



MatteMoro

FAG: DAVE3600 / MAPPEINNLEVERING 1

SULTAN AVTAJEV / S199219

Innhold

Innledning.....	2
Kravspesifikasjon	3
Funksjonelle Krav	3
Ikke-Funksjonelle Krav.....	3
Teknologiske valg	4
Utviklingsmiljø	4
Datalagring	4
Arkitektur og kode	4
Navigasjon	4
Design og brukervennlighet.....	5
Designprinsipper	5
Fargevalg og ikoner.....	5
Tilbakemelding til bruker.....	5
Utfordringer og løsninger	5
Flytskjema og tiltenkt design.....	6
Screenshots	6
Flytskjema.....	6
Planlagt design	7
Endelig design.....	8
Testing.....	10
Enhetstesting	10
Manuell Testing	10
Brukertester	10
Konklusjon	10
Fremtidige forbedringer og utvidelser	10
Kilder.....	11

Innledning

Hei og velkommen til denne rapporten! Hvis du er nysgjerrig på hvordan man kan få små barn til å bli interessert i matte, så er du kommet til rett sted. Denne rapporten handler om en app jeg har laget, som jeg har kalt "MatteMoro". Det er en morsom og enkel app som hjelper barn med å få teken på addisjon.

Appen er laget for små barn, omtrent i alderen 6-10 år. Jeg valgte denne aldersgruppen fordi det er viktig å starte tidlig med matte. Og siden mange små barn ikke kan lese ennå, har jeg prøvd å gjøre appen så enkel og visuell som mulig.

I denne rapporten skal jeg fortelle deg alt om hvordan jeg lagde appen, fra de tekniske greiene til hvordan jeg prøvde å gjøre den brukervennlig og morsom for barna. Så, la oss dykke inn!

Kravspesifikasjon

Funksjonelle Krav

1. Hovedskjerm

- Skal inneholde tre knapper: "Start spill", "Om spillet", og "Preferanser".

2. Spill

- Skal vise addisjonsoppgaver på skjermen.
- Skal ha knapper for tallene 0-9 for brukerinntput.
- Skal kunne vise 5, 10 eller 15 unike spørsmål per spill basert på brukerens valg.

3. Data Lagring

- Skal lagre 15 regnestykker og tilhørende svar i en XML-fil.

4. Randomisering

- Skal vise regnestykkene i en tilfeldig rekkefølge.
- Skal ikke vise det samme regnestykket mer enn en gang i en enkelt sesjon.

5. SharedPreferences

- Skal tillate brukeren å endre antall spørsmål per spill (5, 10 eller 15).
- Skal tillate brukeren å velge språk (norsk eller tysk).

6. Tilbakemelding

- Skal gi konstruktiv og lærerik tilbakemelding på riktige eller feil svar.

7. Avbrytelsesdialog

- Skal vise en dialogboks som spør om brukeren virkelig ønsker å avslutte spillet.

Ikke-Funksjonelle Krav

1. Brukervennlighet

- Appen skal være enkel å bruke og målrettet mot små barn.
- Minimal bruk av tekst, hovedsakelig visuell navigasjon.

2. Design

- Skal følge designregler fra [developer.android.com](https://developer.android.com/design).
- Skal ha ulikt layout i stående og liggende modus.

3. Internasjonalisering

- Skal støtte både norsk og tysk språk.

4. Teknisk

- Minimum API-nivå: 28
- Emulator: Pixel 7 1080*2400 420dpi

5. Dokumentasjon

- Skal inkludere en PDF-rapport under res/raw folderen.
- Rapporten skal inneholde skjermbilder og begrunnelser for designvalg.

Teknologiske valg

Utviklingsmiljø

Jeg brukte Android Studio for å utvikle appen, med Java som programmeringsspråk. Java er et velkjent og solid språk for Android-utvikling, og det gir meg all den fleksibiliteten jeg trenger for å bygge en robust og funksjonell app.

Datalagring

Data i denne appen er lagret på to måter: via XML-filer og **SharedPreferences**. XML-filene brukes for å lagre forhåndsdefinerte regnestykker, noe som gir meg muligheten til å enkelt endre dem uten å måtte gå inn i koden. **SharedPreferences** brukes for å lagre brukerpreferanser som antall spørsmål i en spillrunde og valgt språk, noe som bidrar til en sømløs brukeropplevelse.

