IN1010

Repetisjon Uke 13
GUI 1

(Basert på Dag Langmyhrs presentasjon)

25. april, 2025

Graphical User Interface

Det er enkelt med terminalen, men det kan fort bli kjedelig og ikke så alt for brukervennlig.

Så da kan vi heller bruke GUI. Da er det lettere å lage noe som er mer intuitivt og ser bedre visuelt.

Når det blir enklere for brukeren, så får vi også færre brukerfeil.

Graphical User Interface

Fordeler

- Lettere å lage brukervennlig
- Færre brukerfeil
- Bedre muligheter for visualiseringer, som illustrasjoner
- De fleste synes det ser bedre ut

Graphical User Interface

Ulemper

- Mer krevende og komplisert å programmere
- Har en viss grad av platform-avhengighet

  (Windows, MacOS, Linux, osv.)

- Mange bibliotek å velge i, som kan være platform-spesifikke

  (Qt, GTK, wx, Tkinter, Flutter, Windows (Win32/Forms/WPF/UWP/WinUI), Mac (Cocoa), osv.)

GUI i Java

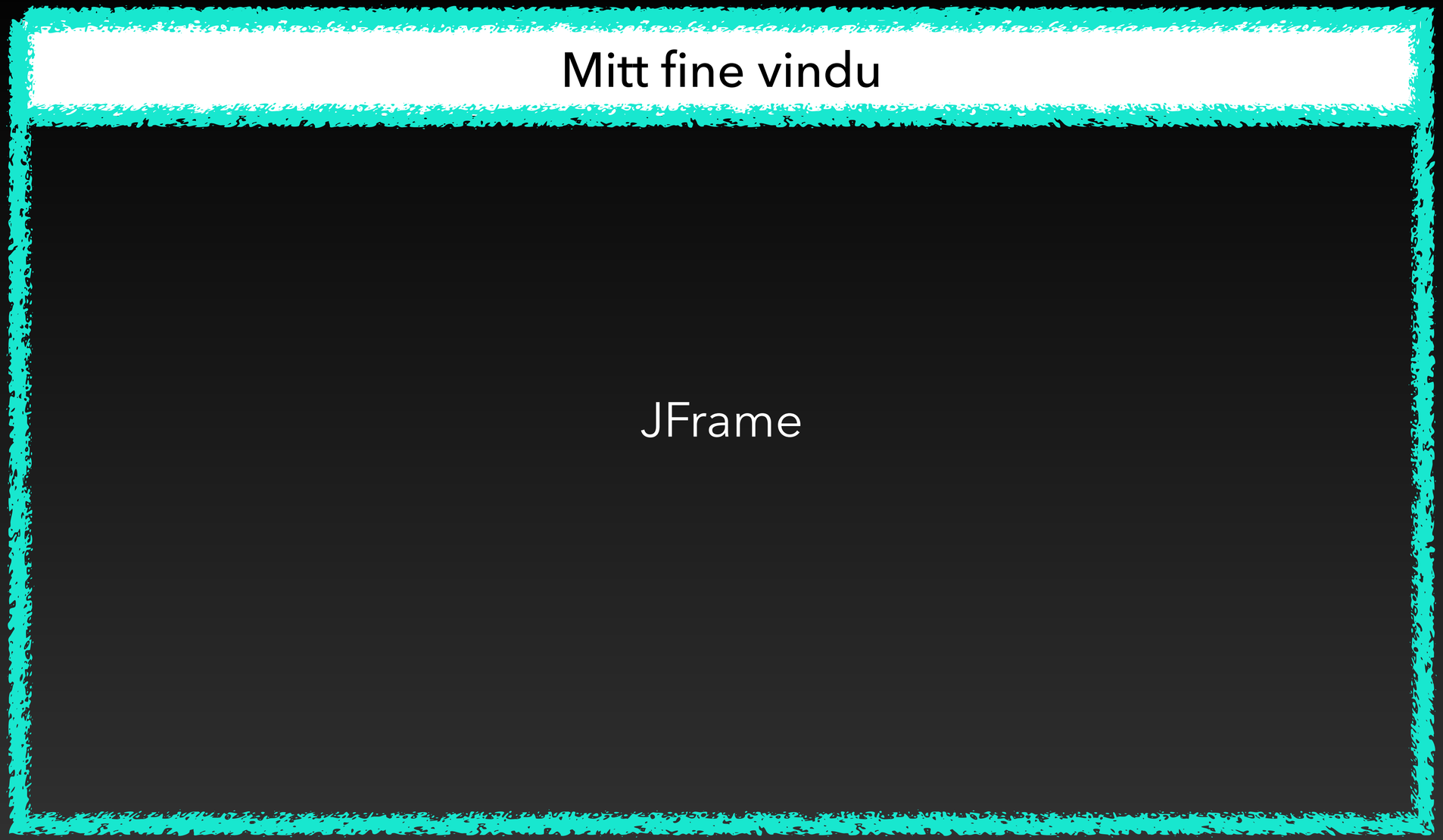
Java har også flere bibliotek, men vi vil i hovedsak bruke to:

- AWT – Abstract Window Toolkit
- Swing – En utvidelse av AWT, med flere avanserte komponenter

