

Applikasjonsutvikling (DAVE3600)

Mappe 3 – HiOA Kids (bokstavlæring)

Rudi André Dahle – s300373
24.09.2017
Høgskolen i Oslo og Akershus

Programfunksjoner

Når man starter spillet kommer man rett inn på startskjermen (fig. 1)



Fig. 1



Fig. 2

Siden appen er rettet mot barn i alderen 3-6 år så har jeg valgt å bruke et symbol de aller fleste, både små og store, kjenner igjen, "play"-knappen. Denne har jeg satt en pulserende animasjon på for å trekke oppmerksomheten mot knappen og indikere at man skal trykke på den. Denne animasjonseffekten har jeg brukt på alle knapper, i mer eller mindre grad.

Når man så har trykket på play-knappen, startes et fragment som fyller heler skjermen og som inneholder spilldelen av appen (fig. 2).

Slik spillet virker så plukkes det ut ti tilfeldige bokstaver fra alfabetet når spillet starter. Disse ti bokstavene er ikke kjent for brukeren, og brukeren skal gjette én etter én bokstav gjennom spillet, helt til han eller hun har klart å gjette alle ti bokstavene riktig.

Hver bokstav presenteres én etter én via en lydfil som spilles av for hver av de ti rundene som spillet varer, og det er nå opp til brukeren å gjette hvilken bokstav som leses opp. Brukeren får presentert fire alternativer nederst i skjermbildet (fig. 2), hvor bare én av alternativene er riktig svar, og de tre andre er tilfeldig plukket ut i fra resten av alfabetet.

Dersom brukeren vil høre bokstaven på nytt så kan han eller hun trykke på den runde blå knappen øverst i skjermbildet. Denne pulserer litt svakt for ikke å trekke oppmerksomheten bort i fra svaralternativene nederst.

Dersom brukeren gjetter feil bokstav vil han eller hun få beskjed om dette ved hjelp av en forhåndsinnspilt lydfil. Her er det én av tre feilmeldinger som spilles av tilfeldig når man velger feil.

Gjetter brukeren riktig så spilles én av fire lydfiler av som sier at brukeren har gjettet riktig bokstav, og skjermbildet oppdateres ved at man har fått et hjerte (fig. 3) og man får opplest neste bokstav man skal gjette.



Fig. 4

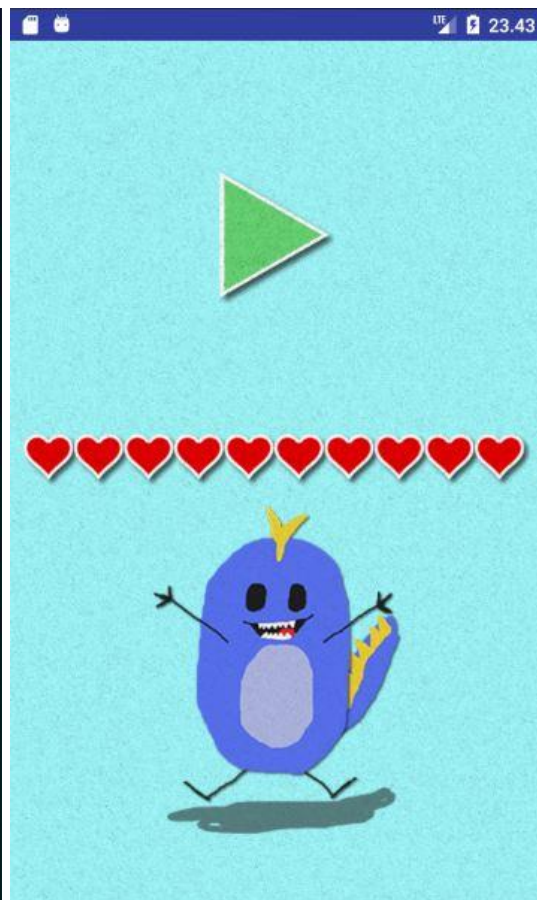


Fig. 5

Når brukeren har klart å gjette alle ti bokstavene som er plukket ut for denne runden, og hjertene er fylt opp, spilles det av en jubelmelding og play-knappen dukker opp igjen slik at man kan spille en runde til.

Designvalg

Da dette er en app rettet mot barn i barnehagealder så har jeg valgt å bruke mye farger, og gjort et forsøk på å få til en papir-effekt på designet.

Fonten jeg har valgt å gå for er en font som blir brukt til magnetbokstaver som mange barn er kjent med i fra kjøleskapet sitt hjemme. Her har jeg prøvd å variere fargene litt, men det kan forekomme at flere av svaralternativene har samme farge ettersom de blir tilfeldig utvalgt.

Som nevnt tidligere er startknappen formet som en play-knapp som de aller fleste barn er godt kjent med. Denne knappen pulserer tydelig slik at man skal forstå at den skal trykkes på.

Knappen for å spille av bokstavlyden på nytt har jeg prøvd å gjøre tydelig ved å ha et symbol av et øre som lytter inne i knappen. Denne knappen pulserer litt roligere for ikke å trekke oppmerksomheten bort i fra svaralternativene.

Svaralternativene blir presentert nederst på skjermen, og alle pulserer likt som play-knappen. Her får man valget mellom fire bokstaver, for ikke å gjøre det for vanskelig for de aller minste.

Hjertene har jeg valgt å ha med for å gi brukeren en følelse av hvor langt i spillet man har kommet, hvor mange bokstaver det er igjen og at man får en følelse av at man vinner noe for hver runde.

Dinosaurtegningene er der for å livne opp skjermbildet litt og å gjøre det mer innbydende for barn.

Teknisk

Jeg var litt bekymret i starten for at dette prosjektet ville bli litt for lite, og ikke utfordrende nok. Men jeg fant fort ut at når jeg skulle benytte meg av lydfiler og grafiske elementer så var det ikke så enkelt som jeg hadde trodd.

Det jeg opplevde var at elementer og metoder sto og ventet på hverandre i en kø, og ble utført sekvensielt.

For å kunne opprettholde en fin flyt i det grafiske samtidig som lydene ble spilt av på riktig måte måtte jeg opprette asynctasks til å ta seg av lydfilene i bakgrunnen. For å forhindre at lyder ble automatisk avbrutt midt i avspillingen måtte jeg sette en liten timer for å kunne sikre at lydene fikk tid til å spille ferdig (dette gjelder i hovedsak riktig-meldingene). Jeg benyttet meg først av Thread.sleep, men fant fort ut at denne bare setter hele UI-tråden på vent, som førte til at alle animasjoner stoppet opp midlertidig. Jeg fant ut at Handler().postDelayed tilbyr den løsningen jeg var på jakt etter. Utfordringen nå var å ha en dynamisk timer på denne metoden som tillot meg å ha ulik lengde på lydfilene. Dette fikk jeg til ved å benytte meg av MediaMetadataRetriever, som tillot meg å hente ut lengden på en lydfil. Da kunne jeg sette Handler().postDelayed til å vente like mange millisekunder som den aktuelle lydfilen varte.

Refleksjon

Dette var en litt vrien oppgave å komme i gang med, for jeg hadde først ingen idé om hva jeg skulle lage, eller hvordan det skulle se ut når jeg nærmet meg denne idéen her.

Jeg ville i utgangspunktet animere dinosaurene. Men dette viste seg å kreve mer tid enn jeg hadde tilgjengelig. Men prosessen med å teste dette førte meg til en del av designvalgene jeg endte opp med.

Som nevnt tidligere trodde jeg dette var en litt for enkel oppgave, men jeg har fått bryne meg på noen løsninger jeg ikke har fått brukt noe særlig av i de tidligere oppgavene.