MAPPE 1: HANGMAN  
STUDENT: S236654 Andres Stenseng Bjørnrud  
DOKUMENTASJON

**Android Design prinsipper:**

Farge valget mitt er gjort med fokus på kontraster slik at øynene fester seg lettere til objekter med betydning. Du vil se igjennom hele prosjektet at det er like farger slik at brukeren blir raskere kjent med brukergrensesnittet og ikke overasket når personen åpner ny aktivitet. På hovedsiden har vi en mørke grå bakgrunn med hvite/lysegrå objekter foran. Jeg valgte å bruke flagg for å representere muligheten for å bytte språk slik at brukeren ser det med en gang. Hadde jeg benyttet en vanelig knapp kan det hende at brukeren bare hadde flyttet blikket rett til 1 PLAYER/1 SPILLER knappen og startet spillet uten å vite om funksjonen. Om brukeren ser et norsk flagg og det er engelsk tekst vil brukeren få en forståelse om at det er flere språk valg. Dette handler om å plassere tanker i hode på brukeren så dem slipper å tenke selv. Som det skrives på nettsiden om design prinsipper, ”Pictures are faster than words”. Pluss at bilder er morsommere enn kjedelige knapper. På GameActivity er like farge valgt unntatt fargen på antall riktige og gale gjettinger. Der valgte jeg å bruke en grønn og rød farge. Slik slipper jeg å ha ekstra objekter som beskriver de 2 tallene. Grønt gir en positiv følelse og rød er negativ følelse(stort sett).

Når et spill er ferdig så passer jeg på og fortelle at brukeren enten gjettet riktig eller feil og spør om brukeren vil forsette eller gå tilbake til hovedmey. Slik blir spilleren informert og får være med på å bestemme om spillet skal fortsette.

Klasse diagram

|  |  |
| --- | --- |
| GameAcitivity | |
|  | categories: ArrayList<String> |
|  | lettersInWord: ArrayList<String> |
|  | lettersUsed: ArrayList<String> |
|  | words: String[] |
|  | language: String |
|  | word: String |
|  | category: String |
|  | numRightGuesses: int |
|  | numWrongGuesses: int |
|  | numWins: int |
|  | numLoss: int |
| + | onCreate(Bundle) : void |
| + | onSaveInstanceState(Bundle) : void |
| + | onRestoreInstanceState(Bundle): void |
| + | startHangman(): void |
| + | chooseCategory (): String |
| + | chooseWord (String) : String |
| + | updateLetters (): void |
| + | blackOutLines (): void |
| + | updateLines |
| + | updateGuessingWord(): void |
| + | guessLetter(View): void |
| + | updateWord(String): void |
| + | resetButtons(): void |
| + | resetTextViews(): void |
| + | wordGuessedRight(): void |
| + | wordGuessedWrong(): void |
| + | onBackPressed(): void |

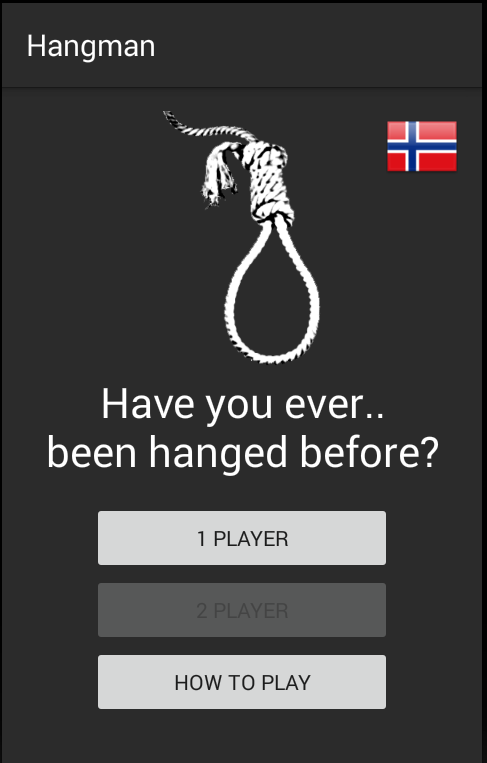
|  |  |
| --- | --- |
| RulesAcitivity | |
| + | onCreate(Bundle) : void |

|  |  |
| --- | --- |
| MainAcitivity | |
|  | NOR: String |
|  | ENG: String |
| + | onCreate(Bundle) : void |
| + | onSaveInstanceState(Bundle) : void |
| + | onRestoreInstanceState(Bundle): void |
| + | onCreateOptionsMenu(Menu): boolean |
| + | start1player(View): void |
| + | startHowToPlay(View) : void |
| + | changeLanguage(View): void |
| + | onBackPressed(): void |

