* 1. Starten

Det var veldig nytt prosjekt som vi fikk på dette Apputvikling faget. Hangman er første appen som vi lager og det var ikke så lett å komme i gang. Første fase var å sjekke litt om hvilke regler spillet har og hva man må ha for dette spillet.

* 1. Antall screenssider

Spillet består av følgende screenssider:

1.2.1

**Menysiden**: Menyen består av fire knapper som brukeren kan bruke dem for å gå gjennom forskjellige screenssider. På bakgrunnen av det siden brukte vi et bilde av Hangman spillet som kopierte vi fra nettet. Knappene ”Spill”, ”Regler”, ”Språk” og ”Avslutt” står i menysiden.



1.2.2

**Spill:** ved hjelp av spillknappen starter spillet og brukeren går inn i nytt screenside etter at han/hun har klikket på spillknappen. Spillsiden består av en galge , en tastatur , en textview(”med strekker som skjuler ordet som man gjetter) , andre textview som (viser hvormange ganger man har vunnet eller tapt) og 2 knapper ”Avslutt” og ”Neste”.



NB!: Vi har brukt mest knapper for navigere på spillet fordi knapper er mest brukt element for navigere inn in daglige Apper.

Når man starter spillet får man et random ord for å gjette. I den første textviewen får man en del streker som er like antall bokstaver som random ordet har. Det betyr at hver strek skuler et bokstav av ordet. For å gjette ordet må man klikke på tastaturet . Riktige bokstaver vises på textviewen og feile bokstaver går bort fra tastaturet.

For å gjette et ord kan brukeren ha 6 feiler. For hver feilgjetting vises en del av kroppen på galgen. Man kan avslutte spillet ved å klikke på avslutt knappen. Knappen ”Neste” er

Ikke aktiv til man gjetter ordet eller til man gjetter 6 feil bokstaver. Hvis man gjetter hele ordet, blir knappen ”Neste” aktiv og kan man gå videre til neste ord. Hvis man gjetter 6 feile bokstaver får man en dialogvindu som sier ”Game Over” og informerer brukeren om vil man avslutte trykker ”JA” og hvis man skal til neste runde trykker ”NEI” og informasjon om hva er ordet som man feiler på. Hvis brukeren trykker Ja da går man tilbake til menysiden og hvis man trykker nei går man til samme stede der man er i, men man må klikke på knappen neste rundet.



På andre textviewen står om hvor mange ganger som man har vunnet eller har tapt. Hvis man gjetter siste ordet i listen kommer en dialogvindu som gir informasjon til brukeren om listen er tom og viser hvormange ganger som har tapt og vunnet . Brukeren kan har bare en mulighet og det er og klikke på ”OK” til å gå tilbake til hoved menyen.

1.2.2

**Regler:** etter man klikker på regler knappen i hoved menyen går man videre til screenside der man kan lese alle reglene om spillet. For å komme tilbake til hovedmenyen klikker man på knappen ”Tilbake”.



1.2.3

**Språket:** dette spillet har to språker ”Engelsk” og ”Norsk”. Brukeren kommer inn i et ny screenside der man velge det språket som man vil. På dette siden ligger to knapper med engelsk flagg på den første og norsk flagg på andre knappen. Etter at man klikker på en av knappen går man til menysiden mens hele teksten har forandret til det språket som man har valgt.



1.2.3

**Avslutt:** for å avslutte helle spillet klikke man på avslutt knappen da går man ut av spillet.

* 1. Klassediagram

LanguageActivity.class

Regler.class

MainActivity .class

valgLanguage.class

playActivity.class

Spillet har aktiviteten ”regler”,”play”,” språk” og ”language” . Language aktiviteten har en hjelpe klasse som heter ”velgLanguage”. Vi har valgt å utleie kjente og enkelte datatyper og metoder på dette spillet for at den skal være mest oversiktlig. MainActivity klassen er en hoved stasjon for alle aktivitetene. Med andre ord brukeren kan gå til alle aktivitetene gjennom MainActivity. Brukeren må komme tilbake fra en aktivitet til MainActivity for å åpne et annet aktivitet.

Mest av funksjonaliteten for dette spillet ligge i playActivity klassen. Det viktigeste metodene som er brukt i dette klassen er :

* **makeRandomWord():** Den lagger alle ordene som ligger på arraylisten i en liste og henter et ord hver gang for gjetting. Metoden er slik at den lagrer den brukte ordet i et annet arraylist for at brukte ordet skal ikke komme opp denne runden. Det betyr at brukeren får ikke gjentakelse på ordene.
* **updateMask():** det er metoden som skjuler random ordet som skal gjettes av brukeren. Metoden viser (\_)er som er like antall bokstaver til det random ordet. Det betyr hvis orde er tre bokstaver skal vises tre under strekker.
* **buttonPress(View view):** Metoden er lagret for å vise riktig bokstaver på textviewen og skjule feilbokstavene på tastaturet.
* **guess():** i guess metoden bestemmes hvordan en bruker vinner eller tapper. Her brukte vi if-else setninger for denne bestemmelsen.
* **updateImage():** vi har tatt bide av forskjellige triner av hangmanen i drawable-kategorien . Her brukte vi if-else setninger for å vise frem bildene. Det bety hver gang som brukeren gjetter en feil bokstav kalles på iden til en av bilde i drawable-kategorien.
* i metodene **onSaveInstanceState**() og **onRestoreInstanceState**() lagrer vi verdier i den første og lese vi fra den andre. De metodene brukes for at man ikke mister data eller verdi for eksempel hvis man bytter fra vertikal versjon til horisontal versjon.
* Metodene **onBackpressed()** og **congratulation()** er brukt for dialog vinduene mine. Vi har brukt AlertDialog.builder for å lage de vinduene.
  1. Design

Vi har prøvd å lage en veldig enkelt design for at brukerne kan forstå og finne frem ting lettere. Bildene som vi har brukt er tatt fra nettet. Vi var gjennom mange spiller og knapper var mest brukt på dem derfor valgte vi å bruke knapper for at brukeren navigere seg gjennom spillet.

1.5 Språk funksjonalitet

Spillet har to språker engelsk og norsk. Vi kodet det i klassen Language og en hjelpe klasse som heter languageValg. På alle onCreate metodene i aktivitetene kallet vi på metoden ”loadlang()” som liger i Language klassen for å bytte språket på teksten på alle aktivitetene.

Vi har to arraylister med 10 ord på hver liste. Det er engelsk ord på engelsk språket og norske ord på norske språket som brukeren skal gjette.

**Konklusjon : Det var veldig nyttig og viktig oppgave som hjulpet oss med å søke å lære mer programmering. Vi gjorde nesten 80% av koden selv og ca.20% ved hjelp av andre og nettet. Vi har lært mye av dette oppgaven for eksempel hvordan man fikser forholdet mellom tekst og bilde eller å lage en ting som er brukelig på mobil.**