

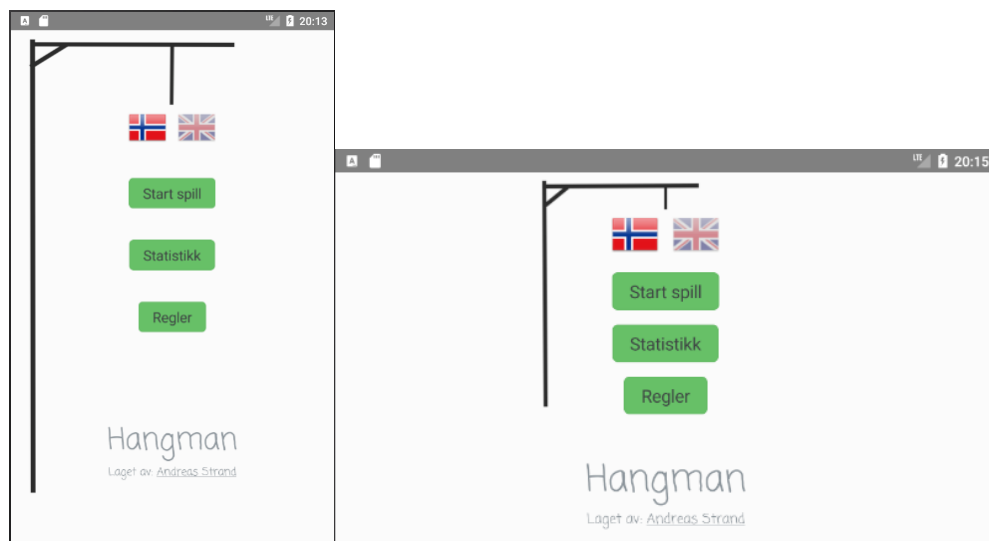
Apputvikling mappe 1 - dokumentasjon

S305036 Andreas Strand

Android utgaven av Hangman jeg har laget består av fire aktiviteter; MainActivity, GameActivity, StatisticsActivity og RulesActivity. Det er tatt til høyde for at skjermen skal kunne endre retning, så for alle .xml filer finnes det også en 'landscape'-versjon.

MainActivity

Dette er hovedaktiviteten som også representerer det første vinduet brukeren ser. Her har brukeren mulighet til å velge mellom å starte et nytt spill, se statistikk over sine tidligere spill økter, se reglene til spillet samt å endre språk. Jeg hadde først en 'spinner' for å la brukeren velge språk, men gikk senere over til å vise et flagg for hvert tilgjengelig språk – da det bare er to språk tilgjengelig.



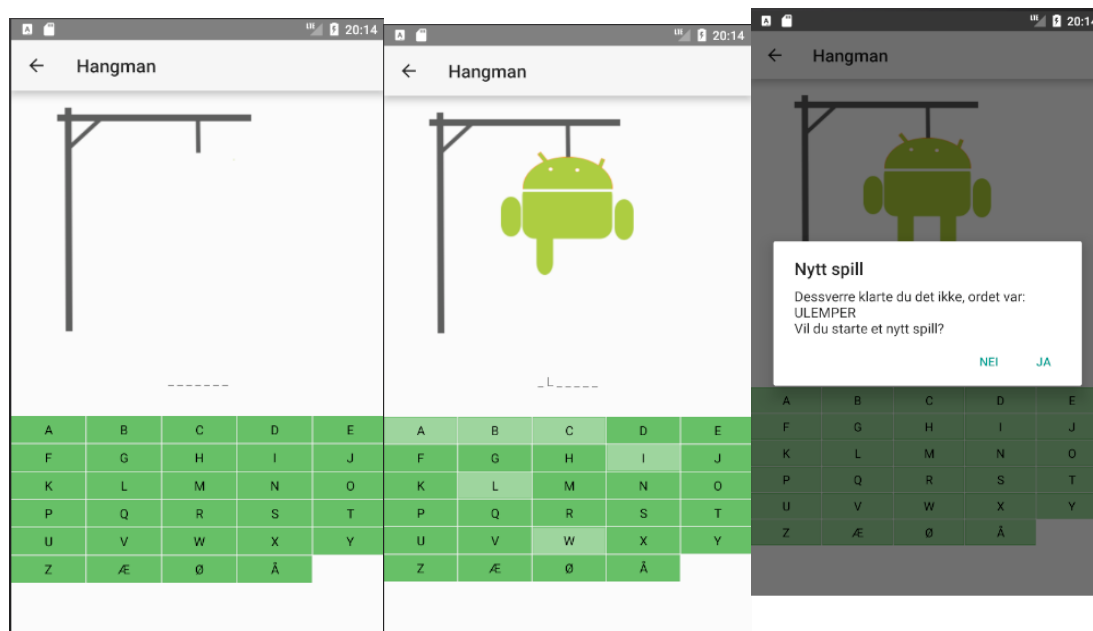
Slik ser altså MainActivity ut. Det er lagt til en galge for å gi brukeren inntrykk av at han/hun er inne i et spill – da det ville sett litt vel stusselig ut uten galgen. Om brukeren her velger det engelske flagget vil også teksten bli engelsk.