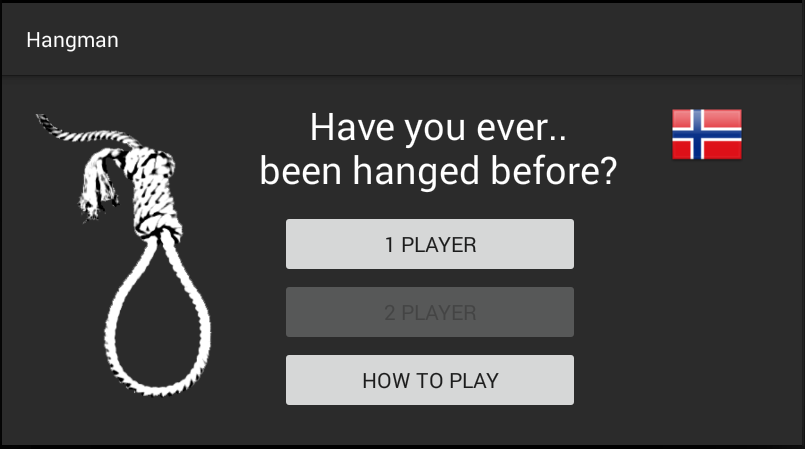
|  |  |
| --- | --- |
| LanguageOptionActivity | |
|  | LANGUAGE: String |
|  | DEFAULT\_LANGUAGE: String |
| + | changeLanguage (Bundle) : void |
| + | onSaveInstanceState(Bundle) : void |

Aktiviteter:

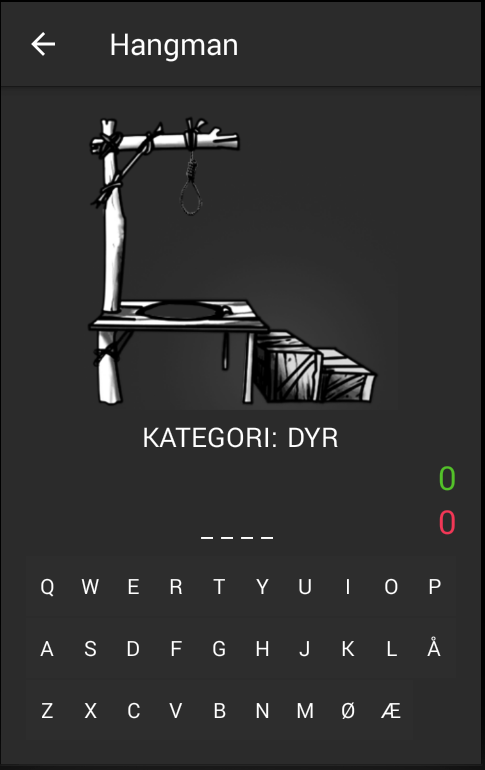
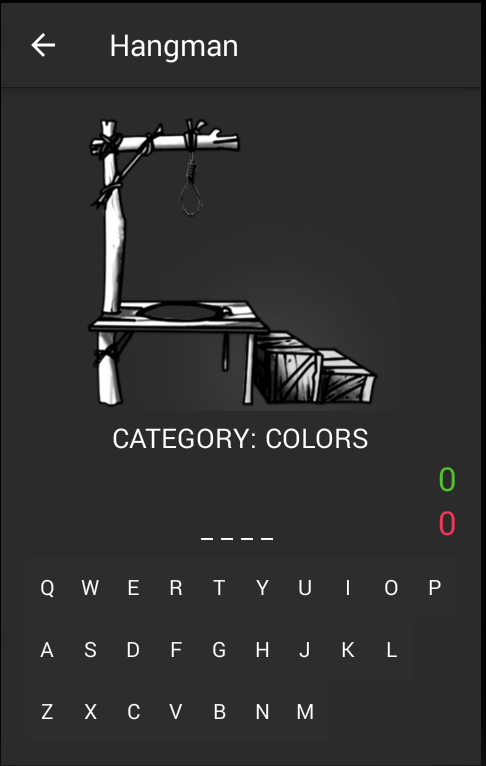
**Main\_Activity:**

****

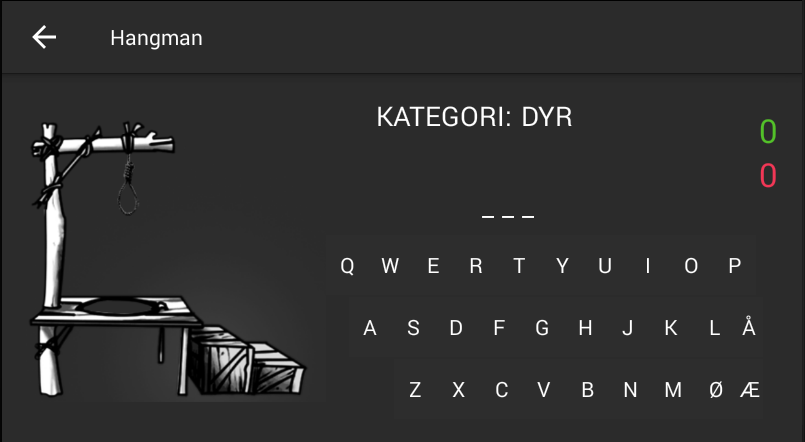
* Første skjermbilde du kommer til når du starter app-en. Aktiviteten består av en ToolBar, ImageView, ImageButton, TextView og tre Buttons
  + Toolbaren har ikke noe spesiell funksjonalitet enn at den viser navnet på appen. Grunnen til at jeg ikke har lagt noen knapper på den er fordi jeg synes slik designet nå er bedre for brukeren. Nå er det enkelt med alle mulige knapper synelig. Hadde jeg skulle implementert flere funksjoner som lydløs etc, ville jeg samlet alle knappene som angår innstillinger i appen.
  + ImageView innerholder et bilde av en knute som blir brukt når man henger folk. Er der kun for å bevise at jeg behersker og endre plassering av flere objekter i landskap modus.
  + ImageButton er en bildeknapp av flagg ut ifra hvilke språk som er tilgjengelig. Har du valgt engelsk vil de norske flagget vises. Altså er tanken at det neste språkvalget vises istedenfor det valgte språk. Ide hentet fra andre spill apper.
  + TextView inneholder en tekst som kun er der for at jeg skal kunne vise at jeg behersker språk bytte i appen ved at teksten forandres til norsk.
  + ImageButton er 3 knapper som leder til 1 spiller modus, 2 spiller modus. (Ikke tilgjengelig), men kan implementeres senere Tanken bak er at man selv kan skrive et ord som en venn kan gjette på og regler. Regler fører deg til en ny aktivitet som heter RulesActivity.
* Main\_Activity er foreldre aktivitet til alle de andre aktivitetene slik at Toolbar kan brukes som navigering. Jeg synes dette er veldig rar bruk av Toolbar ettersom Android telefoner enten har knapper på telefonen eller har en liten svart ”ToolBar” nederst som frakter deg til forrige aktivitet eller rett hjem til ”skrivebordet”. Dette skjer ved onBackPress() metoden i android biblioteket. De fleste apper du ser på Google Play butikken som bruker en Toolbar i form av navigering tror jeg kommer av at appene er laget for multiplatform og dermed ikke spesifisert for både Android og IOS ettersom IOS ikke har noe knapper som fører deg tilbake.
* Jeg vet du ville droppe hele hovedmenyen og bare gått rett inn i spillet, men der jeg er uenig. Fordi det vil virke rotete om du samler alt på en aktivitet og man burde ikke kunne endre språk midt under spillet da det vil komme i veien for ordet du holder på å gjette. Det er bare min mening.
* Kan legges i Landskap modus

