Spesifikasjoner

# Om dokumentet

## Mål med dokumentet

*Dette dokumentet er ment for å formidle hvordan applikasjonen skal oppføre seg, hvilke funksjonaliteter som må implementeres for å oppnå dette, og hvordan disse funksjonalitetene blir implementert.*

*Kravene forbedres og refaktoreres etter hvert som applikasjonen vokser. Ikke tenk på dette dokumentet som et "ferdig" dokument.*

## Dokumentets mulige forbedringer

* Splitt opp kravene for å unngå flere brukerhistorier unødvendig på samme krav.

Innhold

[Om dokumentet 1](#_Toc67664266)

[Mål med dokumentet 1](#_Toc67664267)

[Dokumentets mulige forbedringer 1](#_Toc67664268)

[Klassebeskrivelser: 2](#_Toc67664269)

[Om definisjon av krav 3](#_Toc67664270)

[Formatet krav skrives på: 3](#_Toc67664271)

[Spillbrett 3](#_Toc67664272)

[KRAV: Vise et spillbrett 3](#_Toc67664273)

[Brukerhistorie: 3](#_Toc67664274)

[Akseptansekriterier: 3](#_Toc67664275)

[Arbeidsoppgaver: 3](#_Toc67664276)

[KRAV: Vise brikke på spillbrett 3](#_Toc67664277)

[Brukerhistorie: 3](#_Toc67664278)

[Akseptansekriterier: 4](#_Toc67664279)

[Arbeidsoppgaver: 4](#_Toc67664280)

[KRAV: Flytte brikke 4](#_Toc67664281)

[Brukerhistorie: 4](#_Toc67664282)

[Akseptansekriterier: 4](#_Toc67664283)

[Arbeidsoppgaver: 4](#_Toc67664284)

[KRAV: Robot besøker flagg 4](#_Toc67664285)

[Brukerhistorie: 4](#_Toc67664286)

[Akseptansekriterier: 4](#_Toc67664287)

[Arbeidsoppgaver: 4](#_Toc67664288)

[KRAV: Robot vinner ved å besøke flagg 5](#_Toc67664289)

[Brukerhistorie: 5](#_Toc67664290)

[Akseptansekriterier: 5](#_Toc67664291)

[Arbeidsoppgaver: 5](#_Toc67664292)

[Kort 5](#_Toc67664293)

[KRAV: Dele ut kort 5](#_Toc67664294)

[Brukerhistorie: 5](#_Toc67664295)

[Akseptansekriterier: 5](#_Toc67664296)

[Arbeidsoppgaver: 5](#_Toc67664297)

[KRAV: Velge 5 kort 6](#_Toc67664298)

[Brukerhistorie: 6](#_Toc67664299)

[Akseptansekriterier: 6](#_Toc67664300)

[Arbeidsoppgaver: 6](#_Toc67664301)

[KRAV: Bevege robot ut fra valgte kort 6](#_Toc67664302)

[Brukerhistorie: 6](#_Toc67664303)

[Akseptansekriterier: 6](#_Toc67664304)

[Arbeidsoppgaver: 6](#_Toc67664305)

[GUI 6](#_Toc67664306)

[KRAV: GUI til meny 6](#_Toc67664307)

[Brukerhistorie: 7](#_Toc67664308)

[Akseptansekriterier: 7](#_Toc67664309)

[Arbeidsoppgaver: 7](#_Toc67664310)

[KRAV: GUI til game som håndterer kort 7](#_Toc67664311)

[Brukerhistorie: 7](#_Toc67664312)

[Akseptansekriterier: 7](#_Toc67664313)

[Arbeidsoppgaver: 7](#_Toc67664314)

[KRAV: Options meny for å skifte innstillinger 7](#_Toc67664315)

[Brukerhistorie: 7](#_Toc67664316)

[Akseptansekriterier: 7](#_Toc67664317)

[Arbeidsoppgaver: 7](#_Toc67664318)

[Multiplayer 8](#_Toc67664319)

[KRAV: Spille fra flere maskiner 8](#_Toc67664320)

[Brukerhistorie: 8](#_Toc67664321)

