|  |
| --- |
| Applikasjonsutvikling mappe 3 |
| Produktdokumentasjon |
| Chess |
|  |
| **Lars Sætaberget – s181117** |

**2013-11-27**

|  |
| --- |
| En innføring i applikasjonens klasser og virkemåter |

Innholdsfortegnelse

[Klassediagram 2](#_Toc373360607)

[Kravspesifikasjon 3](#_Toc373360608)

[Prosjektinformasjon 3](#_Toc373360609)

[Redegjørelse for valg 3](#_Toc373360610)

[Portrait mode 3](#_Toc373360611)

[Uendelig tid i sjakklokkene 3](#_Toc373360612)

[En passant annotasjon 3](#_Toc373360613)

[Ubrukte data i databasen 3](#_Toc373360614)

[Mute som egen preference 4](#_Toc373360615)

[Øvrig informasjon 4](#_Toc373360616)

[Klasseoversikt 4](#_Toc373360617)

[Chesspiece 4](#_Toc373360618)

[EnPassant 4](#_Toc373360619)

[PositionHashFactory 4](#_Toc373360620)

[MainMenuActivity 5](#_Toc373360621)

[GameActivity 5](#_Toc373360622)

[ChessboardView 5](#_Toc373360623)

[Chessboard 6](#_Toc373360624)

[Preferences 6](#_Toc373360625)

[Takk til 7](#_Toc373360626)

[Lydeffekter 7](#_Toc373360627)

[Ikoner for sjakkbrikker 7](#_Toc373360628)

[Launcher ikon 7](#_Toc373360629)

[Vedlegg A – Sjakkregler 8](#_Toc373360630)

[Bevegelse 8](#_Toc373360631)

[Fange motstanderens brikker 8](#_Toc373360632)

[Restriksjoner 8](#_Toc373360633)

[Spillets slutt 9](#_Toc373360634)

[Spesialregler 9](#_Toc373360635)

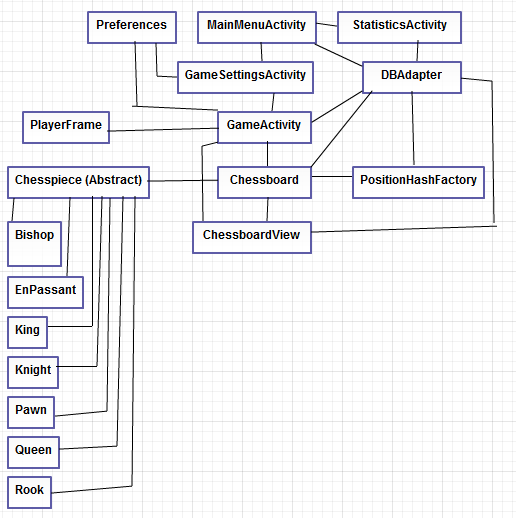
[En Passant 9](#_Toc373360636)

[Rokade 9](#_Toc373360637)

[Promotering 9](#_Toc373360638)

[Vedlegg B – Sjakkannotasjon 10](#_Toc373360639)

# Klassediagram



# Kravspesifikasjon

Applikasjonen skal kunne:

* Spille sjakk I henhold til sjakkreglene. Se vedlegg A – Sjakkregler.
* La spillet spilles med sjakklokker som kan stilles inn til ønsket tid.
* Rotere spillskjermbildet mellom hver tur for å la spillere sitte ovenfor hverandre.
* Spille av lyd ved hvert trekk og la brukeren stille inn volum ved hjelp av hardware-knappene.
* Vise relevant informasjon i spillvinduet, slik som: Navn på spillere, resterende tid for hver spiller, fargen som trekker, fangede brikker, vise lovlige trekk for valgt brikke og indikere forrige trekk.
* Lagre spilldata for fremvisning av statistikk.
* Lagre og gjenopprette en posisjon hvis brukeren forlater applikasjonen midt i et spill uten at spillet er ferdig eller avsluttet gjennom avslutt-knappen.
* Lagre og skrive ut korrekt sjakkannotasjon for spill. Se vedlegg B – Sjakkannotasjon.