Arkitektur og kode

Koden er organisert i klare og forståelige klasser:

- **HovedAktivitet:** Dette er startpunktet for appen. Her kan brukerne navigere til de forskjellige delene av appen.
- **InnstillingerAktivitet:** Håndterer brukerpreferanser som antall spørsmål i en spillrunde og valgt språk.
- **OmSpillAktivitet:** Gir brukerne informasjon om hvordan spillet fungerer.
- **ResultatAktivitet:** Viser resultatene etter at en spillrunde er fullført.
- **SpillAktivitet:** Dette er der selve spillet finner sted. Her genereres og vises regnestykkene.
- **SprokAktivitet:** Lar brukerne endre språk i appen.

Navigasjon

Navigeringen i appen er rettfram og brukervennlig. Appen starter med **HovedAktivitet**, der brukerne kan velge mellom ulike alternativer som "Start spill", "Om spillet", og "Innstillinger". Når brukeren gjør et valg, åpnes den tilsvarende aktiviteten. For å unngå at tidligere aktiviteter legger seg på stacken og bruker unødvendige ressurser, har jeg sørget for at navigasjonen tilbakestill stacken der det er nødvendig.

Design og brukervennlighet

Designprinsipper

Jeg har fulgt designretningslinjene fra developer.android.com for å sikre en intuitiv og brukervennlig opplevelse. Dette har resultert i en enkel og klar navigasjonsstruktur, og et layout som fungerer godt i både stående og liggende modus.

Fargevalg og ikoner

Fargepaletten er spesielt utformet med tanke på små barn, vår primære målgruppe. Appen bruker blå knapper på en hvit eller lys bakgrunn, noe som skaper en rolig, men likevel engasjerende atmosfære. Blått er en farge som ofte assosieres med tillit og ro, mens den lyse bakgrunnen gjør det enkelt å sette søkelys på innholdet.

Tilbakemelding til bruker

Når det gjelder tilbakemelding, får brukeren beskjed om svaret er "riktig" eller "feil" umiddelbart etter at de har svart. Dette gir en form for øyeblikkelig gratifikasjon eller korrigerende, som er viktig for å holde oppmerksomheten hos små barn. Etter å ha fått denne tilbakemeldingen, har brukeren valget mellom å prøve igjen eller gå videre til neste spørsmål, noe som gir dem en viss kontroll over egen læring.

I tillegg har spørsmålssiden en "Hjelp"-knapp som gir en hjelpetekst for oppgaven. Dette er en pedagogisk ressurs som kan hjelpe barnet hvis de står fast, og gir dem den støtten de trenger for å forstå oppgaven bedre.

Utfordringer og løsninger

Min første utfordring var å finne det beste verktøyet for design av appens brukergrensesnitt. Etter litt research bestemte jeg meg for å bruke Figma, som er et populært verktøy for å designe frontend. Jeg brukte noen dager på å lære meg programmet og utviklet et utkast til design som jeg var veldig fornøyd med.

Men da jeg skulle eksportere designet, møtte jeg på et problem: Figma har direkte støtte for eksport bare hvis man bruker Jetpack Compose med Kotlin. Siden jeg utviklet appen i Java, ble dette en betydelig hindring.

Jeg prøvde å løse dette ved å finne en Figma-plugin som kunne eksportere designet til XML-format. Selv om pluginnen fungerte, genererte den en svært rotete og uorganisert XML-fil. Dette førte til at jeg måtte bygge XML-filene fra scratch, noe som var tidkrevende og førte til at jeg ikke klarte å implementere designet akkurat som jeg hadde planlagt.

På grunn av tidsbegrensninger måtte jeg derfor kompromisse på designet. Men denne erfaringen har gitt meg verdifull innsikt i viktigheten av å sjekke kompatibilitet mellom ulike utviklingsverktøy og programmeringsspråk, og hvordan man kan navigere gjennom slike utfordringer.