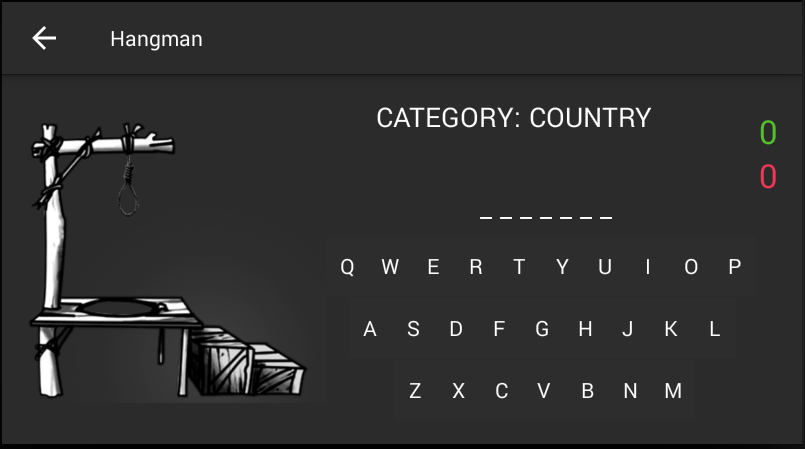
****

**GameActivity:**

****

* Skjermbilde du nå ser inneholder ToolBar, ImageView, to LinearLayout, RelativLayout og to textview.
  + Toolbar igjen blir ikke brukt til annet enn å navigere tilbake til MainActivity.
  + ImageView inneholder bilde av galgen og blir oppdatert etter at man gjetter feil. Bilde oppdateres op til 6 ganger før man har tapt.
  + LinearLayout som ligger under ImageView inneholder 2 TextView som skal vise brukeren hvilken kategori ordet ligger i. Slik blir spillet litt enklere for brukeren og kan øke spill gleden.
  + Under kategori kommer layout en ny LinearLayout som inneholder strekene som skal representere ordet bruken skal gjette på. Antall hvite linjer representerer antall bokstaver i ordet. Når man gjetter en bokstav vil linjen som tilhører den bokstaven fylles med gjettet bokstav.
  + Videre har vi en RelativLayout hvor mitt eget definerte tastatur ligger. Knappene vil forsvinne ettersom du har trykket på dem slik at brukeren ser hvilken bokstav han har allerede brukt. Om språket er satt til norsk vil knapper for Æ Ø Å også kommer opp.
  + Til slutt har vi to TextView som er plassert til høyre i skjermbilde som representerer antall riktig og galt gjettet. Slik kan brukeren sette mål om å for eksempel klarer 5 på rad.
* **Dette skjermbilde er satt til statisk portrett i AndroidManifest.xml. Dette betyr at den ikke kan gå i landskap modus og er gjort fordi jeg rett og slett ikke ser poenget med det på en telefon. Om appen skulle brukes på tablet så kunne det spilt en større rolle. Jeg gjorde et godt forsøk, men det skjedde en feil som jeg rett og slett ikke klarte å rette opp i før innleveringen. Du vil se i koden at jeg har gjort klart for onSaveInstanceState() og onRestoreInstanceState() metodene. Det som skjedde var at når jeg skulle oppdatere bokstav knappene etter å ha gått i landskap modus, med å gjøre det som alt var trykket på til INVISBLE, fikk jeg NullPointerException uten å vite hvorfor. Jeg kunne ha fikset det på en meget tungvint måte ved å ha masse IF-tester som sjekker hvilken okstaver som var INVISBLE før jeg gikk i landskaps modus, men valgte og fjerne det fordi det ville gjort koden meget uprofesjonell. Om du ønsker å teste dette kan du fjerne linjen: android:screenOriantation=”portrait”, fra manifest filen. Feilen gjør ikke at spillet krasjer, men du føres tilbake til hovedmenyen og du kan gå inn i spillet igjen og se at skjermbilde er i landskapsmodus.**





* Når aktiviteten starter kjører det en startHangman() metode nederst i onCreate() metoden. startHangman() metoden gjør dette:
  + Velger tilfeldig en kategori av 3 forskjellige(dyr, farger og land)  
    *String category = chooseCategory();*
  + Velger et tilfeldig ord innenfor kategorien (word er en String)  
    *word = chooseWord(category);*
  + Oppdatere med riktig antall hvite linjer lik antall bokstaver  
    *updateLines(word.length());*
  + Oppretter en ArrayList som skal fylles med bokstaver som er blitt brukt i gjettingen  
    *lettersUsed = new ArrayList<String>();*
  + Så splitter ordet som skal gjettes opp i bokstaver og legges inn i en ArrayList.  
    *lettersInWord = new ArrayList<String>();*

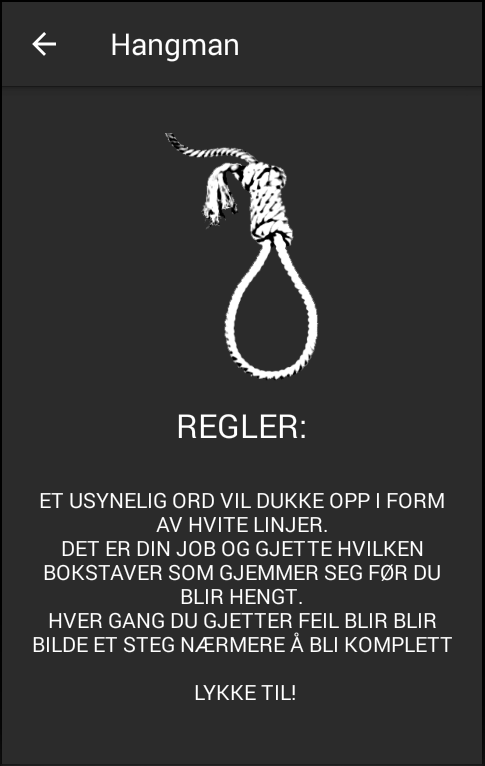
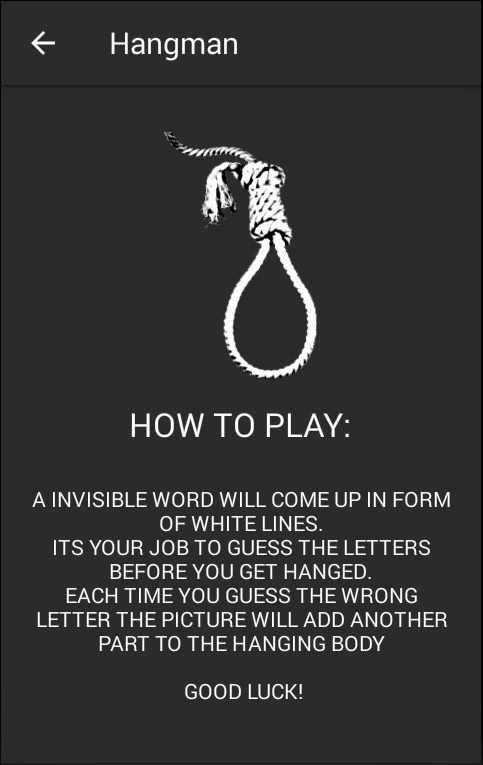
*for(String letter : word.split("")){*

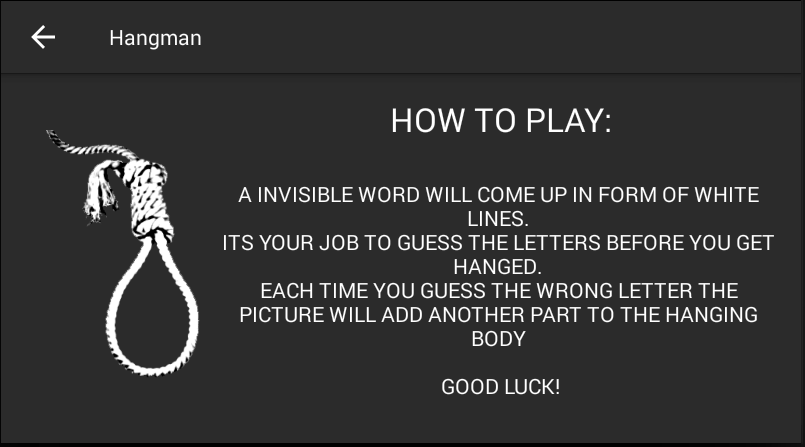
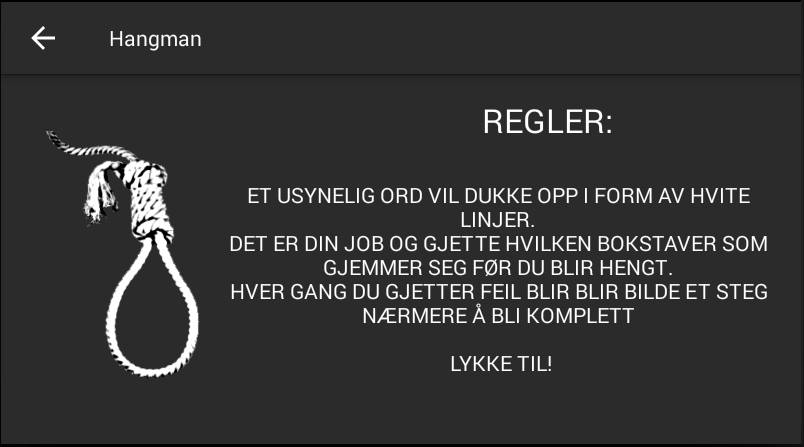
*lettersInWord.add(letter);*

*}*

* + Deretter kan man gjette på bokstaver. Når en bokstav blir trykket på vil metoden *guessLetter()* bli kjørt. Den henter taggen til knappen (som er en bokstav) og sjekker om ordet som skal gjettes inneholder den bokstaven. Hvis den gjør det så oppdateres linjene eller så oppdateres galgen om det er feil gjettet.
  + Når riktig ord er gjettet eller antall forsøk er brukt opp vil det dukke opp en alert boks som forteller deg om du har vunnet eller tapt. Du vil få to alternativer om å starte nytt spill eller gå tilbake til hovedmenyen.
  + Antall riktig gjettninger vil oppdateres med 1 om det gjettet riktig og det samme skjer om det er gjettet feil med antall feil gjettet.

**RulesActivity:**

****

* + Ikke så mye og si enn at den inneholder et ImageView og TextView.
    - Imagevire er samme bilde som er på hovedsiden. Tau.
    - TextView inneholder regler/”how to play” til spillet
  + **Kan legges i Landskap modus**

**LanguageOptionActivity:**

* Denne aktivteten har ikke noe layout fordi den har kun hensikt og kunne endre språk i appen. Den har tre metoder, *changeLanguage(String language, Context context)*, *saveLanguage(String lang, Context con)* og *loadLanguage(Context con).*
  + changeLanguage(String language, Context context) tar imot språk kode(eks. ”no” eller ”en”) og context til aktivteten som ønsker å oppdatere språket.
  + Videre kjører saveLanguage() som lagrer språk verdien i SharedPreferences. Som da kan hentes tilbake ved loadLanguafe()

**Ting jeg utelukket:**

Sitat fra oppgave beskrivelse:

*”Når et ord er benyttet en gang skal det ikke benyttes mer i denne spilleøkten.”*

Jeg valgte å la være å gjøre det slik (i håp om at jeg ikke blir straffet) fordi jeg synes man skal kunne spille uten å bli stoppet. Og siden jeg ville implementere kategorier som økte antall ord betraktelig følte jeg at det ikke var nødvendig.

Om jeg skulle gjort det ville jeg ha opprettet en ArrayList som lastet inn alle ordene fra arrays.xml, og tatt bort en etter en etter at ordet er blitt brukt.

**Til ettertanke:**

* Jeg kunne kanskje ha laget en knapp fra HowToPlay/Regler som går til GameActivity for å la brukeren starte spillet raskere uten om å gå veien igjennom hovedmenyen
* Jeg ville nok ha fjernet toolbar-en øverst og lagt til en drop-down meny for innstillinger istedenfor om flere innstillinger skulle bli implementert.

**Feil i programmet:**

* Som forklart tidelige har jeg en feil med en NullPointerException som dukker opp i GameActivity når man går fra portrett til landskaps modus.
* Noen ganger kan jeg oppleve at den ikke husker språket når den går i landskaps modus, noe som jeg synes er rart siden jeg har lagret sist brukt språk i SharedPreferences.