# Prosjektinformasjon

## Redegjørelse for valg

### Portrait mode

Det er valgt å tvinge portrait mode for flere av skjermbildene i applikasjonen.  
Dette ble i hovedsak gjort for selve spillvinduet ettersom designet som ble valgt for å tilgjengeliggjøre all informasjonen er en ikke-triviell sak å oversette til landscape-modus.  
I dette tilfellet ble det heller valgt å legge fokus på økt funksjonalitet fremfor å tilrettelegge for landscape.

Ettersom spillvinduet ikke kan brukes i landscape, ble det valgt å utvide tvunget portrait til de fleste andre aktiviteter for å implisitt lære brukeren at landscape mode ikke er tilgjengelig i spillet.

### Uendelig tid i sjakklokkene

Det ble valgt å bare behandle 0 tid som uendelig tid når man stiller inn sjakklokkene.  
Dette er satt som default i et forsøk på å lære brukeren at 0 er en valgmulighet.  
I tillegg kan man velge “infinite” i settings-menyen som default verdi, og dette vil også bli oversatt til 0 i tidsfeltene i GameSettingsActivity. I praksis er dette noe som burde testes på brukere for å finne ut om det er lett forståelig, men større brukertester som dette er det dessverre ikke tid til i et såpass begrenset tidsrom.

### En passant annotasjon

Det er valgt å ikke annotere fanger en passant med suffikset: “e.p.” da dette er en valgfri annotasjon som ikke er i særlig stor bruk i praksis. Se vedlegg B – Sjakkannotasjon.

### Ubrukte data i databasen

Databasen lagrer data om spilldata slik som nøyaktig hvorfor et spill tok slutt, når spillet fant sted og så videre selv om applikasjonen ikke bruker disse dataene i dag. Dette er gjort fordi de er gode data å ha til relevante utvidelser. Og man kan naturlig nok ikke lagre disse datene i ettertid.

### Mute som egen preference

Mute er lagt inn som en egen innstilling i preferences-menyen slik at brukeren kan f.eks høre på musikk mens de spiller uten å bli forstyrret av lydene fra spillet.  
Dette ville ikke vært mulig ellers ettersom volumet for musikkavspilling også blir påvirket hvis brukeren stiller inn volumet mens de er inne i applikasjonen.

## Øvrig informasjon

All java kode er kommentert etter javadoc-standarden. Javadoc er generert og finnes i mappen “doc”.

# Klasseoversikt

## Chesspiece

Dette er den abstrakte klassen som spesifiserer de metodene som kreves av alle sjakkbrikke-objekter. Dette er for eksempel metoder for å bevege seg, beregne lovlige trekk og beregne om den angriper en posisjon(for å se etter sjakk). De øvrige underklassene til dette objektet untatt EnPassant vil ikke bli diskutert i dette dokumentet da de ikke er spesielt ulike med unntak av hvilke områder rundt seg de leter etter lovlige trekk.

## EnPassant

Dette er et objekt som er skapt for å la sjakkbrettet ha en enkel måte å holde styr på en passant-muligheter på. Når en bonde beveger seg to hakk vil det dannes et EnPassant objekt bak den slik at motstanderen kan fange den i løpet av neste trekk. Hvis den blir fanget vil den ta med seg bonden som skapte den. Hvis ikke blir den bare fjernet fra sjakkbrettet i henhold til sjakkreglene(Vedlegg A).

## PositionHashFactory

Denne klassen har tre arbeidsoppgaver:

1. Hashe posisjoner for å kontrollere uavgjort på grunn av repetisjon.
2. Opprette og lagre sjakkannotasjoner basert på data om trekket.
3. Gjenopprette posisjoner når et spill har blitt lukket og åpnes igjen fra hovedmenyen.

Hashingen er relativt ukomplisert: En hash starter med å definere hvilken spiller som har trekket, deretter går den gjennom sjakkbrettet og oversetter hver posisjons innhold til en hash-verdi som er unik for den typen innhold(farge og brikke). Etter at hashen er utført sammenlignes den nye hashen med de gamle hashene for å se om den har oppstått to ganger tidligere. Hvis den har det er spillet uavgjort.

Opprettelsen av sjakkannotasjoner er mye kode, men er også forholdsvis ukomplisert.  
Med alle data lagt til rette er det bare å følge oppskriften i Vedlegg B.

Gjenopprettelsen av posisjoner utføres ved å oversette annotasjonene tilbake til data for å utføre trekket. (Notis til sensor: dette kan gå noe tregt i emulatorer hvis det er mange trekk, men fungerer fint på faktiske enheter).

## MainMenuActivity

Aktiviteten som representerer hovedmenyen.  
En ganske rett frem klasse, men en liten ting er det verdt å merke seg:  
Ved trykk på PLAY-knappen vil en av to ting skje:

1. Hvis en posisjon finnes i databasen vil denne lastes inn og brukeren kastes rett inn i spillet.
2. En posisjon finnes ikke og brukeren blir sendt til GameSettingsActivity

For å få starte et nytt spill mens et annet spill pågår må brukeren avslutte det gjeldende spillet.  
Enten ved å fullføre eller ved å trykke “End” eller “End Game”, avhengig av skjermbredden.

Dette er nok en ting som burde testes på en mengde brukere for å se om det er intuitivt, men som sagt er det vanskelig å få til slike større tester i et tidsomfang som i denne oppgaven.

## GameActivity

Denne aktiviteten er hvor selve spillet finner sted.  
Den inneholder selve ChessboardViewet, og har ansvar for kommunikasjon mellom spillet og PlayerFramene. Denne klassen styrer også noe mengde innstillinger. Blant annet er det hele viewet i denne klassen som roteres hvis det er enabled. I tillegg har klassen ansvaret for å stille inn riktige verdier på klokkene etter at en posisjon har blitt gjenopprettet ved avbrutt spill (en oppgave som er langt mindre triviell enn man kanskje skulle tro på grunn av race conditions med tråden som klokken kjører i).

Klassen holder også på ikonene for sjakkbrikkene i minnet, da disse er brukt både i ChessboardView og i PlayerFrames.

## ChessboardView

ChessboardView, som teknisk sett er en Layout, er klassen som styrer visuelle hendelser med sjakkbrettet. Selve brettet er visuelt laget av 64 ImageButtons. Der background er satt til fargen som skal representeres i det området, og bildet er satt til ikonet som representerer brikken som befinner seg i den gitte posisjonen i det underliggende Chessboardet.

Klassen vil også farge knappene noe annerledes ved spesielle hendelser:

* Feltene som inngikk i det forrige trekket farges oransje
* Feltet under den valgte brikken farges rød
* Feltene som er lovlige trekk for den valgte brikken farges grønne

I tillegg håndterer klassen dialogen som oppstår når spillet slutter.  
Og har som ansvar å tilrettelegge for et nytt spill hvis rematch blir valgt.

I tillegg er det denne klassen som spiller av lydeffekten når en brikke beveger seg.

## 

## Chessboard

Chessboard er arbeidshesten i applikasjonen. Klassen er på mange måten kjernen i selve spillet, og den utfører de fleste kritiske backend-operasjonene.

Hovedoppgaven til denne klassen er å opprettholde den logiske statusen til sjakkbrettet.  
Dette betyr at den må sørge for at brikker kan bevege seg til ulike deler av brettet, og må bistå med å kontrollere at et trekk er lovlig, at kongen for brikkens farge ikke havner i sjakk etter et gitt trekk og lignende. I tillegg inneholder klassen mange hjelpemetoder for å finne visse brikker under visse omstendigheter, som blant annet brukes i generering av sjakkannotasjoner, gjenoppbygging av posisjoner og mer.

Denne klassen holder også styr på den logiske delen av sjakklokken.  
Det er i denne klassen at tråden for sjakklokken kjører, og det er tidsverdiene her som eventuelt kan trigge et tap på grunn av at tiden går ut.

Dette er også klassen som sender data til PositionHashFactory for hashing av posisjoner og for generering av sjakkannotasjoner.

I tillegg er det denne klassen som kontrollerer om spillet skal avsluttes på grunn av logiske endelser:

* Sjakk matt
* Stalemate

Her er det også en tellevariabel som kontrollerer om knappmodusen til uavgjort-knappen skal endres til å kreve uavgjort i henhold til 50-trekk regelen (se Vedlegg A) eller om den eventuelt skal endres tilbake til å tilby uavgjort til motstanderen.

## Preferences

Preferences er aktiviteten som lar brukeren endre på instillinger i applikasjonen.

Her kan brukeren velge default verdier for GameSettingsActivity, om skjermbildet i spillet skal roteres etter hvert trekk, om skjermen skal forbli på i GameActivity og om lydeffekter skal mutes.

En spesialtilpasning som er gjort er at den underliggende PreferenceFragmenten i denne klassen (som fungerer som en arbeider). Har en lytter som endrer summary teksten til visse av feltene for å representere den valgte verdien. Dette er gjort fordi det ikke er noen innebygget mulighet for dette i android APIet, og det virker som en logisk funksjonalitet for at brukeren enkelt skal kunne se hvilke verdier som er valgt for de ulike innstillingene.

# Takk til

Lydeffekter  
<http://www.freesfx.co.uk>

Ikoner for sjakkbrikker  
<http://www.iconsdb.com>

Launcher ikon  
<http://www.findicons.com>

Alle disse ressursjene er gratis å bruke til kommersielt bruk med unntak av launcher-ikonet, som kun er gratis å bruke til ikke-kommersielt bruk.

Spesiell takk rettes også til Kristoffer Sørensen som gjorde noen redigeringer på ikonene for ikonene for sjakkbrikkene. Disse ikonene tilhører CC0 1.0 Universal Public Domain Dedication, som betyr at hvem som helst kan redigere de og bruke de til kommersielt eller ikke-kommersielt bruk.

# Vedlegg A – Sjakkregler

1. Den som styrer de hvite brikkene begynner alltid.

Bevegelse  
  
1. Bonden beveger seg rett frem 1 hakk. Kan bevege seg 2 hakk hvis den ikke har beveget seg før.

2. Springeren beveger seg 2 hakk opp/ned og så 1 til venstre/høyre eller 2 hakk til venstre/høyre så 1 opp/ned. Springeren ignorerer alle brikker som står i veien med unntak om en egen brikke står der den prøver å bevege seg til.

3. Løperen beveger seg diagonalt i hvilken som helst retning så langt den vil helt til den blokkeres.

4. Tårnet beveger seg rett frem i hvilken som helst retning så langt den vil helt til den blokkeres.

5. Damen beveger seg som en kombinasjon av tårn og løper.

6. Kongen beveger seg 1 hakk i hvilken som helst retning.

Fange motstanderens brikker  
  
1. Man kan fjerne en av motstanderen sine brikker ved å bevege en av dine brikker til ruten som mostanderen sin brikke står på.

2. Bonden kan ikke fange brikker ved å bevege seg vanlig, men fanger ved å bevege seg 1 hakk diagonalt fremover til en rute som har en av mostanderen sine brikker i seg.

Restriksjoner  
  
1. Ingen brikker kan bevege seg gjennom andre brikker(untatt springeren).

2. Hvis en av din motstanders brikker er i en posisjon hvor den kunne ha fanget din konge om det var motstanderen sin tur er du "i sjakk".

3. Du kan aldri ende turen din i sjakk.

Spillets slutt  
  
1. Hvis du er i sjakk og ikke har noen lovlige trekk er det "sjakk matt" og du har tapt spillet.

2. Hvis du IKKE er i sjakk, men fortsatt ikke har noen lovlige trekk er det uavgjort.

3. Hvis det har gått 50 trekk uten at en bonde har beveget seg og ingen brikker har blitt fanget kan spilleren som har turen kreve uavgjort.

4. Hvis samme posisjon oppstår 3 ganger i løpet av et spill med samme spiller som har trekket er det uavgjort.

5. Spillerene kan når som helst bli enige om uavgjort.

6. En spiller kan gi opp når han/hun har trekket, og gi seieren til motstanderen sin.

7. Hvis man spiller med klokker: Tiden for din klokke løper bare på din tur. Hvis tiden går ut har du tapt spillet.

Spesialregler

### En Passant

Hvis en bonde har beveget seg 2 hakk kan motstanderen fange den med sine egne bønder i løpet av neste tur som om den bare hadde flyttet seg 1 hakk.

### Rokade

Hvis en konge har fri bane til et av sine egne tårn, og verken kongen eller tårnet har beveget seg OG kongen ikke er i sjakk og ikke hadde havnet i sjakk ved å flytte seg til hakket ved siden av seg kan kongen flytte seg 2 hakk mot det gjeldende tårnet mens tårnet flytter seg til ruten ved siden av kongen(på motsatt side).

### Promotering

Hvis en bonde når motsatt side av brettet skal spilleren som kontrollerer bonden umiddelbart erstatte den med fritt valg av: Dame, tårn, springer eller hest.

# Vedlegg B – Sjakkannotasjon

Brettet er delt inn i rader som listes opp fra topp til bunn: 8-1, og kolonner fra venstre mot høyre:   
a-h. (Hvis sett fra hvitt sitt perspektiv).

Første bokstav skal i tilfellet hvor brikken ikke var en bonde noteres med hva slags brikke det var etter disse retningslinjene:

* Springeren annoteres med ‘N’
* Løperen annoteres med ‘B’
* Tårnet annoteres med ‘R’
* Damen annoteres med ‘Q’
* Kongen annoteres med ‘K’

Bønder har ingen slik bokstav, men gjenkjennes ved at ingen slik bokstav skrives.

For eksempel vil et trekk med springer til e4 noteres slik: Ne4.  
Mens en bonde som utfører samme trekk vil noteres: e4.

Neste bokstav kan variere. Hvis det er mulig for en annen brikke av samme type og farge å nå feltet som det ble beveget til skal denne bokstaven eller tallet indikere raden eller kolonnen som brikken befant seg på. Det er viktig at det velges riktig av rad eller kolonne her for å gjøre det entydig bestemt hvilken brikke som beveget seg.

For eksempel hvis et hvitt tårn står på a7 og det andre tårnet står på h7 med ingen brikker i mellom, og h7-tårnet beveger seg til d7 skal det annoteres slik: Rhd7.

Eller en på h8 og en på h1: R8h6.

Etter de identifiserende tegnene, hvis en brikke ble fanget dette trekket skal det neste tegnet være x.

I tilfelle en bonde fanget, skal første tegn være kolonnen den befant seg på, så x, så posisjonen den beveget seg til.

For eksempel bonde: exd5.  
Eller dame: Qxd5.

Hvis en bonde ender sitt trekk med promotering skal det umiddelbart etter den nye posisjonen indikeres hvilken brikke det ble promotert til.

For eksempel: exd8Q for promotering til dame eller exd8N for promotering til springer.

Hvis en spiller gjør rokade med tårnet nærmest kongen annoteres det: O-O.  
Eller med tårnet lengst unna kongen: O-O-O.

Hvis et trekk ender i sjakk skal siste tegn være +.  
For eksempel: Qxd8+.

Hvis et trekk ender i sjakk matt skal siste tegn være #.  
For eksempel: Rxd7#.