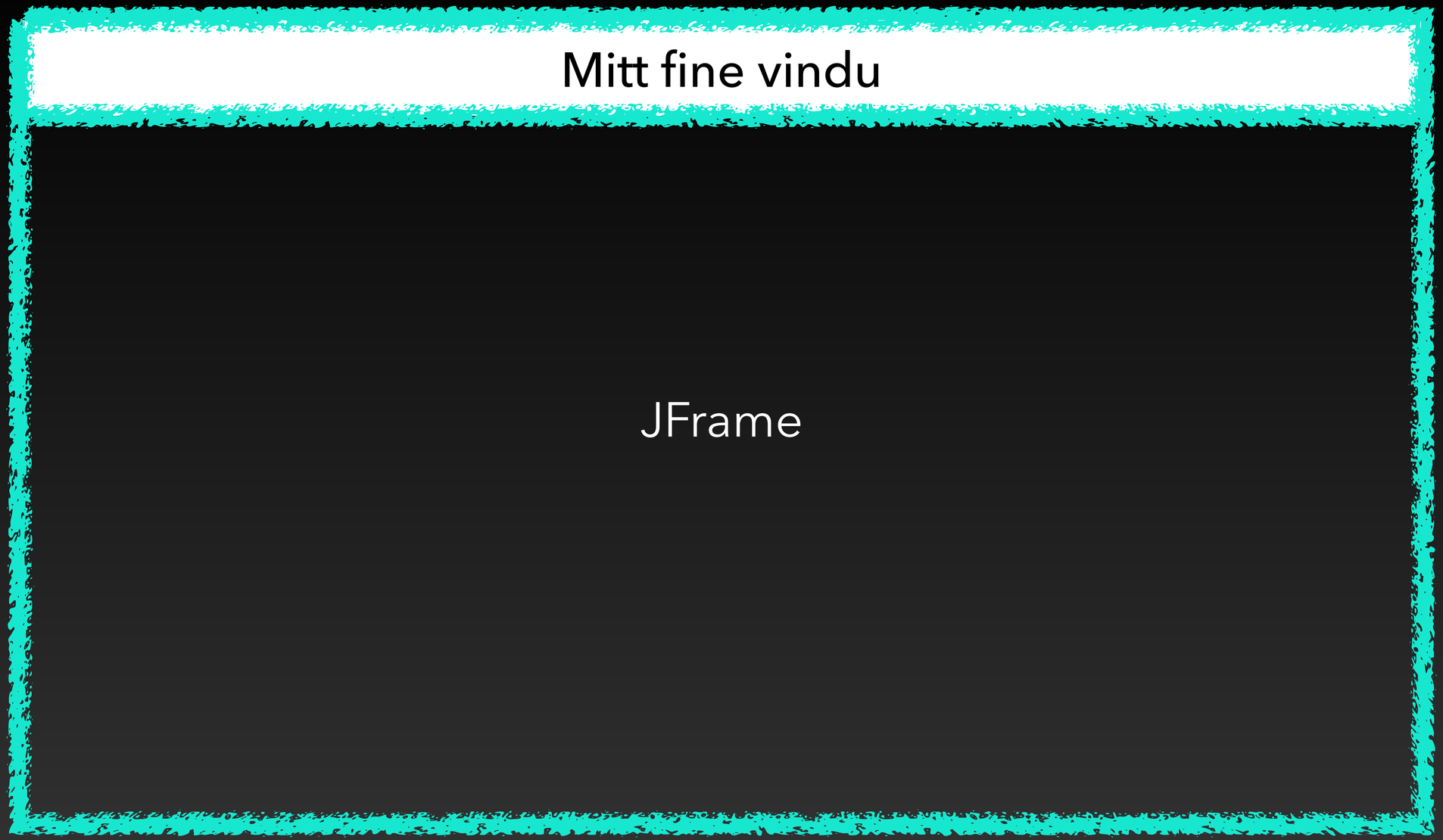
Disse er en del av en vanlig Java installasjon

Vi lager GUI ved hjelp av byggeblokker kalt komponenter

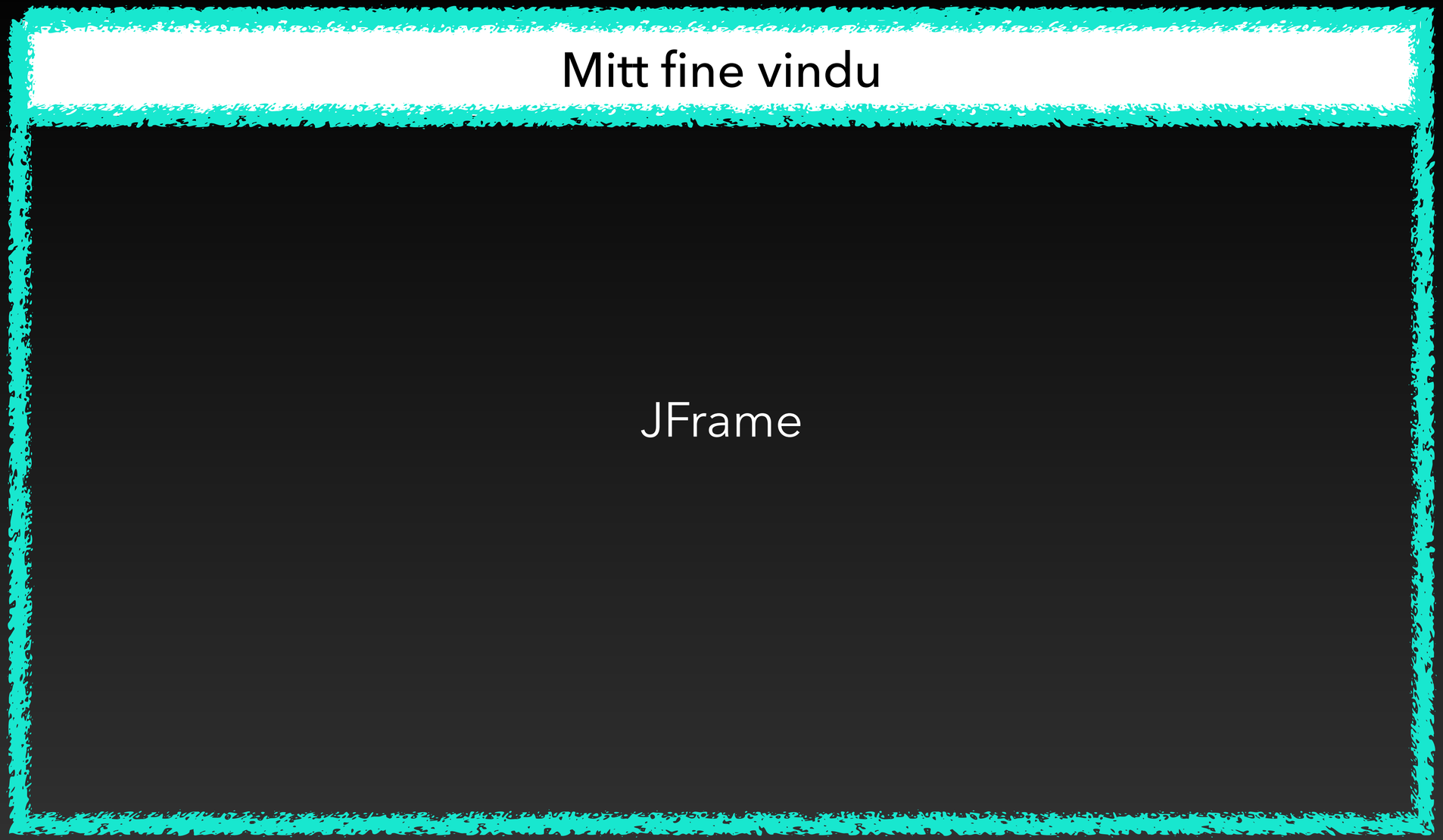
Komponentene er et godt eksempel på objekt-orientert system