Flytskjema og tiltenkt design

Link til flytskjema (tilgjengelig for alle med link):

<https://www.figma.com/file/8SSmI9kWITl8ReuKXWdU5v/Flytskjema?type=whiteboard&t=hRCdytHnjFG29x00-1>

Tips! Hold inne CTRL og bruk «Scroll» til å Zoome inn i de forskjellige delene.

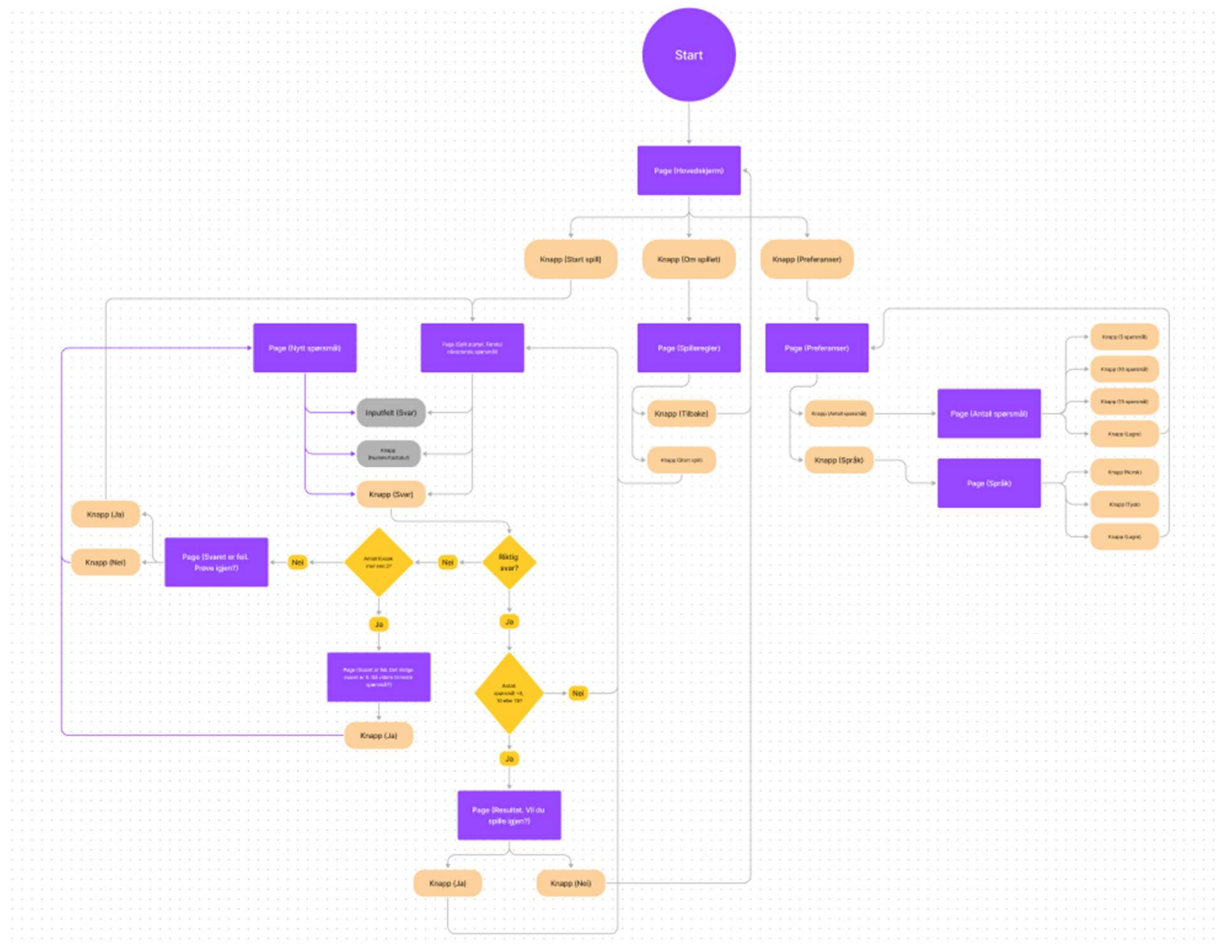
Link til design (tilgjengelig for alle med link):

<https://www.figma.com/file/Mlf8NkWOlNnTXAoMgt9sm/Untitled?type=design&mode=design&t=RCdytHnjFG29x00-1>

