

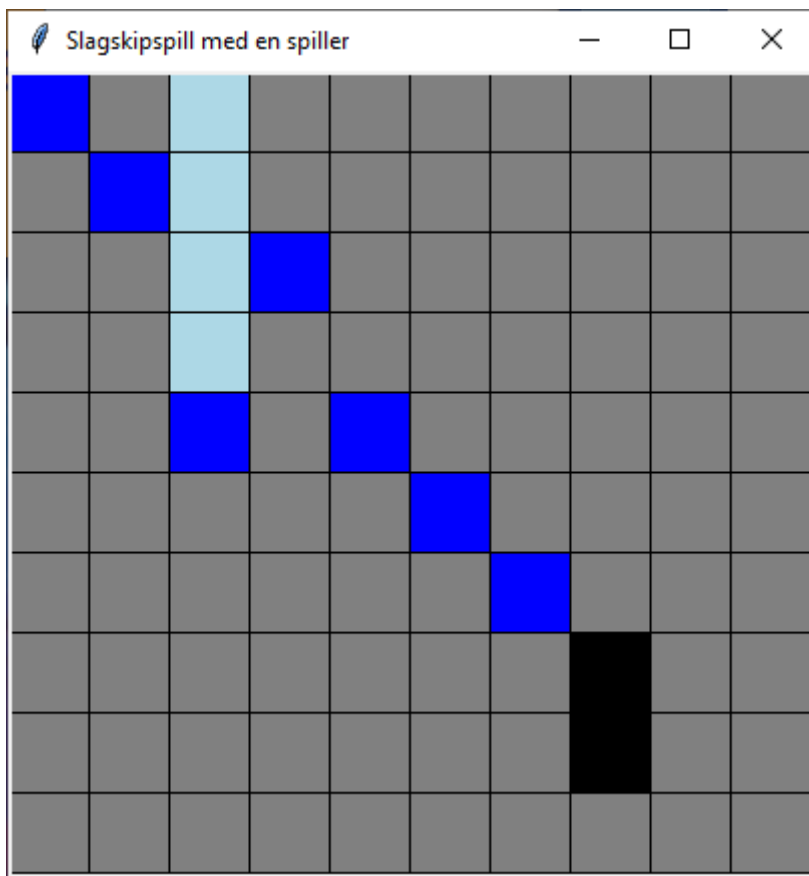
# DAT110 øving 10: GUI Slagskip

## Læringsmål

Du skal lære hvordan å lage et mer komplisert grafisk brukergrensesnitt.

## Oppgavebeskrivelse

Du skal lage et grafisk grensesnitt for å spille slagskipspillet. Du skal bruke spillet fra øving 8 (enten ditt eget eller løsningsforslaget) og lage et grafisk brukergrensesnitt for det, som kan se ut som figuren under:



I dette eksemplet på et grensesnitt brukes følgende fargekoder:

- Grå: Ruter du ikke har skutt på enda
- Mørkeblå: Ruter som ikke inneholdt noe skip
- Svart: Ruter som inneholder et skip som fortsatt flyter
- Lyseblå: Ruter som inneholder et skip som er senket

Brukeren skyter på en rute ved å klikke på den. GUI-et skal vise fram en dialog med en feilmelding hvis brukeren prøver å skyte på ei rute som allerede er skutt på. Når brukeren vinner skal det vises som en dialog, og spillet avsluttes. Du kan avslutte et GUI ved å lukke alle vinduene, i dette tilfellet ved å kalle metoden `destroy()` på hoved-vinduet.

Merk: GUI-et ditt trenger ikke å se ut som på figuren, det som figuren viser er en enkel måte å gjøre det på, og bruker et Canvas med mange ruter. Å bruke knapper i stedet er også en løsning. Ønsker du

en utfordring kan du tegne mer avanserte figurer eller prøve deg på å importere bilder eller annen grafikk og bruke det.

## Deloppgaver

I denne oppgaven trenger du ikke å følge deloppgavene så lenge du løser problemet definert i forrige kapittel. Deloppgavene er inkludert for de som trenger litt veiledning på hvordan å løse oppgaven.

- a) Lag en metode «`innhold_i_rute`» i Slagskipspill klassen fra øving 8. Denne skal ta koordinaten til ei av rutene i spillet som parameter og returnere en verdi basert på innholdet. Definer konstanter som representerer de fire tilstandene ei rute kan ha og returner konstanten for ruta sin tilstand. De fire tilstandene er: ikke skutt på enda, åpent vann, skip som flyter, skip som er senket.
- b) Lag en klasse for en enkelt rute i spillet. Ei rute skal ha koordinaten for ruta (dette er rute 5 i x-retning og 3 i y-retning), en referanse til et Canvas objekt, en referanse til Canvas figuren som representerer ruta, og en referanse til spillet. Alle disse skal overføres til ruta i konstruktøren. Lag deretter en metode oppdater som spør spillet om status for denne ruta og oppdaterer Canvas-figuren til ruta basert på dette.
- c) Lag en klasse for GUI-et. Konstruktøren skal sette opp selve grensesnittet, lage et 10x10 brett med rutene fra oppgave b hvor et rektangel blir opprettet i Canvas-objektet for hver rute.
- d) Lag en metode som håndterer museklikk. Denne skal regne ut hvilken rute brukeren klikket på basert på koordinaten til muse-hendelsen. Så skal den kalle skyt-metoden til spillet fra øving 8 på denne ruta og deretter oppdaterer GUI-et. Til slutt skal den sjekke om brukeren har vunnet, og gi beskjed.
- e) Bind <Button> hendelser på Canvas-objektet til å kalle metode fra forrige deloppgave
- f) Lag en if `__name__ == "__main__"` blokk som starter GUI-et.
- g) **Frivillig:** Utvid GUI-et til å støtte den frivillige oppgave i) i øving 8, hvor du har to spillbrett hvor du spiller mot datamaskina. Dette involverer å lage et GUI for å plassere skip samt GUI-et for spilleren sitt brett hvor spilleren skal se hvor skipene ligger og hvor datamaskina skyter.