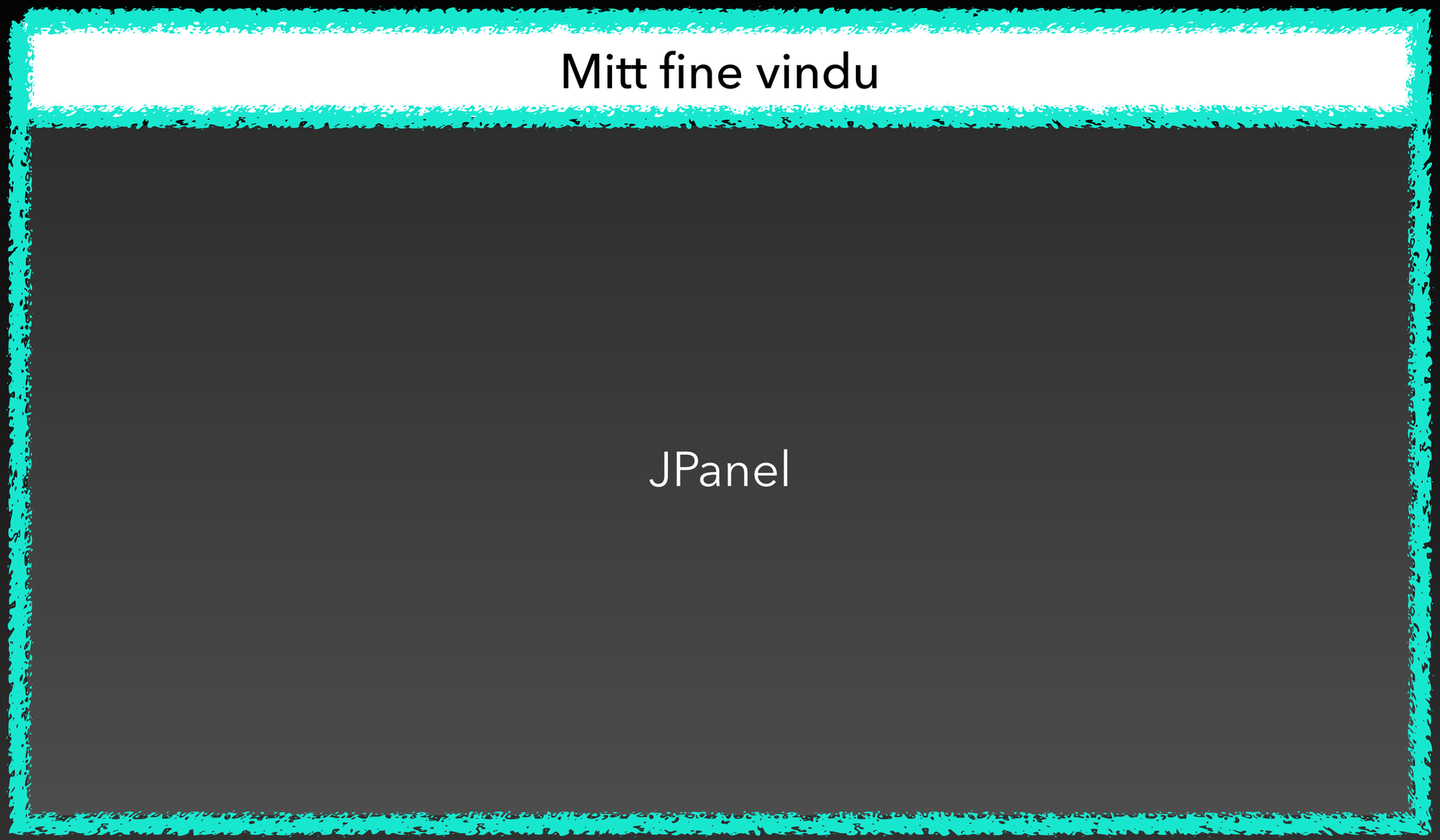
Vi begynner med å opprette vinduet, som i Java kalles ramme (JFrame)