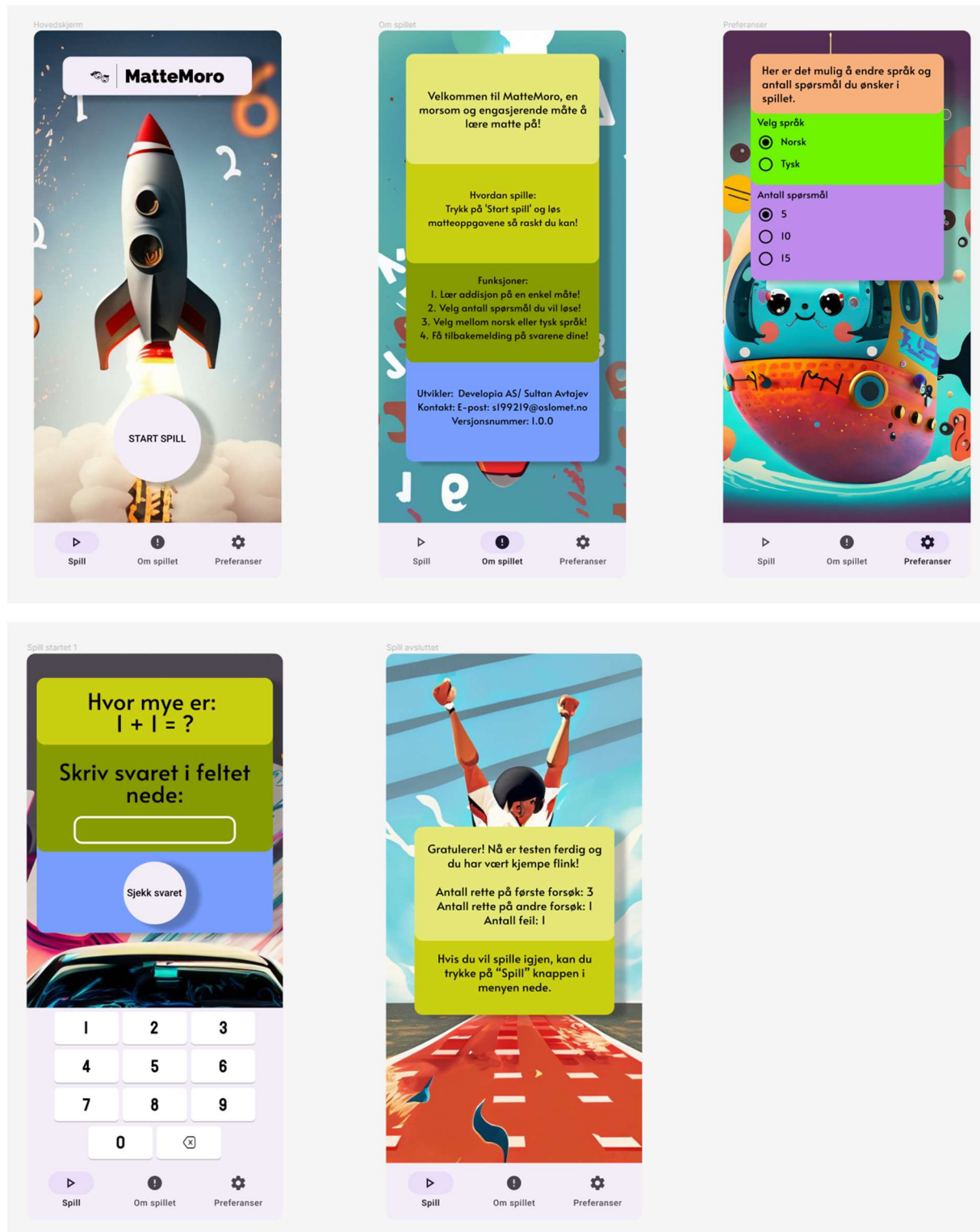
Tips! Hold inne CTRL og bruk «Scroll» til å Zoome inn i de forskjellige delene.

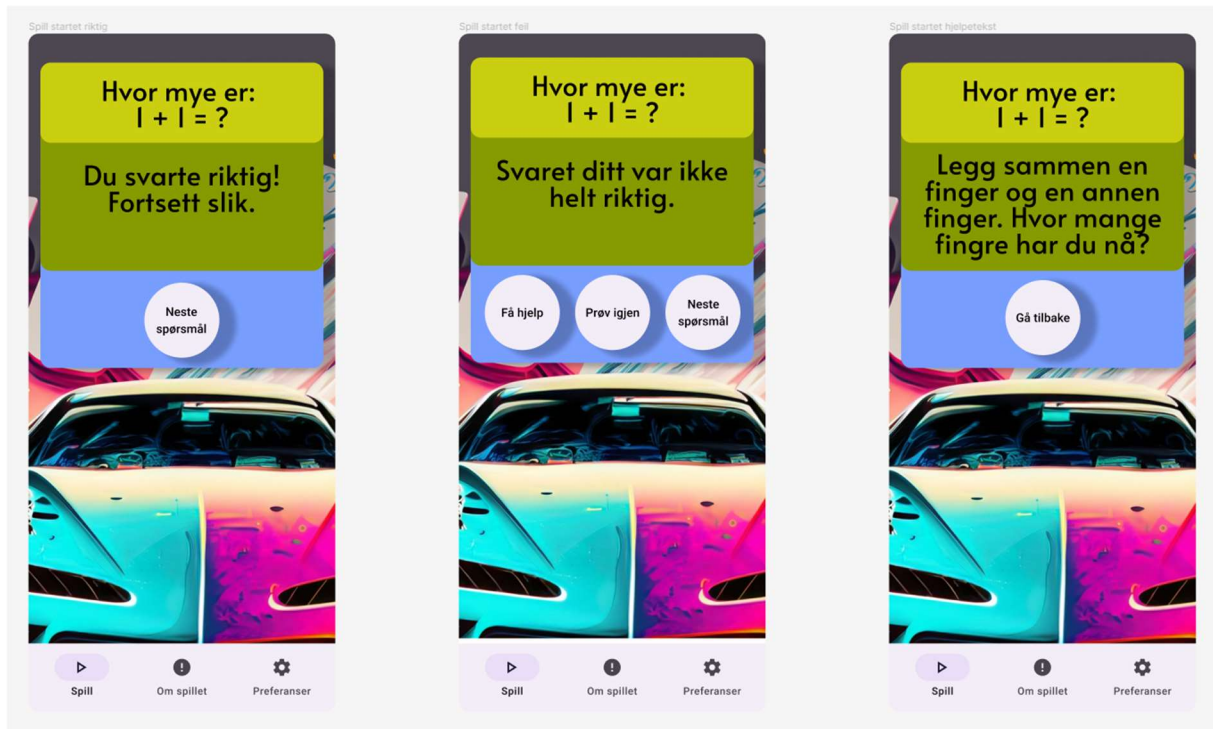
Screenshots

Flytskjema

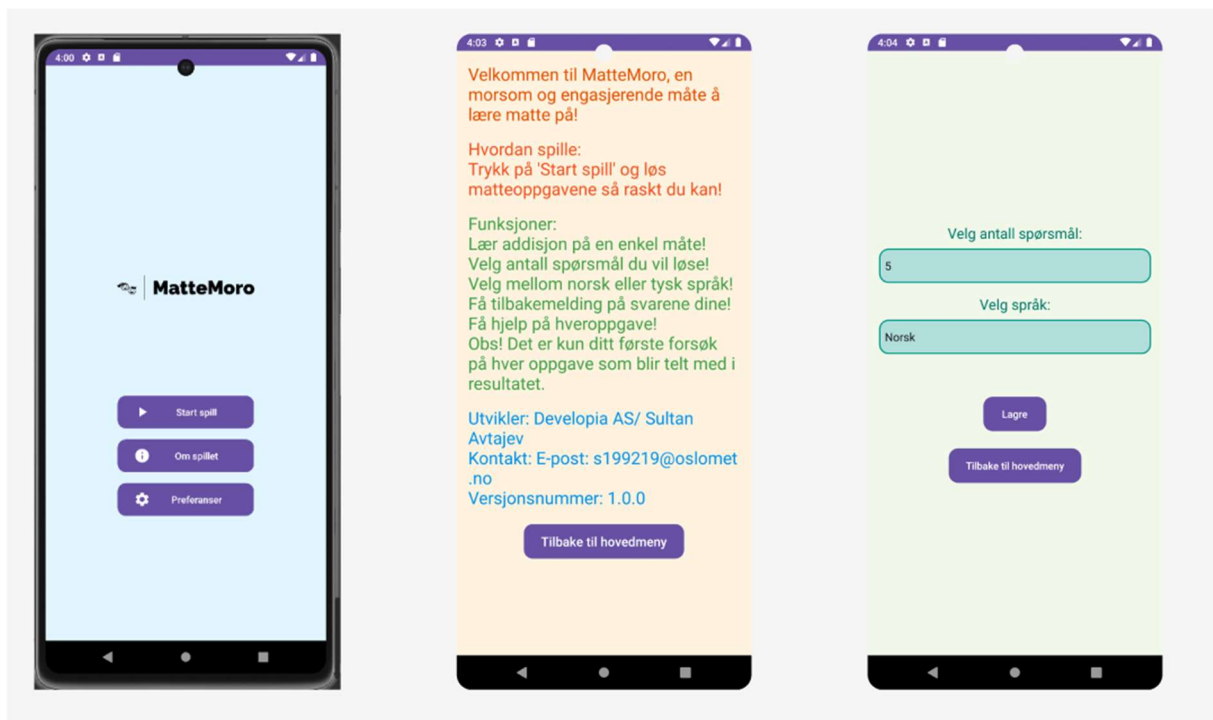


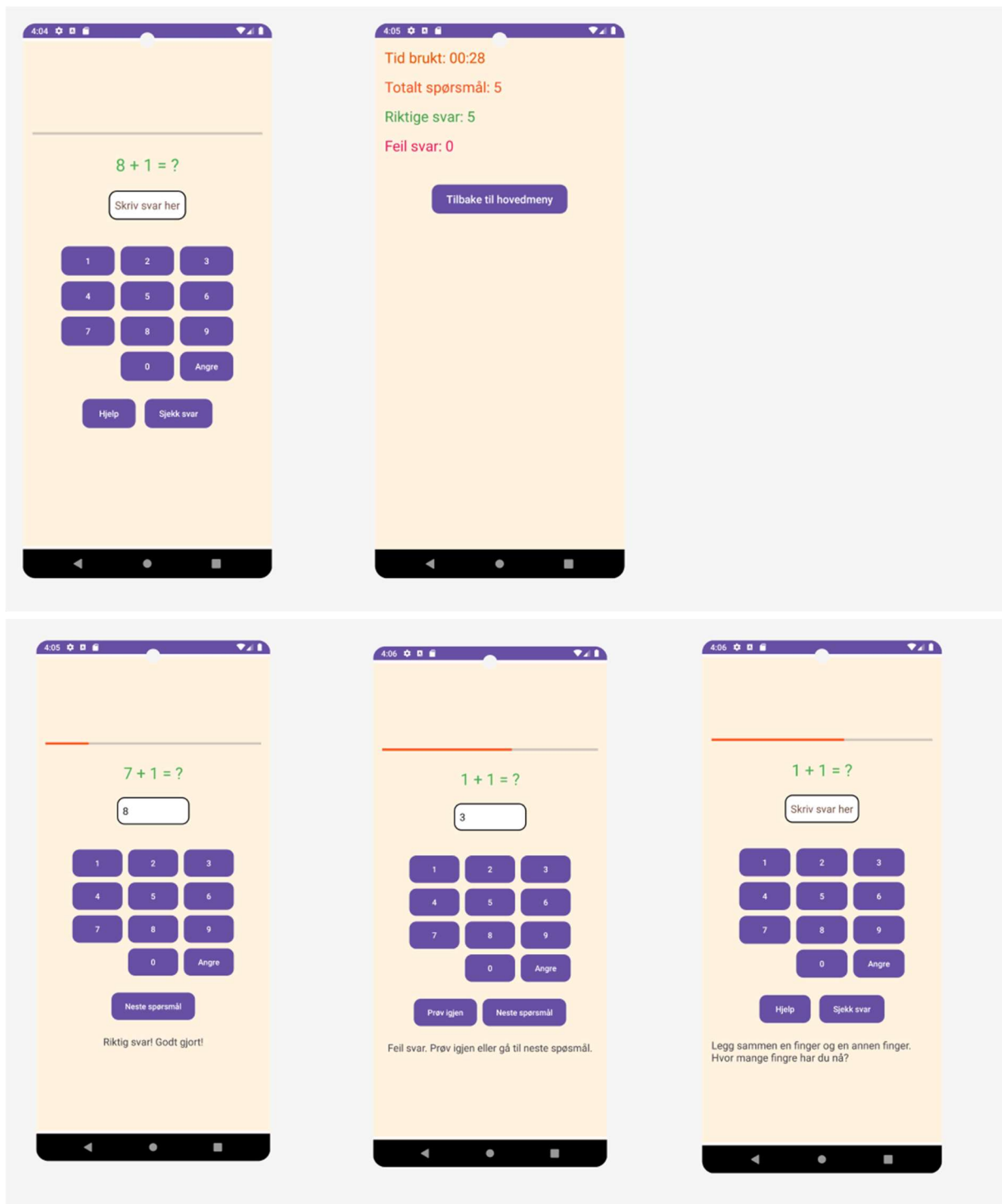
Planlagt design





Endelig design





Testing

Testing har vært en integrert del av utviklingsprosessen for å sikre at appen er både funksjonell og brukervennlig. Her er noen av metodene jeg brukte:

Enhetstesting

Jeg brukte JUnit for å utføre enhetstester på kritiske deler av koden. Dette inkluderte testing av logikk for generering av regnestykker, lagring av brukerinnstillinger, og navigasjon mellom ulike aktiviteter.

Manuell Testing

I tillegg til enhetstesting, utførte jeg også manuell testing for å sjekke brukeropplevelsen og interaksjonsflyten i appen. Dette inkluderte å teste appen i forskjellige skjermorienteringer, og å verifisere at brukerinnstillinger blir lagret og hentet korrekt.

Brukertester