GameActivity

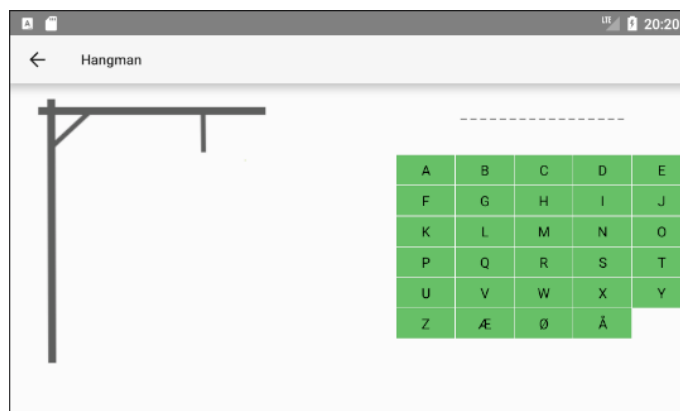
GameActivity representerer selve spill-aktiviteten. GameActivity har instanser av blant annet objekter som 'Hangman', 'MyBootstrapButton' og 'HangmanTimer'. Her blir et antall knapper representert som tilsvarende alfabetet til språket som er valgt (i strings.xml finnes det en string for hvert vilkårlig alfabet). Dette alfabetet blir kjørt gjennom den statiske klassen 'ButtonCreator' som returnerer en HashMap av en knapp og dens verdi (bokstav).

Når telefonen står har jeg valgt å ta i bruk plassen på skjermen ved å gjøre knappene så store som det er mulig for å minske sjansen for at en bruker skal trykke på feil knapp. I landscape så er knappene ca like store, bare at alle knappene ligger på høyre side – slik at brukeren skal slippe å strekke seg langt inn mot skjermen for å trykke på bokstavene som ligger innsert.

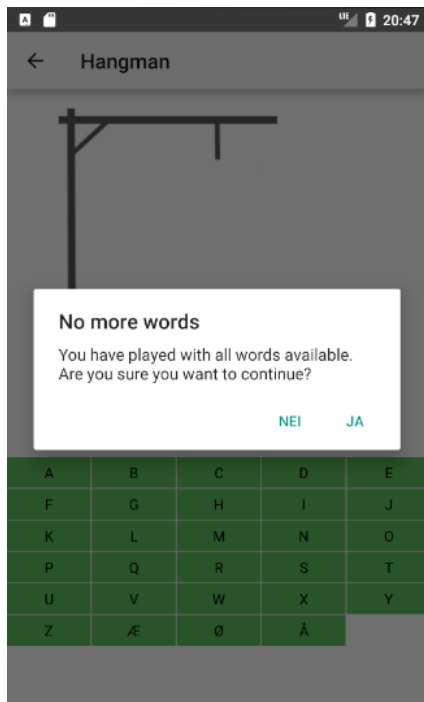
Når en bokstav er blitt sjekket blir den markert grå, og det er ikke mulighet til å trykke på denne bokstaven/knappen igjen. Slik ser GameActivity ut:



Som man ser på det siste bildet, så blir brukeren spurt om å starte et nytt spill etter spillet er ferdig. Om brukeren velger 'JA' så blir et nytt spill startet, om brukeren velger 'NEI' blir brukeren sendt tilbake til MainActivity.