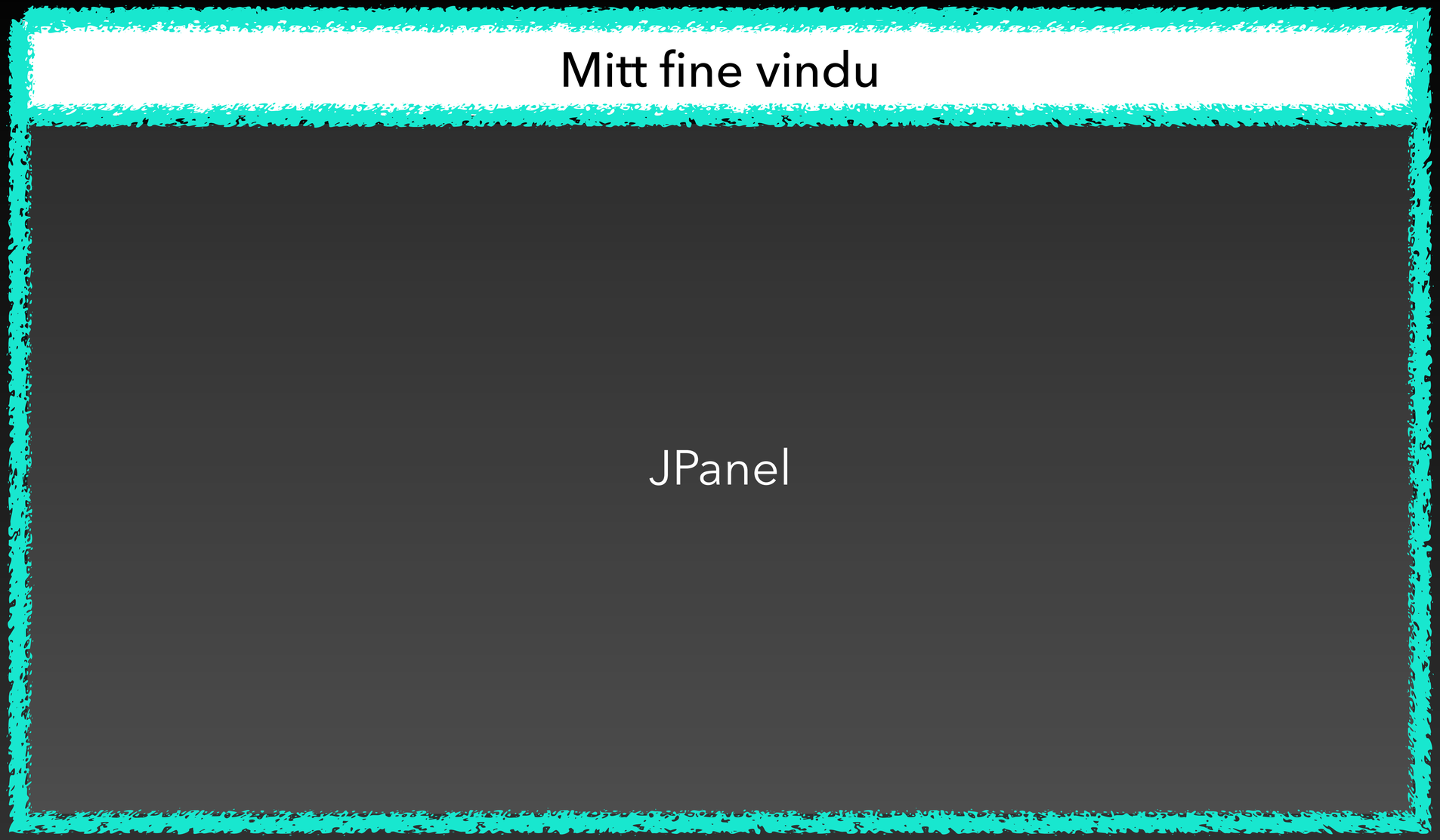
Dette lager et vindus-objekt, med fungerende avslutt-knapp



Selv om vi kan legge til de fleste komponenter rett i et vindu, så ønsker vi mer kontroll