Selv om tiden var begrenset, klarte jeg å arrangere en kort brukertest med yngste søsken (9 år) til min samboer. Dette ga verdifull innsikt i hvordan appen faktisk blir brukt, og hva som kan forbedres. Tilbakemeldingen fra disse testene ble brukt for å gjøre små justeringer i brukergrensesnittet og i tilbakemeldingssystemet.

Konklusjon

Dette prosjektet har vært en utfordrende, men fruktbar reise gjennom utviklingen av en matteopplæringsapp for små barn. Selv om jeg støtte på noen utfordringer, spesielt med hensyn til frontend-design og tidsbegrensninger, har jeg lyktes med å skape en funksjonell og engasjerende app. Den oppfyller det grunnleggende målet om å hjelpe barn med å utvikle matematiske ferdigheter, samtidig som den gir rom for videre forbedringer.

Fremtidige forbedringer og utvidelser

Appen er et solid fundament, men det finnes mange veier for videre utvikling:

Utvidelse av regneoppgaver: Legge til flere typer matematiske operasjoner som subtraksjon, multiplikasjon og divisjon.

Belønningssystem: Implementere et poengsystem eller andre former for belønninger for å øke brukerengasjement.

Detaljert tilbakemelding: Forbedre tilbakemeldingene til brukerne med mer detaljerte forklaringer.

Omfattende brukertesting: Gjennomføre flere runder med brukertesting for å innhente mer nyttig tilbakemelding.

Designforbedringer: En viktig del av fremtidsplanene er å oppgradere designet til det nivået jeg opprinnelig hadde planlagt. Dette inkluderer å finjustere brukergrensesnittet og implementere en høyere kvalitet på visuelle elementer.

Jeg ser frem til å fortsette arbeidet med appen, og jeg er spesielt ivrig etter å implementere designforbedringene for å realisere appens fulle potensial.

Kilder

<https://developer.android.com/develop/ui/views/layout/declaring-layout>

<https://developer.android.com/develop/ui/views/layout/linear>

<https://developer.android.com/develop/ui/views/layout/relative>

<https://developer.android.com/develop/ui/views/layout/binding>

+ video og annet materiale tilhørende faget på canvas