

# COMPUTERSPIL 4

I denne fjerde delaflevering skal I bruge nogle af de ting, som I har lært om grafiske brugergrænseflader til at tilføje nogle ekstra knapper, labels og tekstfelter til **GUI** klassen. Herudover skal I tilføje en menubar. Endelig skal I rette de fejl og mangler, som instruktoren har påpeget i jeres tredje delaflevering.

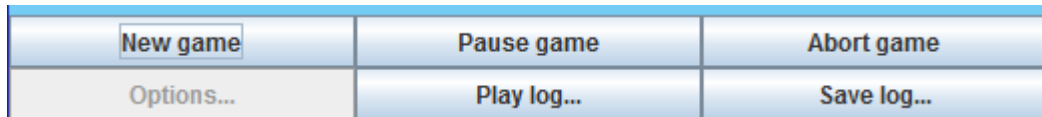
Husk at holde jeres dokumentation opdateret, så den afspejler de ændringer/tilføjelser, som I laver i jeres kode. I behøver ikke at lave regression tests, der afprøver jeres GUI objekter, men de skal selvfølgelig testes manuelt, efterhånden som I laver dem.

Testserveren bruges ikke i denne opgave.

## Opgave 1

Studér hvordan **createButtonPanel** metoden i **GUI** klassen skaber de knapper, som vises under kortet.

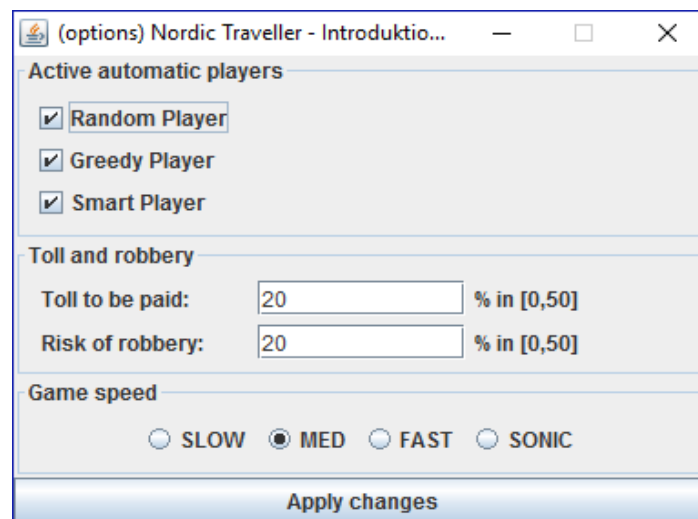
Skab to nye knapper, således at nederste del af vinduet får nedenstående udseende (hvilket bl.a. kræver, at I ændrer parametrene for det **GridLayout**, der bruges):



I kan med fordel kopiere store dele af koden for de eksisterende knapper. De to nye knapper skal bruges i Computerspil 5. Pt. skal I blot lade lamda'erne i **addActionListener** kaldene mappe **ActionEvent e** til hhv. **testPlayButton()** og **testSaveButton()**. Begge metoder ligger i **GUI** klassen (ikke i **Game** klassen).

## Opgave 2

Når I trykker på **Options** knappen (hvilket kun kan gøres, når spillet er standset, f.eks. efter et tryk på **Abort** knappen) åbnes nedenstående dialogboks, som I nu skal udvide lidt:

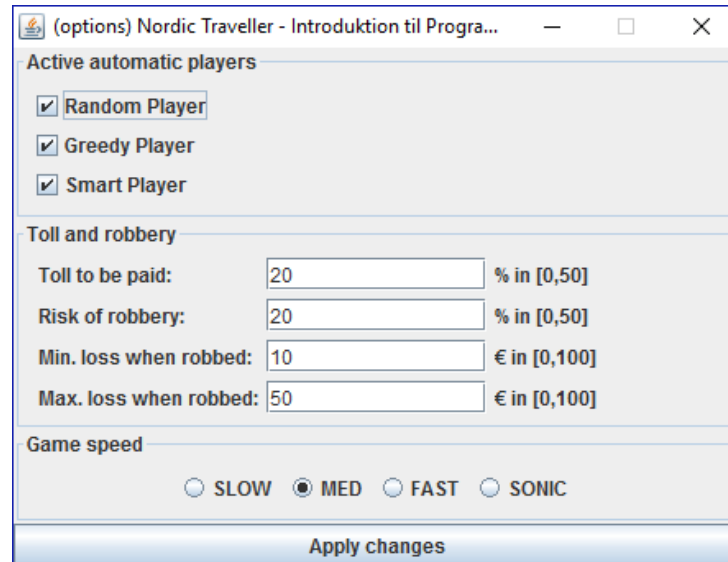


Start med at tilføje to nye feltvariabler ***minLossTextField*** og ***maxLossTextField*** af type ***JTextField*** til GUI klassen (under de eksisterende feltvariabler ***tollSizeTextField***, ***robberyTextField***).