[Akseptansekriterier: 8](#_Toc67664322)

[Arbeidsoppgaver: 8](#_Toc67664323)

[KRAV: Hoste et spill 8](#_Toc67664324)

[Brukerhistorie: 8](#_Toc67664325)

[Akseptansekriterier: 8](#_Toc67664326)

[Arbeidsoppgaver: 8](#_Toc67664327)

[KRAV: Joine et spill 8](#_Toc67664328)

[Brukerhistorie: 8](#_Toc67664329)

[Akseptansekriterier: 8](#_Toc67664330)

[Arbeidsoppgaver: 9](#_Toc67664331)

# Klassebeskrivelser:

*I kravene refererer vi til klasser, her skriver vi informasjon om de mest sentrale klassene.*

* GameRunner: Koordinerer input og backend med frontend.
* GameScreen: Tegner in-game GUI. (Brett deriblant)
* RRApplication: Baseklassen for spillet. Fundamentet.
* Map: Kommuniserer valgt brett og viktig brett-informasjon.
* Grid: Den logiske representasjonen av spillbrettet og elementer der.

# Om definisjon av krav

*Kravene til applikasjonen.*

### Formatet krav skrives på:

1. Brukerhistorie: Ønsket oppførsel, grunnlag for å ha krav i programmet og hvem som har bruk for det.
2. Akseptansekriterier: Hvilke delegenskaper programmet må ha for å oppfylle brukerhistorien.
3. Arbeidsoppgaver: Klasser og metoder som brukes for å implementere de ønskede egenskapene til kravet. Bryter ned krav til deloppgaver.

# Spillbrett

## KRAV: Vise et spillbrett

### Brukerhistorie:

Som en spiller ønsker jeg å se spillbrettet for å orientere meg.

### Akseptansekriterier:

* Når man velger spill fra menyen, skal spillbrettet vises på skjermen.

### Arbeidsoppgaver:

*Klasser som brukes for å oppfylle akseptansekriteriene er:*

* GameRunner, GameScreen, RRApplication, Map.

*Med dem så må man:*

* Hente informasjon om hvilket kart man bruker.
* Hente informasjon om elementene på kartet (brikker, annet).
* Tegne dette på kartet.

*Sentrale metoder er:*

public String getMapPath() // I map.

## KRAV: Vise brikke på spillbrett

### Brukerhistorie:

Som en spiller ønsker jeg å se egen brikke på bretter for å vite hvor jeg er.

### Akseptansekriterier:

* Alle skal kunne se brikken til spilleren på spillebrettet.

### Arbeidsoppgaver:

*Klasser som brukes for å oppfylle akseptansekriteriene er:*

* GameRunner, GameScreen, RRApplication, Map.

*Med dem så må man:*

* Hente spillerposisjon (GameRunner).
* Tegne spiller på kartet (GameScreen).

## KRAV: Flytte brikke

### Brukerhistorie:

Som en utvikler ønsker jeg å kunne bevege spillebrikken i spillet med piltaster for å teste brett funksjoner lettere

### Akseptansekriterier:

* Når en piltast trykkes beveges spillebrikken i en retning.
* Hvis spilleren trykker høyre, vil robotens x-verdi øke med 1.
* Hvis spilleren trykker venstre. vil robotens x-verdi synke med 1.
* Hvis spilleren trykker opp, vil robotens y-verdi øke med 1.
* Hvis spilleren trykker ned, vil robotens y-verdi synke med 1.

### Arbeidsoppgaver:

*Klasser som brukes for å oppfylle akseptansekriteriene er:*

* GameRunner: Koordinerer input og backend med frontend.
* GameScreen: Tegner in-game GUI. (Brett deriblant)
* RRApplication: Baseklassen for spillet. Fundamentet.
* Game: Holder Grid, informasjon om spillet og annen funksjonalitet.

*Med dem så må man:*

* Ta input og kommunisere til backend. (GameScreen og GameRunner).
* Oppdatere posisjoner (Game -> Grid).
* Tegne spiller på kartet (GameScreen).