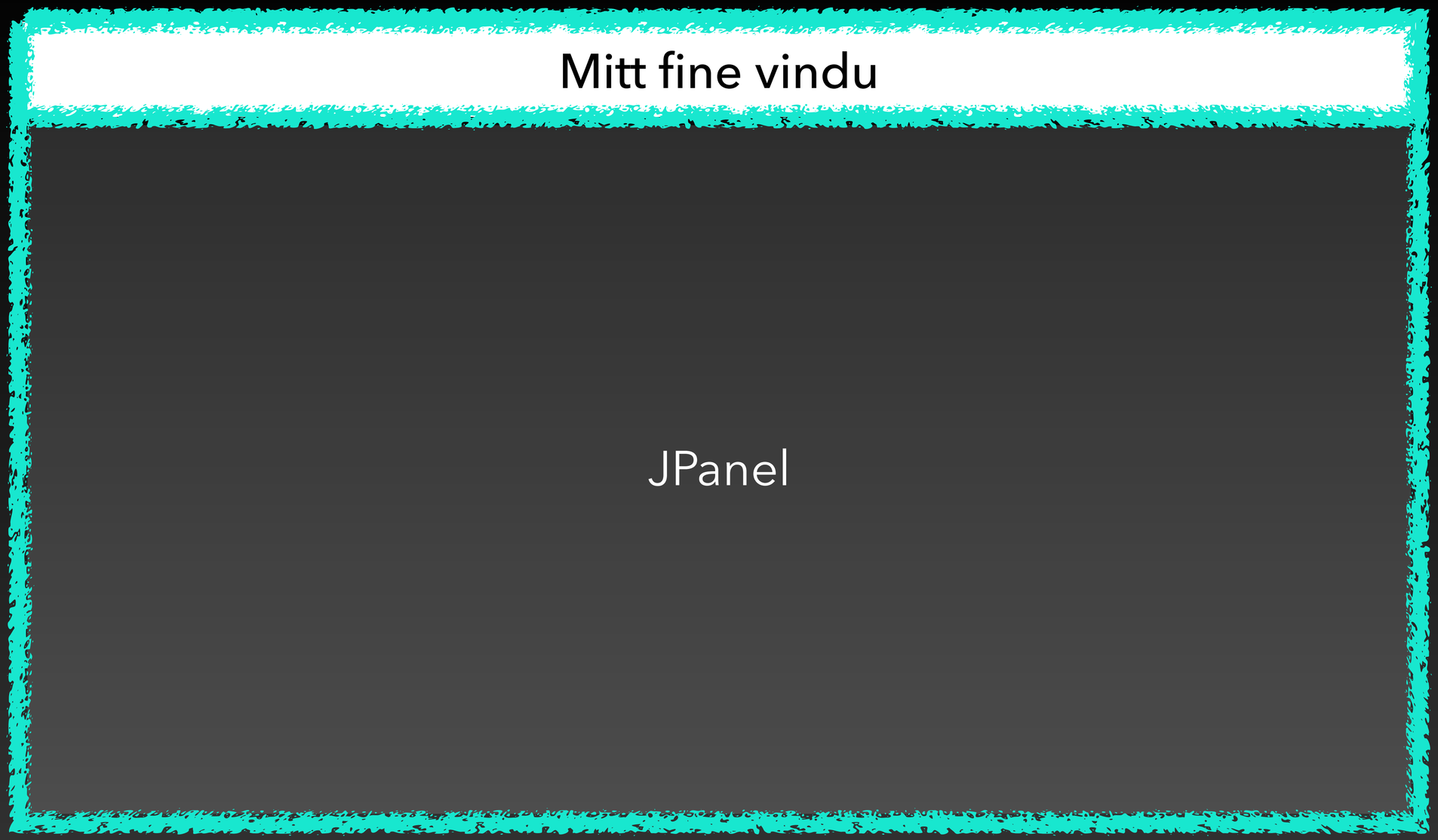


Måten vi gjør det på er å opprette ett panel, eller tegneflate (JPanel)