Studér hvordan ***createOptionsDialogBox*** metoden i ***GUI*** klassen, for hvert af de to eksisterende valg i det midterste panel, skaber:

- En ***JLabel*** med beskrivende tekst (Toll to be paid og Risk of robbery).
- Et ***JTextField***, hvori den aktuelle værdi vises og kan ændres (20). Værdien i tekstfeltet gemmes i en feltvariabel i ***JTextField*** objektet.
- En ***JLabel***, der angiver enheden og det tilladte heltalsinterval (% in [0,50]).

Tilføj dernæst to nye valg til det midterste panel, således at dialogboksen får nedenstående udseende:



I kan med fordel kopiere store dele af koden for de eksisterende valg. Som det ses, kan de to nye valg begge antage heltalsværdierne i intervallet [0,100]. Defaultværdierne skal være henholdsvis 10 og 50. Det ***JPanel*** objekt, hvor I skal tilføje de nye valg, benytter et ***GridLayout***. Husk at modificere konstruktøren til det pågældende ***GridLayout*** objekt, så der er plads til de nye komponenter (se evt. Java API).

## Opgave 3

I skal nu tilføje funktionalitet til de to valg, som I lige har tilføjet. Dette sker i metoden ***applyOptions***.

Aflæs værdierne i jeres to ***JTextField*** objekter, tjek at de begge er heltal i intervallet [0,100], og at minimumsværdien er mindre end eller lig maksimumsværdien. Såfremt der er fejl i tekstfelterne, skal alle fejl vises samlet i én dialogboks. Det vil sige, at I også skal ændre i koden for de andre to tekstfelter. I skal akkumulere alle fejl ved hjælp af ***StringBuilder*** klassen, og vise den samlede tekst ved hjælp af ***showMessageDialog*** metoden. Husk at lave gode fejlbeskeder, så brugeren let kan forstå, hvad der gik galt.

Hvis værdierne er ok, opdateres de pågældende feltvariabler i ***Game*** klassen via metodekaldet:

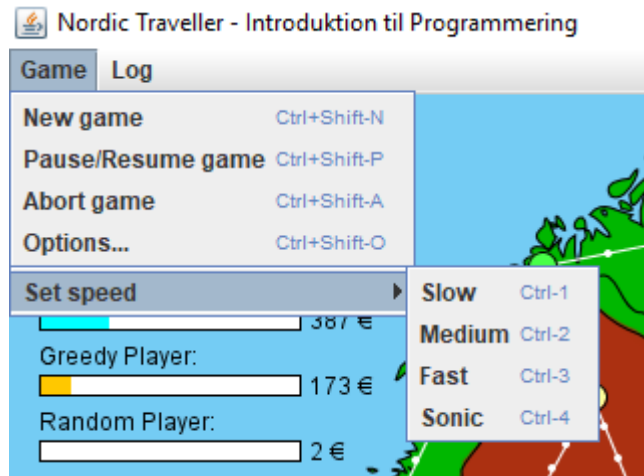
- ***game.getSettings().setMinMaxRobbery(min, max)***

hvor ***game*** er en eksisterende feltvariabel i ***GUI*** klassen.

Når brugeren trykker på ***Options*** knappen, kan hun, i den dialogboks, der kommer op, se de eksisterende valg. Studér hvordan metoden ***applyExistingSettings*** i ***GUI***-klassen virker, og modificér metoden, så den også håndterer de to tekstfelter, som I lige har tilføjet.

## Opgave 4

I skal nu skabe en menubar, hvor *Game* menuen har de viste indgange, mens *Log* menuen har to indgange *Play log...* og *Save log...* De enkelte menuindgange skal have samme effekt, som de tilsvarende knapper nederst i vinduet. Der vil altså nu være to måder, hvormed brugeren kan vælge disse operationer. Ved forelæsningerne er det beskrevet, hvordan man får menuindgangen *New game* til at have samme effekt som *New game* knappen.



Menubaren tilføjes til **mainFrame** i **GUI** klassen ved hjælp af **setJMenuBar** metoden.

Når I skal implementere event håndtering, kan I med fordel læse dokumentationen for **GUI** klassen. Hastigheden af spillet ændres ved hjælp af den private metode **setSpeed(int speed)**, hvor 1=slow, 2=medium, osv.

Dernæst skal I tilføje de viste tastaturgenveje til menuindgangene. Ved forelæsningerne er det beskrevet, hvordan dette gøres for menuindgangen *New game*. *Play log...* skal have genvejen *Ctrl+Shift-L*, mens *SaveLog...* skal have genvejen *Ctrl+Shift-S*.

I behøver ikke at få *Pause/Resume game* menuindgangen til at alternere mellem *Pause game* og *Resume game*, sådan som den tilsvarende knap gør. Men har I lyst til at prøve kræfter med det, er I selvfølgelig velkomne.