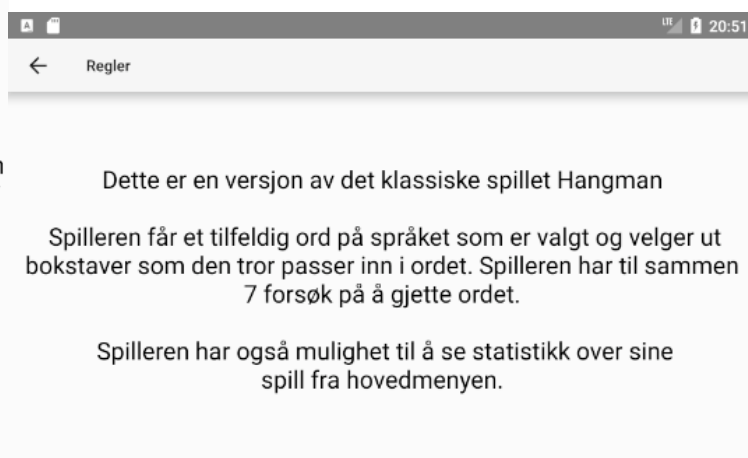
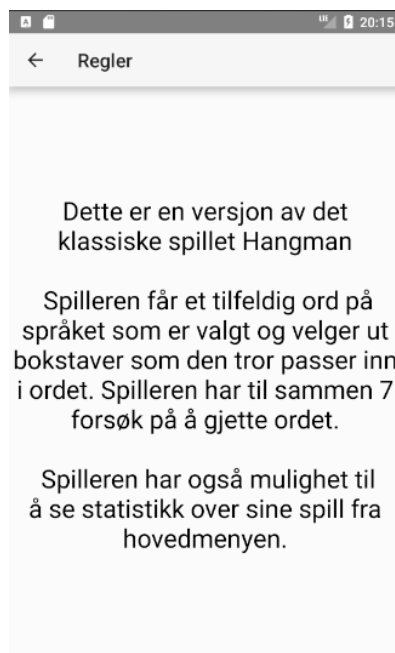


Under her ser vi også et eksempel på hva som skjer etter brukeren har spilt gjennom alle ti ordene (på et språk) på rad. Da får brukeren beskjed om at alle ordene er brukt opp – og om brukeren er sikker på om han/hun ønsker å fortsette med et nytt spill.



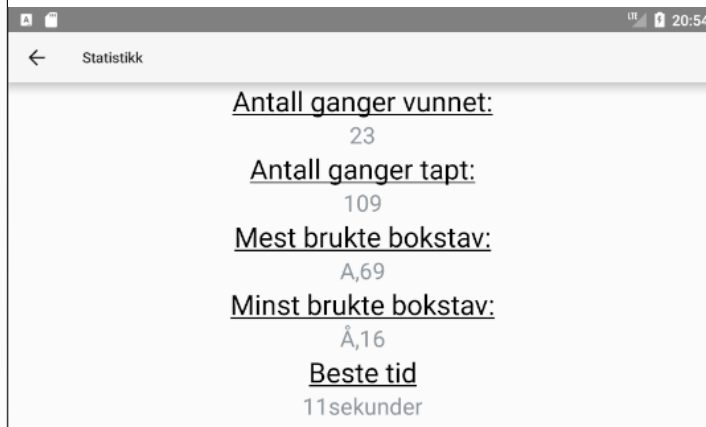
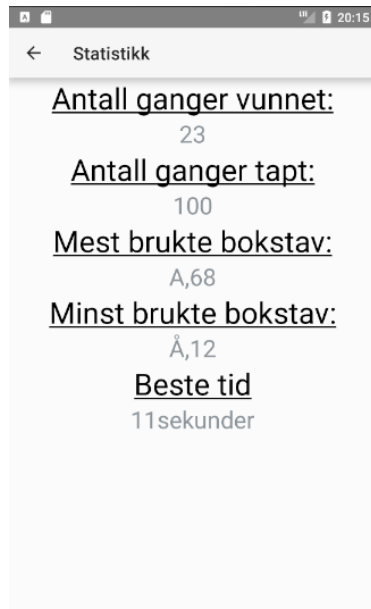
RulesActivity

RulesActivity er den minste aktiviteten i applikasjonen min. Den består kun av et TextView som viser reglene til spillet.



StatisticsActivity

Denne aktiviteten viser all statistikken til tidligere spill. Her vises antall ganger vunnet, antall ganger tapt, hvilken bokstav som er gjettet flest/færrest ganger – samt hva som er den raskeste tiden et spill er vunnet.