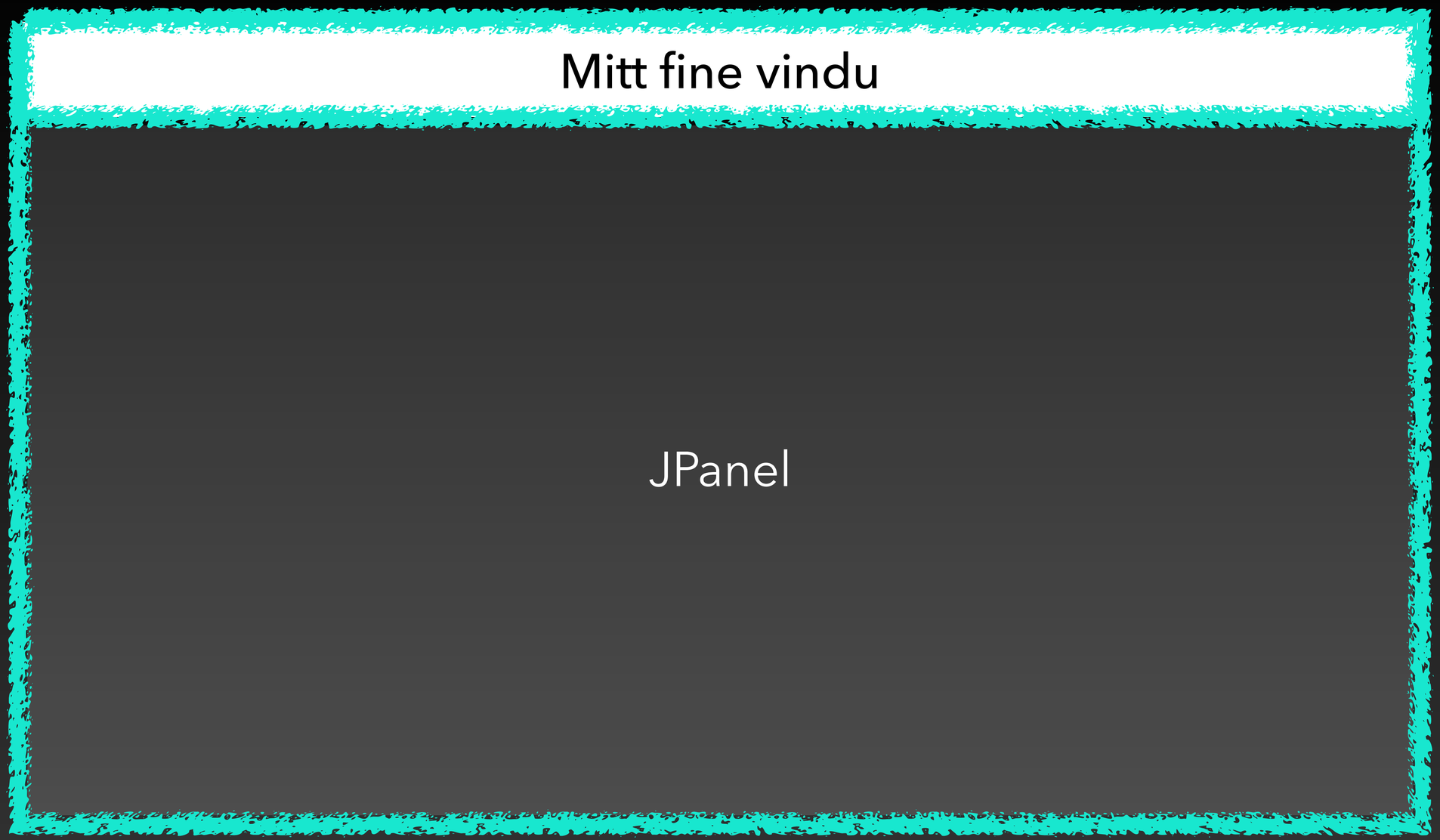


Det første panelet fyller hele innholdet i vinduet, men ikke tittel-linje og rammer

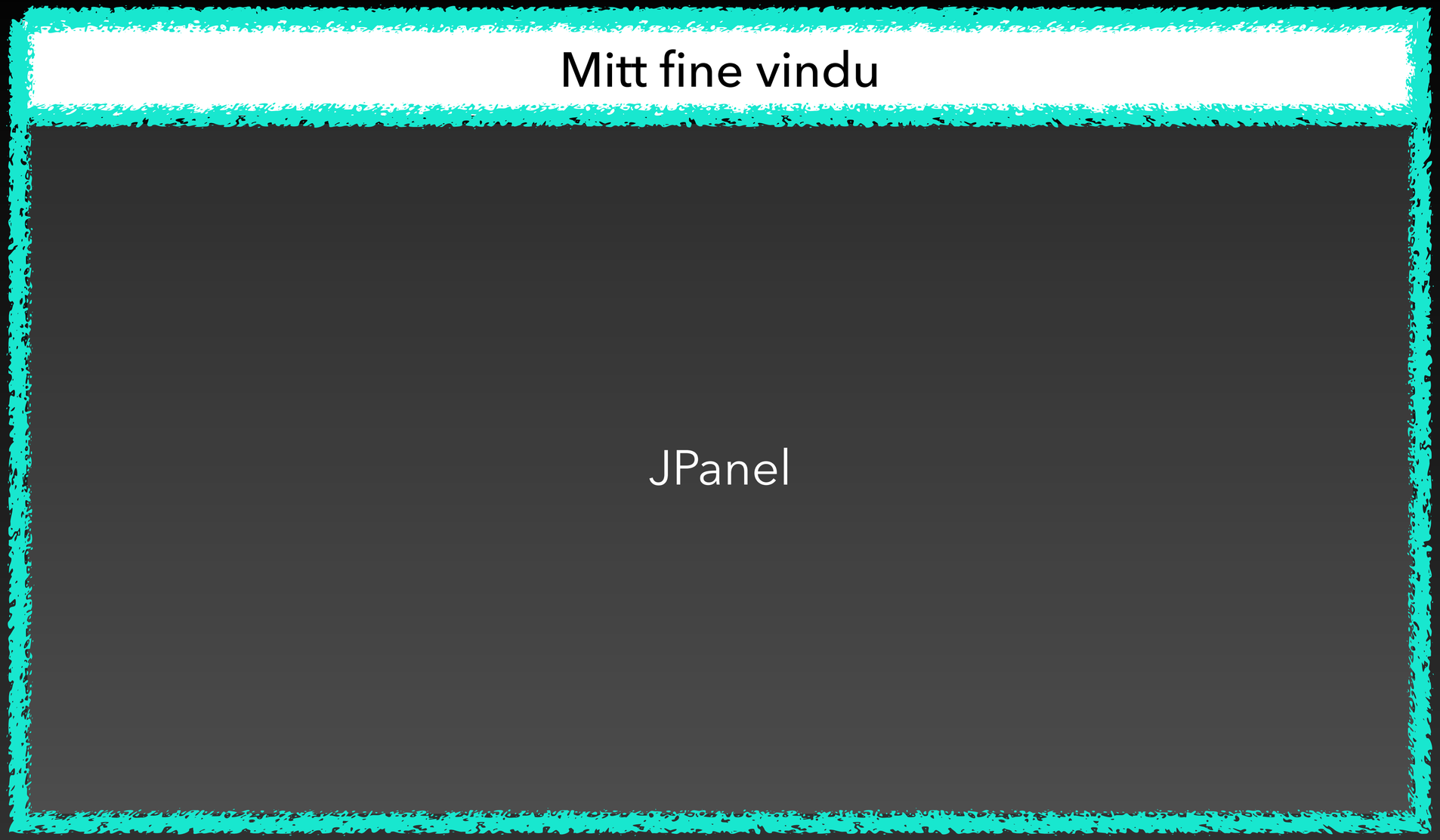
Vi kan ha flere paneler som tar opp hver sin del



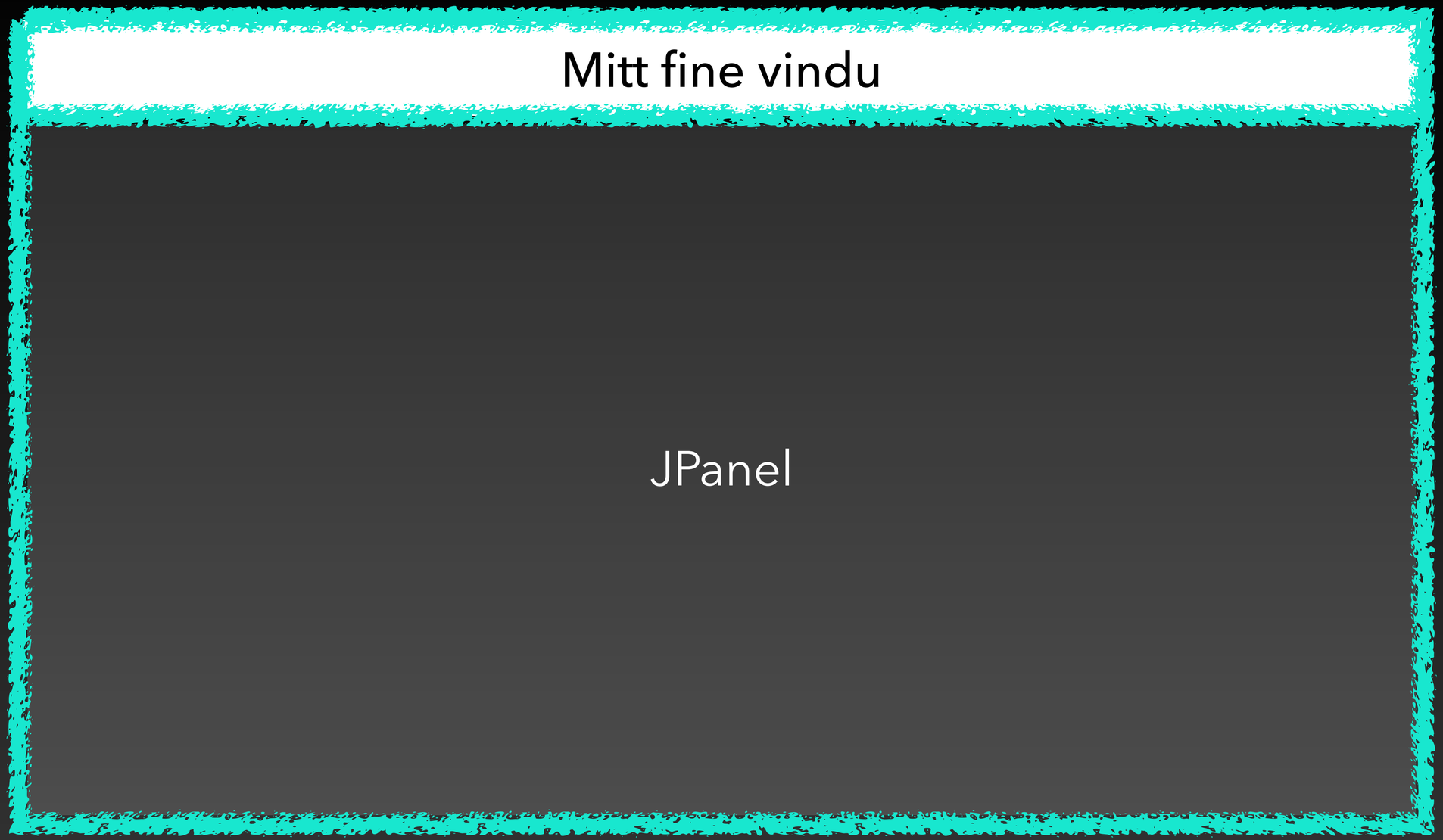
Men foreløpig er vinduet og innholdet kun objekter i Java, og ikke noe som er synlig for oss