## KRAV: Robot besøker flagg

### Brukerhistorie:

Som robot vil jeg besøke flagg for å få poeng.

### Akseptansekriterier:

* Gitt at en robot passere et flaggs koordinater, skal roboten kåres som vinner om flagget er det siste i sekvensen av flagg man må passere for å vinne.
* Gitt at robot lever og passerer et flagg (i riktig rekkefølge), skal en flagg-counter økes.

### Arbeidsoppgaver:

*Klasser som brukes for å oppfylle akseptansekriteriene er:*

* Robot, Flag, Game, Grid.

*Med dem så må man:*

* *Registrere om en robot passerer et flagg om flagget er neste i rekkefølgen.*
* *Sjekke om robot har vunnet (passert alle flaggene).*

*Sentrale metoder er:*

## KRAV: Robot vinner ved å besøke flagg

### Brukerhistorie:

Som robot vil jeg vinne ved å besøke alle flaggene i riktig rekkefølge for at spillet skal ha et mål.

### Akseptansekriterier:

* Gitt at en robot har passert flaggene i riktig rekkefølge skal roboten vinne når den når flagg .
* Roboten har en liste over flagg den har besøkt i riktig rekkefølge

### Arbeidsoppgaver:

*Klasser som brukes for å oppfylle akseptansekriteriene er:*

* Flag, Robot, Game, Grid.

*Med dem så må man:*

* Sjekke om robot har vunnet.

*Sentrale metoder er:*

checkIfLocalHasWonOrLost()

# Kort

## KRAV: Dele ut kort

### Brukerhistorie:

Som et spill vil jeg at spillerne skal få utdelt tilfeldige kort for å bevege robotene sine.

Som et spill vil jeg at kortstokken tømmes etter hvert som spillerne får kort for at spillerne ikke får alle de samme kortene. For å kunne dele ut kort må spillet først ha en kortstokk og minst en spiller som kan motta kort. Dette skjer automatisk som rundene går.

### Akseptansekriterier:

* Gitt at kort har blitt delt ut, skal en spiller ha en liste med mottatt kort.

### Arbeidsoppgaver:

*Klasser som brukes for å oppfylle akseptansekriteriene er:*

* CardDealer
* CardDeck
* Player

*Med dem så må man:*

* Instansiere en ny deck-list i Player, og kalle CardDealer sin deal metode.

*Sentrale metoder er:*

## KRAV: Velge 5 kort

### Brukerhistorie:

Som en spiller vil jeg kunne velge hvilke 5 kort som skal spilles i ønskelig rekkefølge.

Som et spill vil jeg ikke at spillerne skal ha mer enn 5 kort slik at en spiller ikke kan ha alle kortene samtidig.

### Akseptansekriterier:

* Gitt at spilleren velger et kort, skal det bli lagt til i sin egen hånd-liste.
* Gitt at spilleren holder 5 kort, kan han ikke velge flere.
* Gitt at spilleren velger kort, skal kortet fra “9-kort” listen forsvinne, så man ikke kan ta duplikater.

### Arbeidsoppgaver:

*Klasser som brukes for å oppfylle akseptansekriteriene er:*

* CardDealer, Player

*Med dem så må man:*

* Kalle på CardDealer’s dealCards
* Player må ha en liste for valgte og mulige kort.

*Sentrale metoder er:*

## KRAV: Bevege robot ut fra valgte kort

### Brukerhistorie:

Som en spiller ønsker jeg å gi instrukser til roboten med kortene mine, for at den skal bevege seg.

### Akseptansekriterier:

* Roboten flyttes etter kortets instrukser

### Arbeidsoppgaver:

*Klasser som brukes for å oppfylle akseptansekriteriene er:*

* Robot, Grid, Card-package.

*Med dem så må man:*

* Robot sin posisjon eller retning i Grid må endres ved hjelp av metoden i TurnCard og StepCard.

*Sentrale metoder er:*

# GUI

## KRAV: GUI til meny

### Brukerhistorie:

Som spiller vil jeg kunne se et grafisk brukergrensesnitt, der jeg kan navigere ved hjelp av piltaster eller peker som gjør opplevelsen mer brukervennlig.

