

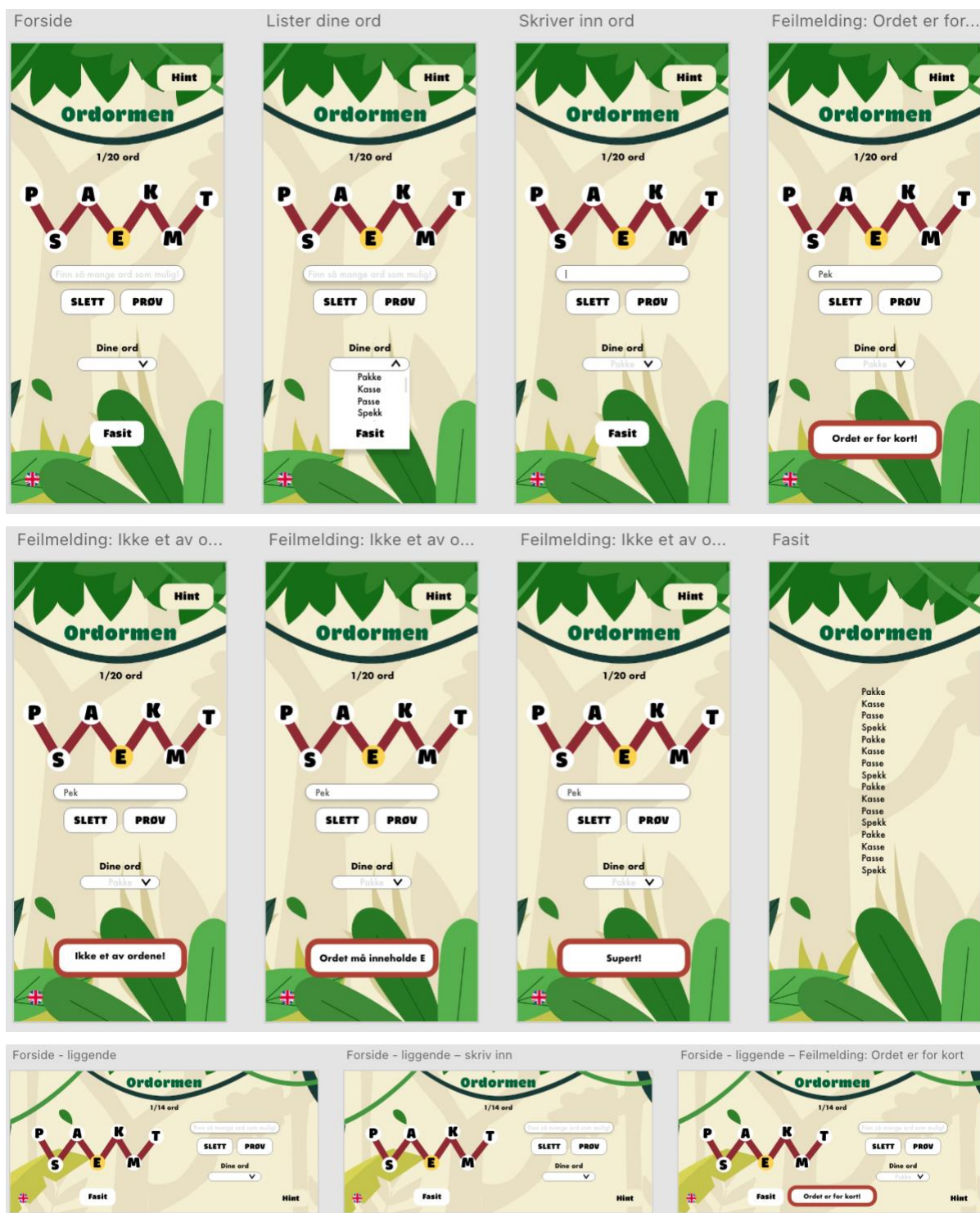
Ordormen - en spillapplikasjon

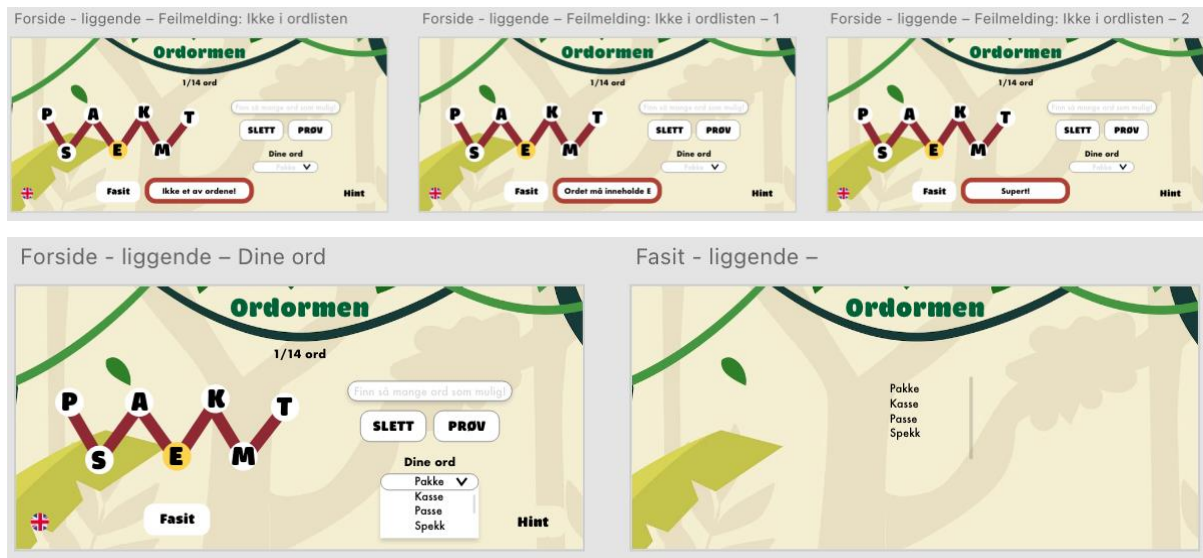
Appen Ordormen er en digital ordspill applikasjon hvor bruker skal finne ord, hvor målet er å inkludere bokstaven i midten av stjerna i alle ordene. Applikasjonen er inspirert av VGs ordstjerne (se figur 1.1). Ordormen inneholder tilsvarende funksjoner som ordstjernen nemlig bokstavknapper et tekstfelt, slettknapp, prøvknapp, ordoversikt, tilbakemeldinger og mulighet for hint ved at deler av et ord blir synlig. Norsk bokmål er hovedspråket for applikasjonen, men appen er også oversatt til engelsk. Det er mulig å redigere språkinnstillingene på mobilen som fører til oversettelse av innholdet i grensesnittet. Dette gjør at spillet er inkluderende på tvers av språk for å treffe en bredere målgruppe. Mappe 1 startet med studering av VGs ordstjerne, etterfulgt av en designfase, koding av layout og kode applikasjonen med Java. Underveis i kodeprosessen ble det gjort små endringer og kjøring for å raskt oppdage og rette opp i funksjonalitet som trengte forbedring. Hver prosess blir forklart nærmere senere i rapporten.



Figur 1.1 - VGs ordstjerne

Prosjektarbeidet startet med å studere ordstjernen som VG.no publiserer daglig. Testet spillet ved å finne ord, studere tilbakemeldingene, hvordan hintene er utformet, listen med funnede ord og annen generell funksjonalitet. For å gjøre kodejobben enklere ble det laget en skisse for stående og liggende skjermbilde i AdobeXD, en slags prototype, som visualiserer grensesnittet til Ordormen. Her inkluderes illustrasjoner, knapper, fonter, typografi og plasseringen av dette, se figur 1.2.





