**Implementasjon av spillet:**

* Note 1 ble først laget på breadboard, og alle oppgaver gjennomført der. Etter at alt fungerte her, designet vi først skjematikk, la inn alle komponentene i Altium, og lagde en PCB. Denne fikk vi så frest ut hos elpro-labben på NTNU. Når vi lagde PCBen passet vi på å ha så få som mulig traces på oversiden, da disse er vanskelige å lodde, da komponentene står her, og vi ikke har dekkende viaer. Vi har heller ikke brukt noen viaer, da dette er mer jobb å bruke på en slik PCB. Alt av signaler går i bottom layer. Alt ble etter det loddet opp og testet, og med unntak av to mindre feil på kretskortet og endre pinout, fungerte alt uten problem.
* All tekst som vises på skjermen blir først skrevet til SRAM, og når alt er klart, så blir en oled\_repaint(); kjørt for å tegne data fra SRAM til skjermen. Bildet av katten som vises har grunnet plassmangel blitt skalert ned til 64x32, (skjermen er på 128x64), og blir ved hjelp av en skaleringsalgoritme skalert opp til full skjermstørrelse når bildet vises. Filmsnutten er i full 128x64 størrelse og hvert av bilene vises to ganger for å være en lengre sammenhengende filmsnutt. Alle bildene har blitt skalert, gjort om til monochrome? BMPs (bitmap) og så eksportert til arrays.
* Den grafiske representasjonen av spillet som vises på skjermen er basert på data som blir sendt fra Node 2 over CAN.
* Den kapasitive slideren på «usb-multifunction-card» brukes til å bestemme ønsket posisjon til solonoiden, og opereres ved å ta på ønsket punkt. Joysticken brukes til å navigere i menyen og høyre kapasitive knapp brukes til å skyte. Hvis man prøver å jukse ved å ikke skyte, så vil solonoiden skytes automatisk, slik at man mister ballen.
* Når spillet startes, beveger solonoiden seg sakte til venstre, helt til den slutter å bevege seg, og da kalibreres posisjonen.
* Spillet går ut på at man må prøve å holde ballen «i luften» så lenge som mulig, og når den «faller ned», så er runden over, og highscore blir notert. Beste highscore blir lagret på SRAMen. Signalet på IR-mottakeren er forholdsvis støyete, spesielt når solonoiden skyter, og det er derfor implementert et EMA (exponential moving average) filter for å filtrere ut det faktiske signalet. Score regnes i desisekunder.
* Node 1 kommuniserer med SRAM, skjermen og ADS ved hjelp av latchen og GALen, mens den kommuniserer serielt med RS-232 transcieveren, og over SPI med CAN-kontrolleren.
* Node 2 bruker I2C til å styre DACen for å styre motoren, mens den leser ut enkoder-posisjon direkte over en 8-bits bus???, og noen kontrollsignaler. Node to har også kontrollsignaler til solonoiden, som styres av elektronikk plassert på kretskortet til Node 1. Servoen styres av ??????
* På «status» skjermen på Node 1 kan man se forskjellige verdier som blir sendt over CAN.
* ???Musikken som spilles av genereres av Node 2 ved å slå motoren av og på med en gitt frekvens og amplitude, og kommandoene for dette blir sendt fra Node 1.???

**Ekstra:**

* PCB
* Buffering of display in SRAM
* Graphics/film
* Music

**Ting å gjøre klart:**

* Altium
* Seriell kommunikasjon
* Kode
* Breadboard
* Sette opp spillet