### Akseptansekriterier:

* Gitt at man velger på en av knappene, skal input bli registrert og du skal få tydelig feedback på knappen funker.

### Arbeidsoppgaver:

*Klasser som brukes for å oppfylle akseptansekriteriene er:*

* Applistener, MainMenuScreen

*Med dem så må man:*

*Sentrale metoder er:*

## KRAV: GUI til game som håndterer kort

### Brukerhistorie:

Som spiller ønsker jeg å kunne se kortene jeg har blitt tildelt og ha tilgang til å velge kort.

### Akseptansekriterier:

* Gitt at jeg velger kort i en viss rekkefølge, skal funksjonene bli eksekvert i riktig rekkefølge.
* Det skal ikke gå an å velge samme kort flere ganger.

### Arbeidsoppgaver:

*Klasser som brukes for å oppfylle akseptansekriteriene er:*

*Med dem så må man:*

*Sentrale metoder er:*

## KRAV: Options meny for å skifte innstillinger

### Brukerhistorie:

Som spiller vil jeg ha tilgang til innstillinger som å skifte lyd og display mode.

### Akseptansekriterier:

* Fullskjerm-knapp skifter bildet.
* Mute demper musikk.

### Arbeidsoppgaver:

*Klasser som brukes for å oppfylle akseptansekriteriene er:*

* OptionsScreen

*Med dem så må man:*

*Sentrale metoder er:*

# Multiplayer

## KRAV: Spille fra flere maskiner

### Brukerhistorie:

Som spill-server vil jeg ha oversikt over alle spillers kort og roboter, og kunne sende informasjon til klienter om hvordan brettet ser ut og spillere befinner seg.

### Akseptansekriterier:

* Gitt av en spiller beveger seg på sin skjerm, skal det samme skje på andre brukers skjerm.
* Gitt at en spiller vinner vil alle få beskjed.

### Arbeidsoppgaver:

*Klasser som brukes for å oppfylle akseptansekriteriene er:*

* Client
* Server
* Game

*Med dem så må man:*

* Få alt i spillet til å snakke sammen!

*Sentrale metoder er:*

## KRAV: Hoste et spill

### Brukerhistorie:

Som en spiller ønsker jeg å kunne sette opp et spill hvor andre kan bli med.

### Akseptansekriterier:

* Når jeg hoste et spill startes, og andre skal ha tilgang til å koble seg til.

### Arbeidsoppgaver:

*Klasser som brukes for å oppfylle akseptansekriteriene er:*

* Server

*Med dem så må man:*

* *Starte server.*
* *Sende spill-informasjon.*

*Sentrale metoder er:*

## KRAV: Joine et spill

### Brukerhistorie:

Som spiller vil jeg kunne trykke Join i Multiplayer menyen for å bli med i et spill som en annen person hoster.

### Akseptansekriterier:

* Multiplayer meny må ha funksjoner til å kobles til en server.

### Arbeidsoppgaver:

*Klasser som brukes for å oppfylle akseptansekriteriene er:*

* Client

*Med dem så må man:*

* Koble til ekstern server (en annen spiller).

*Sentrale metoder er:*

*KRAV: Vegger må være solide*

### Brukerhistorie:

Som en vegg ønsker jeg å blokkere andre spillere og lasere

### Akseptansekriterier:

* Når en robot eller laser prøver å bevege seg mot en vegg, skal de bli nektet avgang.

### Arbeidsoppgaver:

Vegg-klasse som holder styr på hvilken rute og i hvilken direction det befinner seg en vegg

En metode man kan kalle for å spørre om denne tilen har en retning den veien du vil, eller tilen i den retningen du vil dra til har en vegg mot din retning.

*KRAV: Laser må kunne bevege seg*

### Brukerhistorie:

Som en laser vil jeg kunne bevege meg i en rett linje til jeg treffer en vegg

### Akseptansekriterier:

* Når laseren skytes skal den gå gjennom alle tiles i retningen
* Skal ikke penetrere vegger
* Skal penetrere roboter

### Arbeidsoppgaver: