiteten i spelet lite, men är inget hinder för att genomföra idén.

Övrigt

När det kommer till breadboard och kablar kommer vi ta det vi har hemma, dvs inga special grejer här. Eventuellt en Buck-converter 5V till 3,3V för strömsättning av ESP32 modulen.

Mjukvara och services

- Atlas MongoDB
- Heroku
- Node.js
- Arduino IDE (eventuellt)

MongoDB

För att spara användarens highscore är NoSQL databasen MongoDB en fördelaktig lösning då gruppmedlemmar har tidigare använt sig denna databas.

Heroku

Heroku kommer användas för att hosta vår webserver. Det är likt MongoDB en lösning gruppen har arbetat med tidigare vilket är anledning till detta val. Det är även ett vanligt verktyg att använda vid mindre projekt.

Node.js

Node.js kommer användas för att bygga webservern. Finns även här många alternativ att välja mellan men Node.js är ett familjärt verktyg för Daniel. Därmed en möjlighet för övriga gruppmedlemmar att få en inblick över bygget av webbservrar.

Arduino IDE

Kommer eventuellt att användas för att skriva kod till ESP32 för att hantera Wi-Fi kommunikation. Det beror på huruvida Wi-Fi chippet går direkt från Arduinon eller om vi väljer att skriva kod för ESP32an.

Bibliotek

- Snakekod.
- Arduino ESP32 library (eventuellt).
- AVR Dude
- AVR lib

Planen är att bygga egna libs för att hantera knappar, i2c kommunikation och UART till Arduinon. Beroende på hur komplicerat det är blir det antingen ett färdigt lib för själva skärmen eller ett eget lib.

Koden för Snake kommer vara inspirerad av tidigare implementationer av Snake då spelutveckling inte ingår i teamets expertis men den kan komma att behöva anpassas efter hård- och mjukvara. (Självklart källhänvisad där det behövs).

AVRDUDE kommer att användas för att flasha över koden till Arduinon.

AVR biblioteket kommer att användas för att få tillgång till register etc. till atmega chipet.

Arduino/Espressif Arduino biblioteken kommer eventuellt att användas för att skriva kod för ESP32 för att hantera Wi-Fi och mottagning via UART.

Tidsestimering

*Total tillgänglig tidsbudget:

8 timmar * 10 dagar * 4 personer = 320

ca 320h*

Estimerad total tidsåtgång för MVP: ca 170h exkl 30h extramarginal (iteration 1-3)

Estimerad total tidsåtgång för utökad funktionalitet: ca 120h (iteration 4-5)

Generella tidsförhållanden

Generellt vad som förväntas kräva mer tid än annat:

- I2C kommunikation mellan arduino och oled-skärm = 2x timmar
- Fullt funktionell knappstyrt snakespel på oled-skärm = 2x timmar
- databashantering på mongoDB via esp32 Wi-Fi för datalagring = x timmar
- utläsning av data på en heroku hostad hemsida = x timmar

Kronologisk tidsestimering

Iteration 1:

55h + 10h extra marginal

- Sätta upp Gemensam arbetsmiljö
 - Kanban board
 - Github repo
- Display library
- Koppla upp Arduino mot internet via ESP32
- I2C kommunikation från Arduino till en OLED skärm.
- Knappavläsning
- Koppla databas mot node.js

Iteration 2:

60h + 10h extra marginal

- Snake
- Statemachine
 - Menysystem
 - Highscore
 - Snake
 - (Options?)
- REST API för uppdatering av databas (node.js)

Iteration 3:

45h + 10h extra marginal

- Uart kommunikation från Arduino till Esp32
- Skicka score vid avslutat spel
- Ta emot score från REST API
- Highscore databas som uppdaterar hemsida

Iteration 4:

20h

- Kontrollera spelparametrar

Iteration 5:

80h

- Kontrollera spelet från hemsidan
- Multiplayer
- Större spelyta

Avgränsningar

- Under idéstadiet fanns ambitionen att köra all funktionalitet direkt på en Esp32:a. Efter research så beslutades det att använda en Arduino med ESP32 som wifi-brygga. Detta pga att utvecklingsmiljön kring Esp32 skiljer sig ganska mycket från AVR biblioteket och tidbegränsningen gör att gruppen inte hinner sätta sig in i det.
- En avgränsning kring utökad funktionalitet för multiplayer och direkt styrning från websida kommer förmodligen också behövas.