Figur 1.2 - Skisser i AdobeXD

Tema og målgruppe

Jungel er temaet for appen, og er derfor spillets navn ble Ordormen. Det ble tatt inspirasjon fra eksotisk jungel og ormen skal forestille en slange som representerer ordstjerna. Designet er en kontrast til det enkle uttrykket som VGs ordstjerne bærer preg av. Ordormen er en barnevennlig versjon av spillet som er rettet mot målgruppen: Barn og unge som liker ordspill, i alderen 8-16 år. For å tydeliggjøre målgruppen og sette seg inn i den mulige appbrukeren, ble det laget en personas for å være bevisst på målgruppen applikasjonen skal kommunisere med (Bertelsen, 2021), se figur 2.1.

PERSONAS



Navn: Marthe Bendiksen
Alder: 12 år
Utdannelse: 7.klasse
Bosted: Bergen
SOME: Snapchat, Instagram

Bio: Marthe Bendiksen er en kreativ jente som liker skole og særlig norsktimene. Hun har konkurranseinstinkt og når hun begynner med en oppgave vil hun gjøre alt for å klare å fullføre. Å leke og spille spill som inneholder å gjette og stave ord synes hun er veldig artig, som kryssordene i Donaldbladene og wordfeud på telefonen. Hun er kreativ og i dagligdagen dikter hun opp ord som hun synes burde være en del av ordforrådet. Bendiksen har en lang reisevei med bussen hver dag til og fra barneskolen hvor hun pleier å holde på med mobilen.

Utfordringer: Synes ikke det alltid er lett å se sammenhenger mellom hint og svar i vanskelige oppgaver. Hun kjeder seg på bussturene hver dag.

Målsetninger: Å bli flinkere å stave i norsk og lære seg flere engelske ord. Bruke tiden sin effektivt på lange bilturer eller til og fra skolen.

Teknologi: Bendiksen har hatt smarttelefon i noen år allerede og har god kjennskap til ulike sosiale apper, spill og lignende. Hun bruker mest telefonen i sin hverdag.

Figur 2.1 - Personas (Foto: https://www.freepik.com/free-photo/cute-cheerful-little-girl-orange_9479204.htm#query=10%20year%20old%20girl&position=44&from_view=search)

Designvalg

Verdiord kan være med å gjøre budskapet tydelig og formidle appens verdier og mål (Sagdahl, 2019). Leken, natur og elegant er de tre ordene som beskriver applikasjonen som lages i dette prosjektet. Samlet moodboard som inkluderer verdiordene, farger og fonter er også laget, se figur 3.1. Et moodboard er en visuell samling av stemning, følelser, tanker, konsept og stil uttrykt gjennom bilder og tekst (Clancy, 2022). I moodboardet vises blant annet illustrasjonsstilen, fontfamilien som ønskes brukt. Inspirasjon fra nettet gjorde utviklingen av appdesignet enklere ved å samle tankene for konseptet på ett sted.



Figur 3.1 - Moodboard

Ormen er tegnet i Adobe Illustrator og eksportert inn i drawable-mappene som png for å være tilpasset skjerm, se figur 3.2. Jungelbakgrunnen er også illustrert i Adobeprogrammet og eksportert på samme måte men i to formater, et stående og et liggende (se figur 3.3). Begge illustrasjonene har et enkelt og lekent uttrykk som er i 2D illustrasjonsstil som i kombinasjon kan minne om hvordan mange eldre tegneserier på trykk og på tv er illustrert (se figur 3.4).



Figur 3.2 – Ormen



Figur 3.3 - Jungelbakgrunn, horisontal og vertikal



Figur 3.4 - Bakgrunn og orm i kombinasjon

For å skape en mer kreativ plassering av bokstavene som skal brukes i ordspillet er de plassert i vinkler som danner slangen (se figur 3.4). For å skape en kontrast til bakgrunnen og den røde slangen er bokstavknappene hvite med sorte versaler. Versaler er store bokstaver på fagspråket (Rannem, 2022). Siden den midterste bokstaven skal være inkludert i alle fasitordene, er denne uthevet med en gul varm farge som skiller seg fra de andre men har det varme uttrykket ved seg som resten av stilen. Knappen med gulfargen kan være med å påminne brukeren om å inkludere bokstaven i alle ordene. Dersom bruker derimot glemmer midterste bokstav vil appen gi tydelige tilbakemeldinger, som vil bli forklart mer senere i rapporten. Den gule knappen indikerer gult trafikkllys som indikerer “pass på”, dermed opptreer fargen som en metafor for å være observant på denne spillregelen. Gul fargepsykologi tilsier at fargen er positiv, kreativ og uttrykker motivasjon noe som nettopp ønskes av den midterste knappen, at den skal vekke kreativiteten og motivasjon hos bruker (Røyer, Hanselmann, 2022).

Fontene som er brukt forsterker verdiordene nevnt tidligere. Fonten Louis George Café (Yining, 2022) og Bakso Sapi (Locomotive, 2018), tilhører begge den groteske fontfamilien som vil si typografi uten seriffer (Rannem, 2021). Groteskfonter er vist å være enklere å lese på skjerm (Johansen, 2018), så fontene er tilpasset den flaten det er designet til. Appdesignet består av to fonter, da får man balanse og et skille mellom knapper som er Louis George Café, som er funksjoner brukeren kan interagere med. Overskrifter og hint er en annen font, Bakso Sapi, for å indikere at dette er tekst man ikke interagerer med og heller gir ut beskjeder og er tittelen på appen.

Fontene følger et hierarki i punktstørrelsen. Overskriften har høyest pkt-størrelse på 30dp, deretter kommer tilbakemeldingene og hintene, så kommer tekstfeltet bruker kan skrive inn ordene, deretter kommer teksten på knapper og hvor det står antall ord og listen med “Dine ord”. Variasjonen mellom de ulike fontstørrelsene skaper tydelig forskjell mellom viktig tekst og mindre viktig (Johansen, 2018). Lettleselige fonter og størrelser gjør at applikasjonen er tilpasset universell utforming ved å gjøre teksten lesbar og lett å navigere med (Uiux Studio, 2022). Den minste fontstørrelsen er 15dp som finnes ved “Dine ord”, dette gjør at selv den minste størrelsen er stor nok for mange til å kunne lese (se figur 3.6). Linjene med tekst er korte og konsise som gjør det enklere å lese for svaksynte (Moe, 2020). Har også en god linjeavstand, mellom tekstlinjene for oversiktighet og flyt i lesingen og få brukeren til å forstå budskapet som formidles i applikasjonen (Moe, 2020). For å skape et konsekvent design som er enkelt å lære for brukeren, er alle knappene utformet likt slik at man vet hvor bruker kan trykke når det er funksjoner som går an å trykkes har tilsvarende utforming gjennom form og farge (Johansen, 2021).

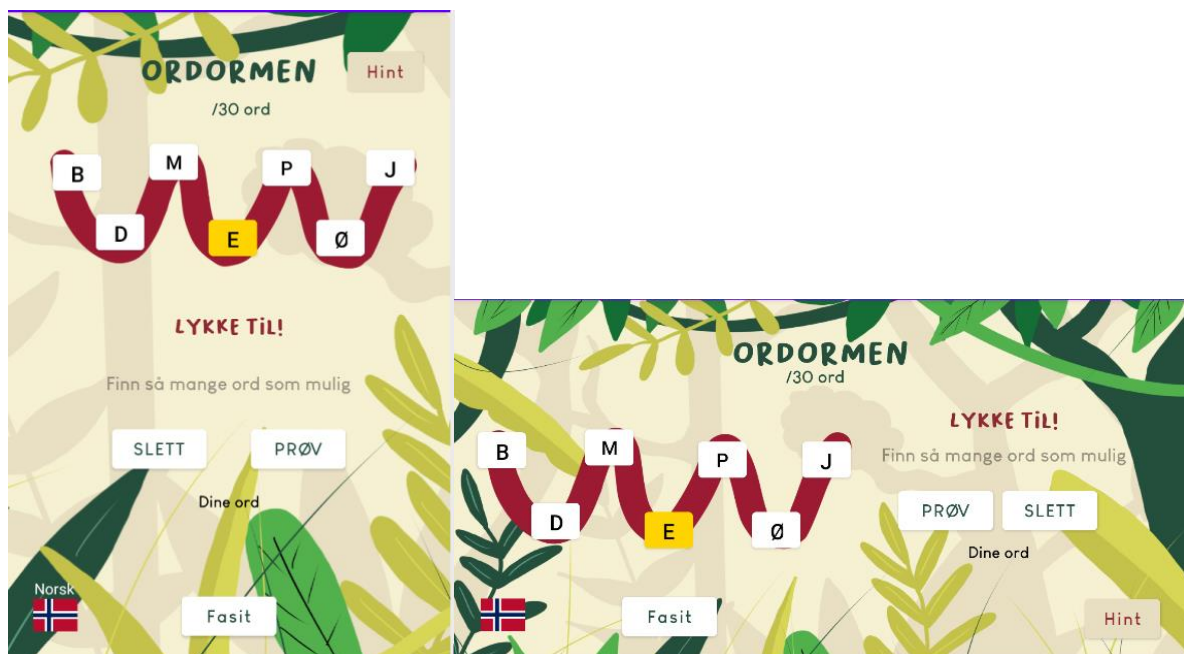


Figur 3.5 - Fontene



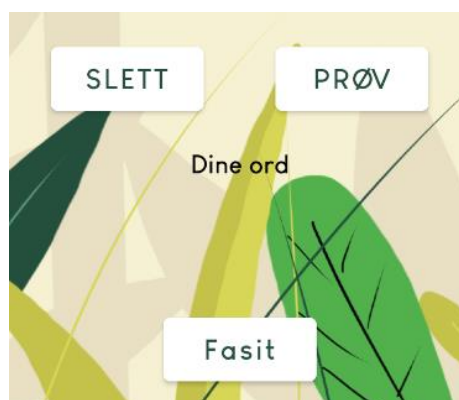
Figur 3.6 - Dine ord fontstørrelse

Avstand mellom elementer har også en effekt på lesbarheten (Johansen, 2021). Appen viser tydelig at bokstavene i slangen henger sammen fordi de har lik avstand til hverandre og er samlet på en rød linje (se figur 3.7). Ormen er i nærhet til knapper og tekstfeltet under som viser at figur og elementer hører sammen. Tekstfeltet og knappene "slett" og "prøv" står med nærhet til hverandre for å indikere at disse tre henger sammen. I tillegg til "dine ord", som har minimalt med avstand til knappene ovenfor viser at dette er også i tilhørighet med funksjonaliteten over, men er lavere i hierarkiet på grunn av lavere punktstørrelse på fonten (se figur 3.7). "Dine ord" er en huskeliste for bruker og står ikke i fokus. Hintknappen og fasitknappen er plassert med større avstand fra de andre midtstilte elementene for å ikke ta oppmerksomheten bort fra disse elementene. Hintknappen og fasitknappen skal brukes ved behov og er derfor lengre unna i hver sin retning. Knappene er kodet til å tilpasse lengden til typografien. Det gjør at når man skifter til engelsk blir knappenes størrelse justert automatisk etter lengden på ordene innenfor rammene.



Figur 3.7 - Avstand mellom elementer

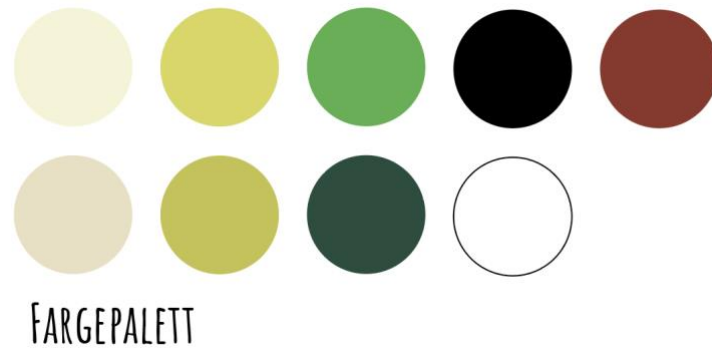
Tidligere studier som er gjort med forskjellige utformede knapper på digitale flater, resulterte det i at knapper med avrundede hjørner er mer attraktive og fristende å klikke på enn knapper som hadde skarpe kanter (Kushwaha, 2019). Avrundede kanter uttrykker varme, tillit og reduserer den kognitive belastningen (Kushwaha, 2019). Derfor har jeg brukt dette til å designe knappene i applikasjonen. Selve layouten er midtstilt med unntak av språket og flagget under (se figur 3.8).



Figur 3.8 - Avrundede knapper

Det ble laget et fargepalett til appen som det var ønsket skulle representere jungel (se figur 3.9). Fargepsykologi er effektiv måte å formidle sine mål og symbolikk i designet innen grafisk design (Røyert, Hanselmann, 2022). Farger har en mening og betydning som man kan utnytte

for å få fram et budskap og forsterke det gjennom å vekke følelser, skape minner og fange oppmerksomheten vår (Røyert, Hanselmann, 2022). Appens fargepalett består hovedsakelig av varme farger fra grønt, gul og rød, som tilsier varme og nærhet (Røyert, Hanselmann, 2022). Noe som ønskes at appen skal uttrykke, varme og at man kan vekke interesse hos brukeren til å gjette ord og spille.

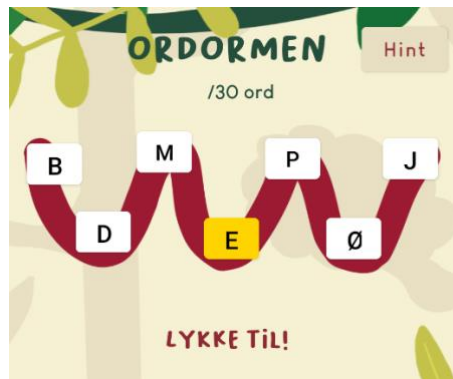


Figur 3.9 - Fargepalett

Rød ble brukt på slangen, overskriften og tilbakemeldingene fordi rød er en signalfarge (Røyert, Hanselmann, 2022). Den trekker oppmerksomhet og står i kontrast til den grønne bakgrunnen når man ser på fargesirkelen, se figur 3.10. Signalfargen rød passer til overskriften siden den skiller seg ut fra bakgrunnen. Rød skaper et helhetlig inntrykk og en rød tråd når flere av elementene er av tilsvarende farge som fontfargen på hint-knappen og “Lykke til” TextViewet (se figur 3.11). Rød kan passe til å representere slangen på bakgrunn av dens gift som kan oppleves aggressiv og farlig. Hintene som gis er et signal til brukeren og skal fange oppmerksomheten.



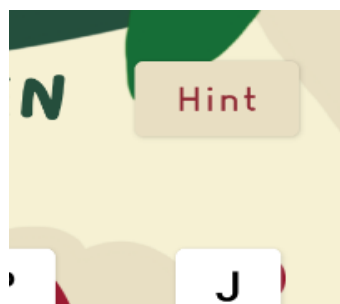
Figur 3.10 - Fargesirkel



Figur 3.11 - Bruken av fargen rød

Grønn er brukt i bakgrunnen, på blader og trær (se figur 3.3). Ro, prestisje og rikdom er fargepsykologien til grønn (Røyert, Hanselmann, 2022). Målet med appen er å leke og finne ordene som er gjemt i ordstjernen. De sju bokstavene danner en rik mengde med ord og brukeren kan bli rik på mestringsfølelse som er hensikten med spillet. Grønn forsterker nemlig dette. For å konsentrere seg skaper grønnfargen ro i bakgrunnen slik at det er mulig å trekke oppmerksomheten til de elementene som bruker skal interagere med.

Knappene “slett”, “prøv” og “fasit” er hvite som uttrykker renhet, uskyldighet og anonymitet (Røyert, Hanselmann, 2022). Fargen tar ikke oppmerksomheten fra bokstavene, tilbakemeldingene og hintene. I og med knappene er der for å gi feedback gjennom å slette bokstaver, sjekke svaret eller se fasiten, er de myke og nøytrale i kontrast til den fargerike bakgrunnen. “Hint”-knappen skiller seg ut ved å ha en beige bakgrunnsfarge med rød typografi. Knappen står ut fra de andre med at den gir deler av et fasitsvar og ønsker derfor at designet representerer dens funksjonalitet (se figur 3.12). En mer anonym knapp som blander mer inn i bakgrunnen har hensikten til å være mindre synlig og senke sannsynligheten av fristelse for brukeren til å få tips. Dette kan være med på å utfordre brukeren til kreativitet i høyere grad.



Figur 3.12 - Hint-knapp

Fargene på fontene er også med å skape et slags hierarki. Overskriften og knappene er mørk grønn og hintene som kommer opp der det står “Lykke til” er røde, samme farge som slangen (se figur 3.7). Vekting sier noe om tykkelsen til fonten, som ofte går fra light til bold (Uhr, 2021). Bokstavknappene og hintene har høyere vekting fordi det er viktig informasjon for brukeren. Overskriften har også høy vekting siden dette er spillets navn og logo. All annen tekst er i lavere vekting. Bokstavene, “dine ord” og default-teksten i tekstfeltet er nyanser av sort. Fargen sort er brukt til detaljene i bakgrunnsillustrasjonen og typografien plukker opp denne fargen og skaper en helhet. Lys-mørk kontrast kalles dette og uttrykker tidløshet, verdi og eleganse (Sporaland, 2019). Denne tydelige kontrasten tar også hensyn på svaksynte med tanke på universell utforming (Lu, Lindberg, 2022). Siden leken og elegant er to av verdiordene til løsningen, er det valgt en leken font i rødt og en mer elegant og tidløs font som er hyppigst brukt i appen som er brukt med sort og grønn farge.

Universell utforming er sentralt i digitale løsninger og fargeblindhet er dermed viktig å ta hensyn til (UUtilsynet, 2022). Applikasjonen består av grønnfarger og rødt, noe som kan være utfordrene for personer med protanopia-type fargeblindhet, rødblindhet (Lu, Lindberg, 2022). Da vil fargene rød og grønn bli ulike grønn-toner og ikke uttrykke så stor forskjell. En endring av dette kan øke applikasjonens verdi av universell utforming. Fargene er ikke avgjørende å skille fra hverandre for å kunne bruke appen, dermed beholdt jeg fargevalget ved innlevering.

Koden

Koden er strukturert med tydelige variabelnavn uten bruk av bokstavene æ, ø og å (se figur 4.1). Alle elementer i “activity_main.xml” og “activity_fasit.xml” har fått en id, slik at de enkelt kan hentes og refereres til via Javaklassene (se figur 4.2). Kommentarer i koden ble skrevet underveis, både for å huske selv hva deler av koden gjør, for å få andre til å forstå koden, og generelt skape oversiktlig kode som er lett å finne fram i senere. Stylingen av appen er gjort i layout for vertikal skjerm og layout-land for horisontalt design, men også “styles” og “colors” som er xml-filer under “values”, inneholder deler av designet. Fargene brukt er henvist til hex fargekoder i colors.xml (se figur 4.3) fra styles.xml som blant annet er filen som presiserer elementer sin størrelse, margins, constraint layout plasseringer og tekststørrelser (se figur 4.4).

```

TextView inputText;
TextView antallOrd;
TextView feilmelding, feilmelding1, feilmelding2, feilmelding3, feilmelding4;
TextView positiv;
Button buttonDelete, buttonCheck, fasiten, buttonHint;
Button buttonB, buttonD, buttonM, buttonE, buttonO, buttonP, buttonJ;
String word = ""; //Ordet som lages.
ArrayList<String> wordsFound = new ArrayList<>(); //String array med ordene som er funnet
ArrayList<String> wordsFound1 = new ArrayList<>();
ArrayList<String> fasitOrdeneList = new ArrayList<>();
ArrayList<String> wordsRemaining = new ArrayList<>(); //Array med ordene som gjenstår. Or
String[] ordFunnet;
String[] fasitOrdene;
String[] wordsLeft;
String hintWord = "";
ListView show;
String letterE;
ArrayAdapter<String> adapter;

int correctWordCounter = 1; //Teller riktige ord

```

Figur 4.1 - Variabelnavn eksempel

```

<androidx.constraintlayout.widget.ConstraintLayout
    android:id="@+id/buttonsLayout"

```

Figur 4.2 – ID eksempel

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <resources>
3      <color name="purple_200">#FFBB86FC</color>
4      <color name="purple_500">#FF6200EE</color>
5      <color name="purple_700">#FF3700B3</color>
6      <color name="teal_200">#FF03DAC5</color>
7      <color name="teal_700">#FF018786</color>
8      <color name="black">#FF000000</color>
9      <color name="white">#FFFFFFFF</color>
10     <color name="yellow">#FFD400</color>
11     <color name="green">#254E3C</color>
12     <color name="red">#8F2934</color>
13     <color name="light">#E8DEC2</color>
14 </resources>

```

Figur 4.3 - colors.xml


```

<style name="dineOrd">
  <item name="android:color">@color/white</item>
  <item name="android:radius">13dp</item>
  <item name="android:layout_width">wrap_content</item>
  <item name="android:layout_height">wrap_content</item>
  <item name="android:layout_marginTop">444dp</item>
  <item name="android:textColor">@color/black</item>
  <item name="android:backgroundTint">@color/white</item>
  <item name="android:textSize">15dp</item>
  <item name="fontFamily">@font/louis_george_cafe_bold</item>
  <item name="layout_constraintEnd_toEndOf">parent</item>
  <item name="layout_constraintBottom_toBottomOf">parent</item>
  <item name="layout_constraintVertical_bias">0.3</item>
  <item name="layout_constraintStart_toStartOf">parent</item>
  <item name="layout_constraintTop_toTopOf">parent</item>
  <item name="android:textAlignment">center</item>
</style>

<style name="dineOrdFasit">
  <item name="android:color">@color/white</item>
  <item name="android:radius">13dp</item>
  <item name="android:layout_width">300dp</item>
  <item name="android:layout_height">400dp</item>
  <item name="android:layout_marginTop">444dp</item>
  <item name="android:textColor">@color/black</item>
  <item name="android:backgroundTint">@color/white</item>
  <item name="android:textSize">15dp</item>
  <item name="layout_constraintStart_toStartOf">@+id/title</item>
  <item name="fontFamily">@font/louis_george_cafe_bold</item>
</style>

```

Figur 4.4 - Styles.xml

Det ble utført research på ulike layouts i Android Studio. Constraint Layout ble inkludert i Android Studio i 2016 og ble raskt defaultoppsettet fordi den tilbyr et bredt utvalg av egenskaper for å plassere elementer og klarer å skape komplekse oppsett på en enkel måte (Harchani, 2019). I tillegg med at designet inneholdt knapper som var plassert i ulike vinkler i forhold til blant annet slangeillustrasjonen ble en Constraint Layout valgt på bakgrunn av informasjonen som ble innhentet. Her jobber man med posisjoner og hvordan de står i forhold til andre elementer ved å eksempelvis anvende kodelinjer som du kan se i figur 4.5. Google og YouTube ble aktivt brukt for å gjøre research på hvordan å oppnå et layout som ønsket. I starten var det knyttet å forstå hvordan å bruke layouttypen. En ulempe var at Design-viewet prosjekterte appdesignet annerledes enn da emulatoren ble kjørt. Dermed måtte mye av plasseringer endres underveis, noe som tok tid for å skape et design som var tilsvarende skissen laget i AdobeXD.

```

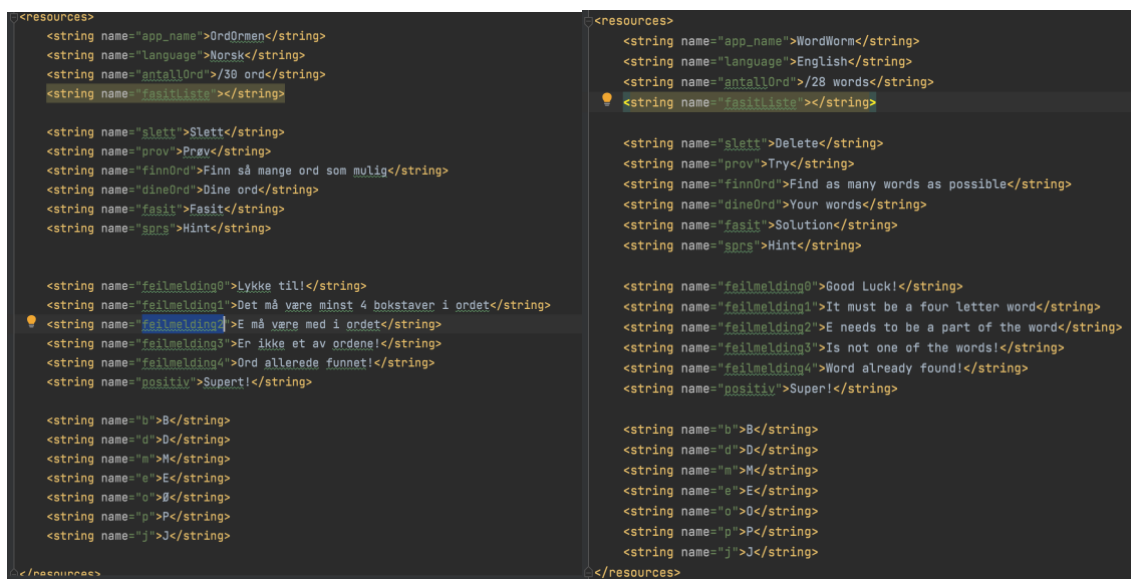
<item name="layout_constraintStart_toStartOf">@+id/title</item>
<item name="layout_constraintEnd_toEndOf">parent</item>
<item name="layout_constraintBottom_toBottomOf">parent</item>

```

Figur 4.5 - Constraint Layout posisjon

All tekst er samlet i strings.xml som både ligger i values og values-en (se figur 4.6). Dette fører til at man kan endre språket på applikasjonen ved å justere innstillingene på telefonen. Under

values ligger den norske versjonen av appen, mens under values-en ligger den engelske oversatte versjonen. Norsk er default for min app så det engelsk man kan endre til om man heller ønsker den versjonen. Språkendringen skjer som nevnt inne i telefonens innstillinger, men kunne gjerne som en videreutvikling av appen, inkludert en knapp inne i selve applikasjonen hvor man kan endre språket. Selv om det ikke er en knapp for språkendring direkte i appen, vises nåværende språk ved et flagg og tekst nederst i venstre hjørne for å informere brukeren om valgt språk (se figur 4.7). Flaggene er png-filer funnet på nett og er plassert i drawable-filene. Det er viktig at bildene heter det samme i drawable og drawable-en slik at korrekt bildeversjon blir hentet opp ved språkvalget. Figur 4.8 viser mappestrukturen til drawable-filene der bilder og illustrasjoner som bakgrunn, flagg, og orm er lagt inn med samme navn.



```
<resources>
  <string name="app_name">OrdOrmen</string>
  <string name="language">Norsk</string>
  <string name="antallOrd">30 ord</string>
  <string name="fasitliste"></string>

  <string name="slett">Slett</string>
  <string name="prov">Prøv</string>
  <string name="finnOrd">Finn så mange ord som mulig</string>
  <string name="dineOrd">Dine ord</string>
  <string name="fasit">Fasit</string>
  <string name="sprs">Hint</string>

  <string name="feilmelding0">Lykke til!</string>
  <string name="feilmelding1">Det må være minst 4 bokstaver i ordet</string>
  <string name="feilmelding2">E må være med i ordet</string>
  <string name="feilmelding3">Er ikke et av ordene!</string>
  <string name="feilmelding4">Ord allerede funnet!</string>
  <string name="positiv">Super!</string>

  <string name="b">B</string>
  <string name="d">D</string>
  <string name="e">E</string>
  <string name="o">O</string>
  <string name="p">P</string>
  <string name="j">J</string>
</resources>
```

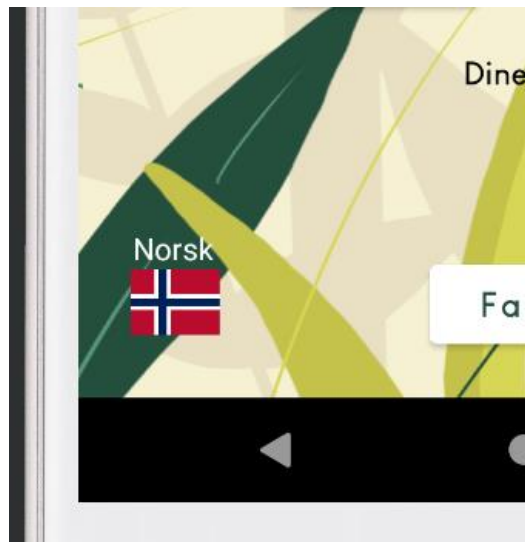
```
<resources>
  <string name="app_name">WordWorm</string>
  <string name="language">English</string>
  <string name="antallOrd">28 words</string>
  <string name="fasitliste"></string>

  <string name="slett">Delete</string>
  <string name="prov">Try</string>
  <string name="finnOrd">Find as many words as possible</string>
  <string name="dineOrd">Your words</string>
  <string name="fasit">Solution</string>
  <string name="sprs">Hint</string>

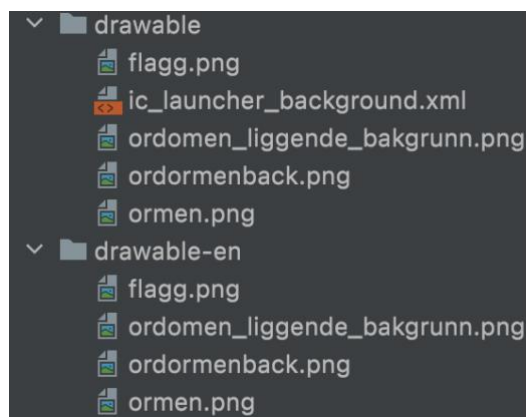
  <string name="feilmelding0">Good Luck!</string>
  <string name="feilmelding1">It must be a four letter word</string>
  <string name="feilmelding2">E needs to be a part of the word</string>
  <string name="feilmelding3">Is not one of the words!</string>
  <string name="feilmelding4">Word already found!</string>
  <string name="positiv">Super!</string>

  <string name="b">B</string>
  <string name="d">D</string>
  <string name="e">E</string>
  <string name="o">O</string>
  <string name="p">P</string>
  <string name="j">J</string>
</resources>
```

Figur 4.6 - Strings.xml

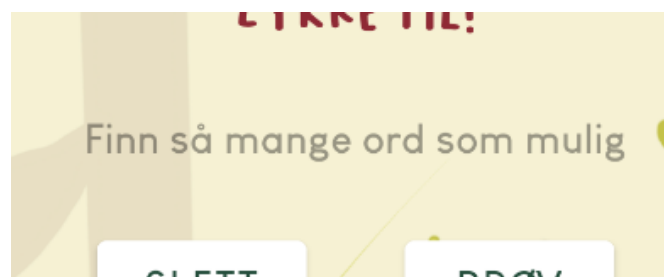


Figur 4.7 - Nåværende språk visualisering



Figur 4.8 - Mappestruktur drawable

Inputfeltet hvor bokstavene blir skrevet ut etter klikk på de ulike bokstavene, har en default-tekst, et visuelt hint til brukeren om hvilken tekst som skal skrives her nemlig “Finn så mange ord som mulig” (se figur 4.9). Hjelpeteksten løser problemet med å ikke vite hva man skal gjøre, før det oppstår (Lu, Lindberg, 2022).



Figur 4.9 - Visuelt hint; Finn så mange ord som mulig

Arrays.xml ligger også under values og values-en mappene. Dette er for å oppbevare fasitordene på de to ulike språkene men også to tomme array som anvendes når brukeren finner ord og holder kontroll på hvilke ord som er tatt og hvilke som gjenstår. Siden den norske versjonen har andre sju bokstaver enn den engelske versjonen blir også fasitordene forskjellige, derfor er det praktisk å ha arrays.xml som hentes opp avhengig av språkinnstillingene til brukeren.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string-array name="solution_array">
    <item>"ØMMEDE"</item>
    <item>"ØMME"</item>
    <item>"ØDEM"</item>
    <item>"ØDEDE"</item>
    <item>"PEPP"</item>
    <item>"MØDE"</item>
    <item>"MEDE"</item>
    <item>"JØDE"</item>
    <item>"JEPP"</item>
    <item>"JEEP"</item>
    <item>"EBBEDE"</item>
    <item>"EBBE"</item>
    <item>"DØPE"</item>
  </string-array>
  <string-array name="found">
  </string-array>
  <string-array name="remain">
  </string-array>
</resources>
```

Figur 4.10 - Arrays

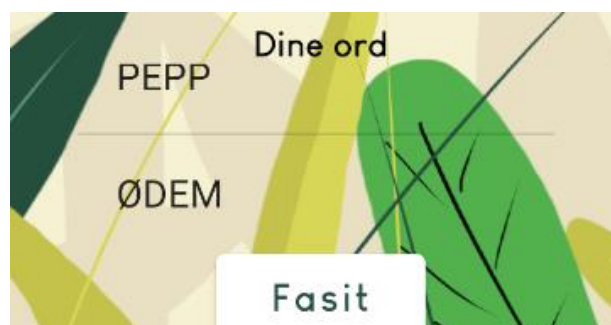
Fasiten er en egen Javaklasse, activity_fasit.xml som brukeren føres til når "fasit"-knappen blir trykket. Både stående og liggende er det overskrift, bakgrunnsillustrasjon og en skrollbar arraylist adapter som presenterer fasitordene (se figur 4.11). Disse listene er av samme type som når brukeren finner ord og lagrer dem i tilsvarende liste som bruker kan interagere med og utnytte for oversikt over ord som er funnet (se figur 4.12). De norske fasitordene og bokstavene ble hentet fra en av VGs tidligere ordstjerne, med E i midten. Fasitordene på engelsk ble funnet ved hjelp av nettsiden Wordtips

(<https://word.tips/unscramble/BEDMPJO/words-contain/e/length/4/>), hvor de samme bokstavene som på norsk ble skrevet inn og fikk engelske ord. Da ble det skrevet ned alle ordene som fantes som kombineres med E, ord med fire, fem og seks bokstaver.

Nåværende applikasjon har ingen tilbakeknapp på fasitsiden som gjør at bruker må anvende tilbakeknappen på telefonen. Dersom det var mer tid til å ordne, ville en tilbakepil bli plassert øverst i venstre hjørnet slik at man ikke er avhengig av telefonens tilbakeknapp. Dette hadde forbedret navigasjonen som kan være med på å skape et enklere grensesnitt å navigere fram og tilbake i tillegg til å redusere forvirringer (Miljødirektoratet, 2022). En slik pil vil være hensiktsmessig i forgrunnen som viser tydelig at brukeren ikke sitter fast men kan komme seg tilbake til forrige skjermbilde.



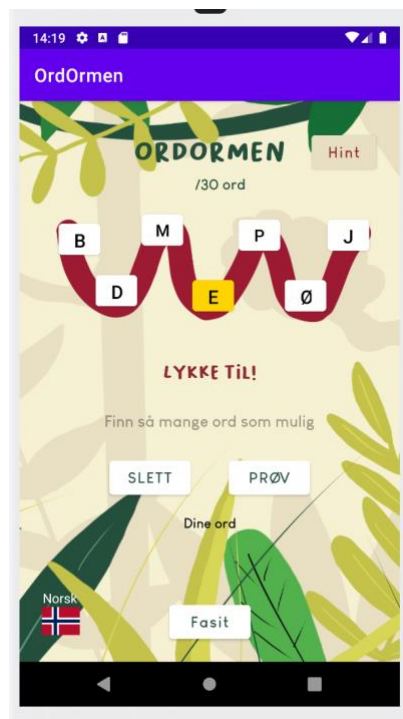
Figur 4.11 - Array med fasitordene



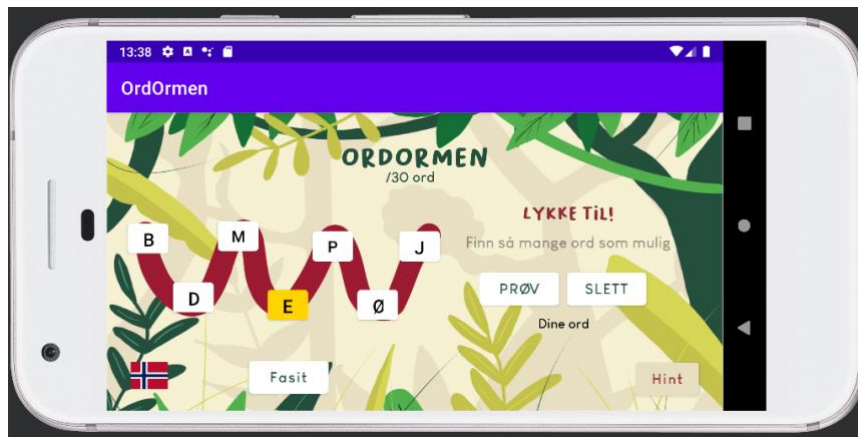
Figur 4.12 - Ord som er funnet array

Applikasjonen er tilegnet liggende og stående format ved å ha en merkbar forskjell i designet hvor elementene er plassert etter hvordan man holder mobilen, men bevarer alle knapper, tekst, farger og størrelser for å være konsekvent i grensesnittet for å oppnå læring etter noen gangers bruk (Johansen, 2021). Vertikalt er hovedelementene midtstilt, som overskrift, ormen, prøv og slett-knappene, tilbakemeldingene og lignende (se figur 4.13). Dette er for å trekke øynene mot midten av skjermen og fokusere på de viktige delene altså bokstavene, tekstfeltet,

knappene og tilbakemeldinger. Som nevnt tidligere har disse egne designforklaringer. Horisontalt er skjermen todelt, med informasjon liggende til venstre og høyre for midten (se figur 4.14). Dette tiltaket er gjort for å få plass til ormen og andre elementer. Flagg og fasitknapp er nederst på skjermen i begge retninger, mens hintknappen er flyttet fra høyre hjørnet til samme nivå som fasitknappen for å skape balanse. Ordormen med bokstavene er plassert til venstre og fragmentet med lykke til og til og med “Dine ord” er plassert til høyre (se figur 4.14). Siden leseretningen er fra venstre til høyre vil jeg bringe dette inn i løsningen. Fange blikket på det viktigste til venstre og deretter kan øynene bevege seg til der man trykker og får tilbakemeldinger. Dersom jeg skulle arbeidet mer med det horisontale designet ville jeg byttet om på “slett” og “prøv” knappene for å få mer konsekvent design men ble ikke tid til dette byttet. Overskriften, antall ord funnet og bakgrunnen er gjentagende i begge retninger for å være konsekvent.



Figur 4.13 - Vertikal appdesign



Figur 4.14 - Horisontal appdesign

Under overskriften er en teller og oversikt over ord, “/30 ord”, tallet til venstre for skråstreken vil dermed øke når man løser ordene (se figur 4.15). I en forbedret versjon ville jeg løst problemet med å ha /30 ord både på engelsk og norsk, i og med at dette er nå kodet i Javaklassen MainActivity. Forbedret versjon kunne heller ha vært varierende i strings for engelsk og norsk for å adaptivt etter hvor mange ord det er i den engelske versjonen siden den inkluderer tre færre ord enn på norsk.



Figur 4.15 - Ordteiler

Trykker brukeren på “Hint” blir det presentert deler av løsningsordene som gjenstår å finne. Koden sjekker om hintordet er funnet eller ikke ved å bruke to arrays. Når bruker finner et fasitord legges ordet inn i et array mens i et annet array blir det samme ordet fjernet fra fasitlisten (se figur 4.16). Dette er for å holde oversikt over hvilke hint fra hvilke ord som brukeren har igjen å finne. Koden sjekker derfor om ordet er funnet eller ikke og gir bare hint fra et av de resterende ordene i fasiten, med første og siste bokstav med en “_” som representerer antall bokstaver mellom de. Hint dukker opp samme sted som det står “Lykke til” (se figur 4.17).

```
//Gi hint når hint blir trykket
View.OnClickListener showHint = (view) -> {
    //Luke ut hint fra ord som er allerede tatt.
    //Sjekk om ordene funnet finnes i wordsRemaining og ikke i wordsFound
    //Plukk random hint fra resterende ord av wordsRemaining i fastOrdeneList
    if (!wordsRemaining.contains(wordsFound)) {
        //Velger random av de ordene som er igjen i arrayet
        Random random = new Random();
        int indeks = random.nextInt(wordsRemaining.size()); //indeksen til elementer i wordsRemaining
        String randomizedHint = wordsRemaining.get(indeks); //randomisert hint

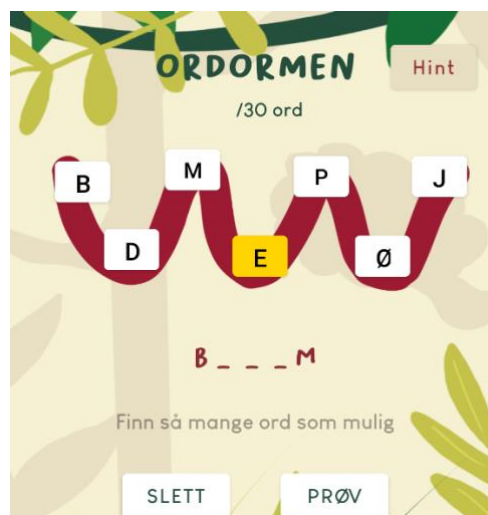
        hintWord = hintWord + randomizedHint.charAt(0);

        for (int i = 0; i < randomizedHint.length() - 2; i++) {
            hintWord = hintWord + " _ ";
        }

        hintWord = hintWord + randomizedHint.charAt(randomizedHint.length() - 1);
        feilmelding.setText(hintWord);
        //Resetter slik at neste gang hint blir trykket blir det ett og ett hint som dukker opp.
        hintWord = "";
    }

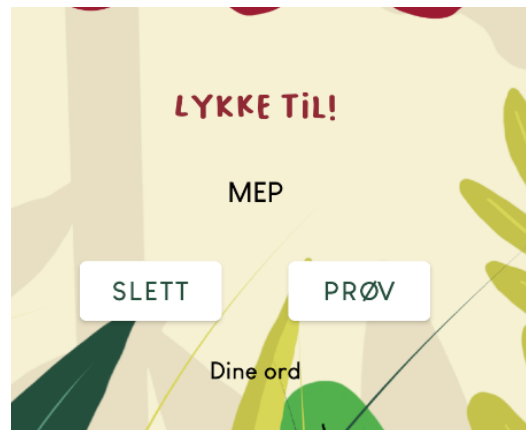
    if (wordsRemaining == null) {
        //Person alle ordene er tatt ligger alle i words found og ingen på remaining. Alltså funnet alle ordene.
        feilmelding.setText("Du har funnet alle ordene!");
    }
};
```

Figur 4.16 - Sjekk med hint



Figur 4.17 - Hint

“Slett” er kodet for å fjerne siste karakter som er skrevet i inputfeltet. Når hele ordet er slettet, resettes feltet slik at man kan påbegynne nytt ord (se figur 4.18). Konvensjonen om søppelbøtte kunne blitt brukt som ikon i stedet for ordet “slett”. Kjente konvensjoner er effektivt ved kommunikasjon gjennom UX-design, som formidler raskt et budskap og tar opp mindre plass i grensesnittet men også er forståelig på tvers av språk (Lu, Lindberg, 2022).



Figur 4.18 - Slette fra inputfeltet

Bevaring av tilstanden er sentralt for å kunne snu telefonen underveis i et spill uten at spillet resettes. Dette gjøres i Javakoden med `onSaveInstanceState` og `onRestoreInstanceState`, hvor man tar vare på elementer man ønsker blir stående ved rotasjon (se figur 4.19). I Ordomen blir ordtelleren, hint, ordlisten med funnet ord og tekstfeltet, bevart ved endring av tilstanden.

```
//Får applikasjonen til å huske tilstanden ved rotasjon og lignende, og resette seg når appen lukkes
@Override
protected void onSaveInstanceState(Bundle outState) {
    super.onSaveInstanceState(outState);

    antallOrd = (TextView) findViewById(R.id.antallOrd);
    feilmelding = (TextView) findViewById(R.id.feilmelding);

    // outState.putString("hint", hintWord);
    outState.putString("hint", feilmelding.getText().toString());
    outState.putStringArrayList("wordsFound", wordsFound);
    outState.putString("antallOrd", antallOrd.getText().toString());
}

protected void onRestoreInstanceState(@NonNull Bundle savedInstanceState){
    antallOrd = (TextView) findViewById(R.id.antallOrd);
    show = (ListView) findViewById(R.id.ordFunnet);
    feilmelding = (TextView) findViewById(R.id.feilmelding);
    inputText = (TextView) findViewById(R.id.inputText);

    hintWord = savedInstanceState.getString( key: "hint");
    wordsFound.addAll(savedInstanceState.getStringArrayList( key: "wordsFound"));
    adapter = new ArrayAdapter<String>( context: this, android.R.layout.simple_list_item_1, wordsFound);
    show.setAdapter(adapter);

    antallOrd.setText(savedInstanceState.getString( key: "antallOrd"));

    //feilmelding.setText(hintWord);
    feilmelding.setText("");
    feilmelding.setText(savedInstanceState.getString( key: "hint"));

    word = savedInstanceState.getString( key: "word");
    inputText.setText(savedInstanceState.getString( key: "word"));
}
```

Figur 4.19 - Tilstandsbevaring

Feedback til brukeren dukker opp ved trykk på “prøv”, da får man opp tilbakemeldinger som “Supert”, “Det må være minst 4 bokstaver i ordet”, “Ikke et av ordene” og “E må være med i

ordet”. Koden sjekker input opp mot alle scenarioene med for få bokstaver, ord uten at E er klikket, ikke i ordlisten osv. “Toast” er brukt til å vise tilbakemeldingene til brukeren (se figur 4.20). Toastlengden er satt til “long” slik at brukeren får tid til å få med seg innholdet før beskjedens forsvinner. En toast er valgt siden det gjør at man sparer plass ved at det er en pop-up beskjed som forsvinner, slik at beskjedens ikke er vist fram unødvendig lenge i en egen TextView (se figur 4.21).

```
//Dersom E ikke er med i ordet kommer feilmelding  
else{  
    Toast.makeText(getBaseContext(), getResources().getString(R.string.feilmelding2), Toast.LENGTH_LONG).show();  
    ((TextView) findViewById(R.id.inputText)).setText("");  
    word = "";  
}
```

Figur 4.20 - Toast



Figur 4.21 - Toast visualisering for bruker eksempel

Kilder

Bertelsen C. (2021, 13. Juni) *Hva er personas?* Inbound. <https://www.inbound.no/blogg/hva-er-personas>

Sagdahl M.S. (2019, 20. Juni) *Verdi*. SNL. <https://snl.no/verdi>

Clancy M. (2022) *Make beautiful, shareable moodboards in minutes with Milanote*. Milanote. <https://milanote.com/guide/create-better-moodboards>

Rannem Ø. (2022) *Versaler*. SNL. <https://snl.no/versaler>

Røyert H, Hanselmann E. (2022, 22. Februar) *Farger som virkemiddel i bilder*. NDLA. <https://ndla.no/nb/subject:a6b56b7e-2149-4216-92b6-3095feb870f3/topic:f740668e-def8-4a94-b1c7-de94f550678f/resource:1:112508>

Yining C. (2022, 01. Juni) *Louis George Café*. Dafont. <https://www.dafont.com/louis-george-caf.font?text=Finn+s%E5+mange+ord+som+muliq>

Locomotive. (2018, 03. Mars) *Bakso Sapi*. Dafont. <https://www.dafont.com/bakso-sapi.font?text=Finn+s%E5+mange+ord+som+muliq>

Rannem Ø. (2021, 07. November) *Grotesk skrift*. SNL. https://snl.no/grotesk_skrift

Johansen E. (2018, 07. Mai) *Lesbarhet*. NDLA. <https://ndla.no/nb/subject:1:f7d7f164-fb40-4d21-9813-6a171603281d/topic:2:172416/topic:2:186407/resource:1:179922>

Uiux Studio. (2022) *The best google font for 2022 ux webdites*. Uiux Studio. <https://www.uiux.studio/best-google-fonts.html>

Moe M. (2020, 07. Desember) *Lesbarhet*. DinTekstforfatter. <https://dintekstforfatter.no/lesbarhet/>

Johansen E. (2021, 09. Juni) *Enhetlig design*. NDLA. <https://ndla.no/nb/subject:1:00a0141d-2307-4a5a-a154-0c821449f6d2/topic:76038b07-d45d-47f9-b20c-92ac6b004222/topic:84e1f7e3-b8cc-427b-8f2b-58b6ccb6b535/resource:1:175485>

Kushwaha A. (2019, 22. August) *Choosing rounded buttons to build web interfaces*. EndlessStudio. <https://www.oodlesstudio.com/blog/why-choose-rounded-buttons/>

Uhr. (2021, 09. April) *Fonter og typografi*. Uhr. <https://www.uhr.no/om/for-presse/profilmanual/typografi/>

Sporaland I.G. (2019, 10. Mai) *Lys-mørk kontrast*. NDLA. <https://ndla.no/subject:1:00a0141d-2307-4a5a-a154-0c821449f6d2/topic:76038b07-d45d-47f9-b20c-92ac6b004222/topic:a1139e31-c1c2-424a-9ae8-aeff0ef091d4/resource:d2a81f26-ba0c-4614-9710-1c00f6c7af8f>

Lu S, Lindberg M. (2022) *Universell utforming*. Forbrukertilsynet <https://www.forbrukertilsynet.no/wp-content/uploads/2016/06/Dokumentasjon-av-Universell-utforming.pdf>

UUtilsynet. (2022) *Bruk av farger*. UUtilsynet. <https://www.uutilsynet.no/veiledning/bruk-av-farger/206>

Harchani R. (2019, 12. Juni) *Always use constraint layout for creating complex layouts*. Medium. <https://medium.com/nybles/heres-why-you-should-always-use-constraint-layout-for-creating-complex-layouts-in-android-42802f0a3b58>

Miljødirektoratet. (2022) *App-retningslinjer for miljødirektoratet*. Miljødirektoratet. <https://design.miljodirektoratet.no/designsystem-for-miljodirektoratets-apper/app-retningslinjer-for-miljodirektoratet/>

Johansen E. (2021, 14. Desember) *10 tips til design av digitale flater*. NDLA. <https://ndla.no/nb/subject:1:20bc82bc-62e3-4276-8629-84a65b8a6ad2/topic:2:540ad21d-aa94-4df0-aacb-3dfec5caeb7c/topic:b2abe783-5ec1-4126-a8e9-650e6add329e/resource:86679c00-c911-471f-9373-c0da1895884d>