Måten statistikk blir lagret i spillet er gjennom å bruke SharedPreferences. En klasse 'StatisticsService' er laget med statiske metoder for å hjelpe til med å enkelt hente ut og lagre statistikk.

Design

Når det kommer til design prinsipper som er blitt brukt i appen, har jeg benyttet meg av blant annet følgende:

- **Jeg har laget** et enkelt brukergrensesnitt hvor alle elementene i et 'View' kommer klart fra. De kommer klart frem ved å bruke kontraster på elementene.
- **Kontrast på tekst** er også tenkt på. Her er blant annet w3c sine prinsipper fulgt. Det brukes svart farge på skriften på knappene i applikasjonen som har bootstrap sin 'klassiske' grønn-farge. Vanligvis bruker Bootstrap-knappene hvit skrift, men dette ville brutt prinsipper da jeg også har hvit bakgrunn.
- **'Touch targets'** – elementer i applikasjonen som kan berøres, ligger godt separert fra hverandre – med unntak i GameActivity. Her ligger bokstav-knappene inntil hverandre, men det er sørget for at det er litt mellomrom mellom hver knapp, slik at man ser hvor en knapp avslutter.
- **Jeg har ønsket** å bruke så stor skrift-størrelse som mulig uten at skal se for ekstremt ut. Dette for at brukere ikke skal ha problemer med å lese skriften. I tillegg til det har jeg brukt 'scaleable pixels' for alle visuelle tekstlige representasjoner. Dette gjør at brukere som har valgt en større skriftstørrelse generelt på Android-telefonen sin vil få se enda større størrelse på teksten enn det som er satt 'default'.
- **En navigasjons-bar** er lagt til i alle aktiviteter, slik at brukere har mulighet til å navigere tilbake til hoved-aktiviteten til enhver tid.
- **Det er lagt til rette** for skjermlesere, ved at bilder også har en tekstlig beskrivelse.
- **Gruppering av elementer** er også en viktig faktor. Dette er ikke en enorm applikasjon, men man ser for eksempel i GameActivity at hangman-bildet ligger øverst, og knappene er separert for seg god under bilder. Samme ser vi i MainActivity, alle berørlige elementer ligger sentralt på skjermbildet.
- **Uten at det** er testet på flere andre skjermer er 'density-independent' pixels (dp) brukt for alle elementer. Disse baserer seg på skjermstørrelsen og ikke direkte på antall pixler man har på skjermen. Det gjør det enklere å tilpasse flere typer skjermer.

Klasser i applikasjonen

På siste side i denne dokumentasjonen ligger en visuel oversikt over alle klasser som er laget i applikasjonen. Her følger en liste med en kort beskrivelse over alle klassene, aktiviteter er markert med fet skrift og underlinje:

MainActivity – Hovedaktivitet. Kan velge å navigere til andre applikasjoner

StatisticsActivity – Aktivitet som viser statistikk til brukeren.

StatisticsService – Klasse som tar i mot spillet sitt 'Resources' objekt og kan hente ut og lagre statistikk.

StatisticsTransfer – Et DTO (Data-Transfer-Object) som brukes innad i StatisticsService-klassen for å sende verdier frem og tilbake mellom funksjoner. Kunne strengt talt vært en inner-class av StatisticsService, men er gjort slik for enkelthetens skyld.

GameActivity – Aktivitet som representerer spill-aktiviteten.

HangmanTimer – Objekt som brukes som en stoppeklokke i GameActivity for å se hvor lang tid brukeren bruker på et spill.

MyBootstrapButton – Arver fra klassen 'BootstrapButton' (som er hentet fra et eksternt bibliotek gjennom *Gradle*). Jeg trengte å serialisere objektene mine for å bevare 'staten' til knappene når skjermen blir snudd på. Derfor er MyBootstrapButton laget – og implementerer Serializable.

ButtonCreator – Klasse som tar imot alfabet og ressurser til spillet og genererer knappene (MyBootstrapButton) som brukes i GameActivity.

Hangman – Representerer spillogikken og inneholder en instans av klassen 'Word'.

Word – Brukes som en hjelpeklasse til 'Hangman'. Var en inner-class til 'Hangman', men måtte separeres når jeg måtte serialisere objektene.

RandomWordService – Klasse med statiske metoder som inneholder en liste av brukte ord. Statisk for å kunne aksesseres fra flere steder.

RulesActivity – Aktivitet som viser reglene til spillet.