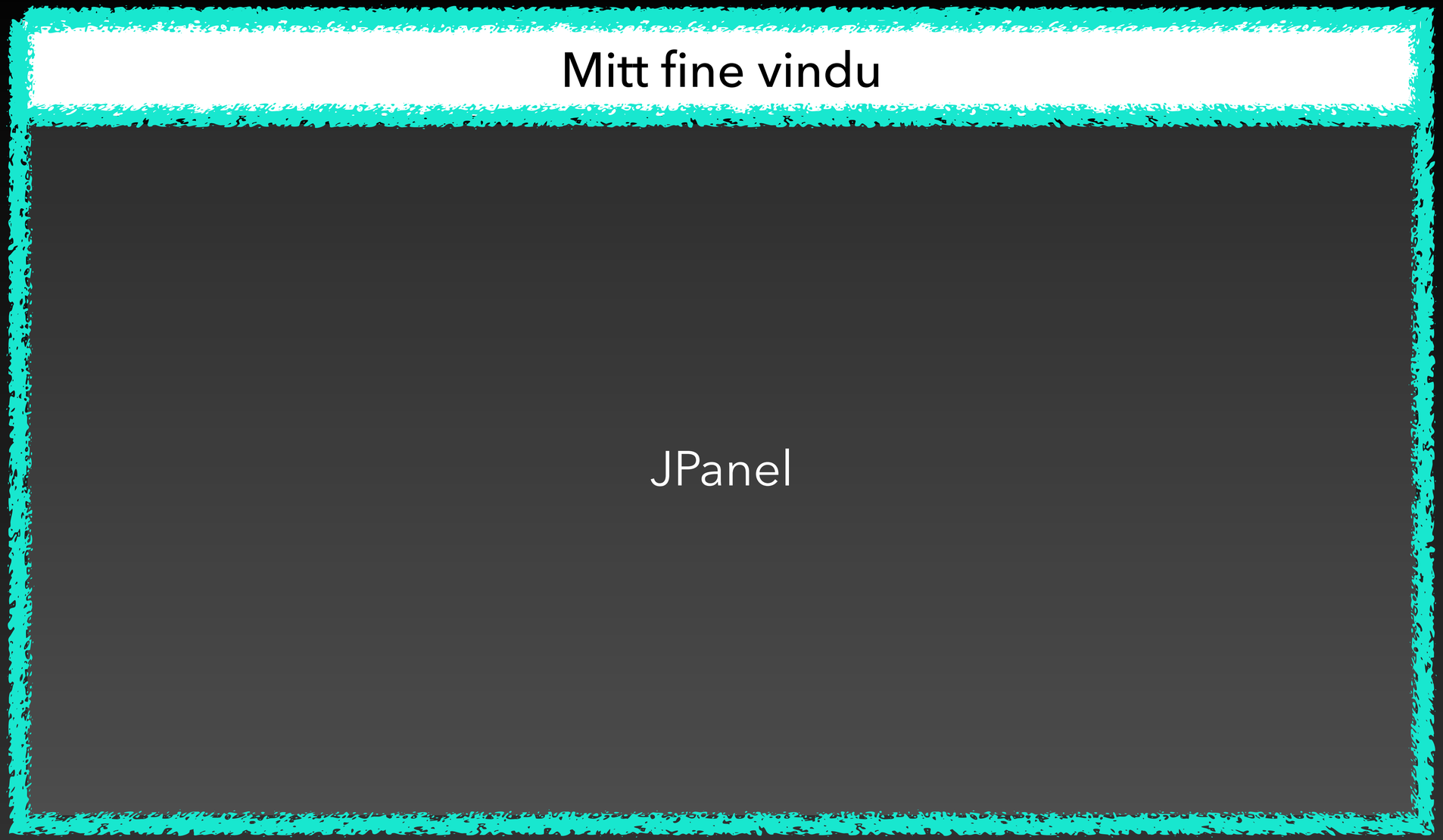
Så det får vi gjøre noe med



Først så "pakker" vi vinduet. Det betyr at Java prøver å plassere alt slik at det passer sammen slik vi ønsker, og så vil den så godt den klarer tilpasse vindusstørrelsen

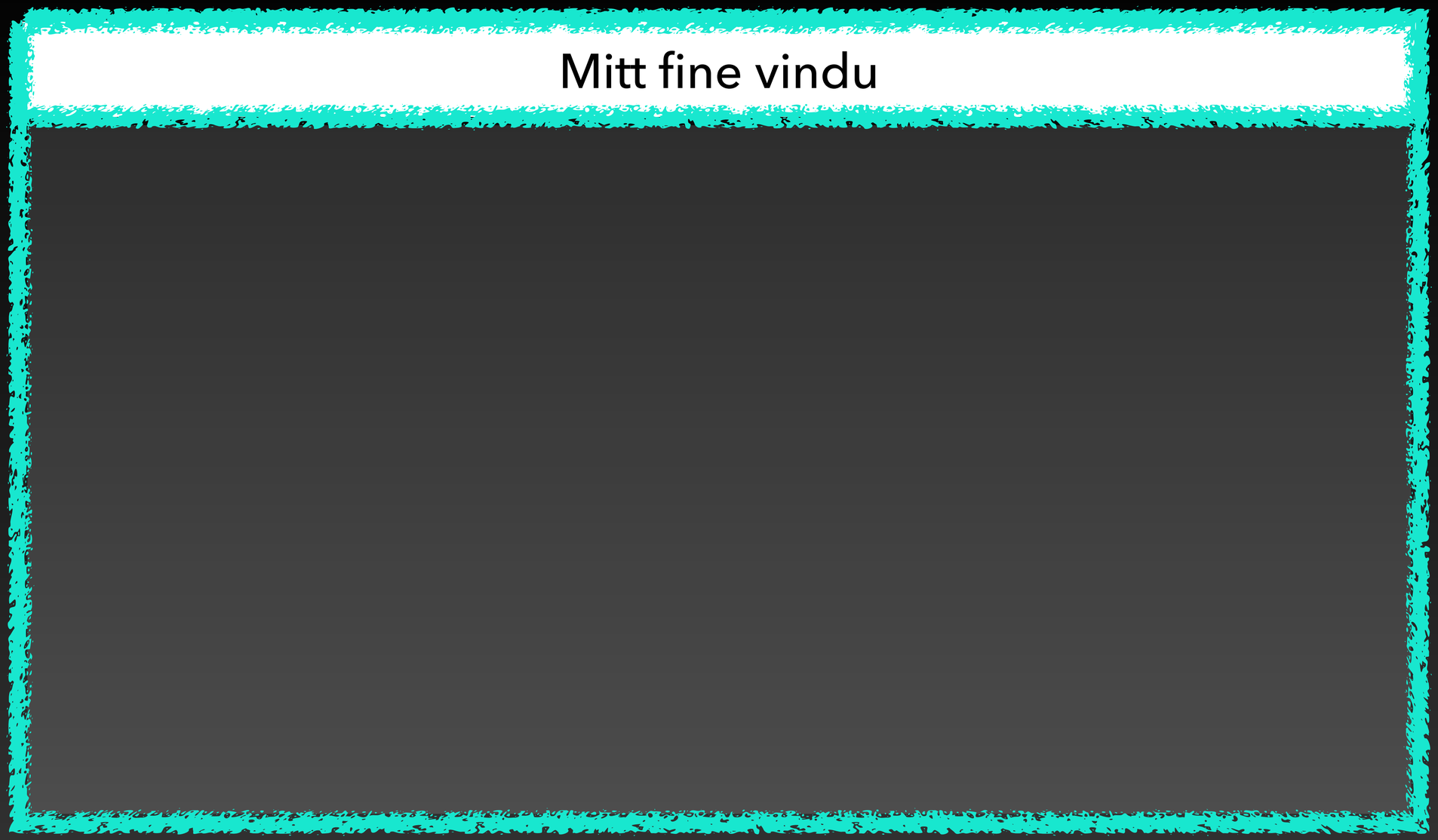


Når vi vet størrelsen på vinduet, så kan vi plassere vinduet midt på skjermen

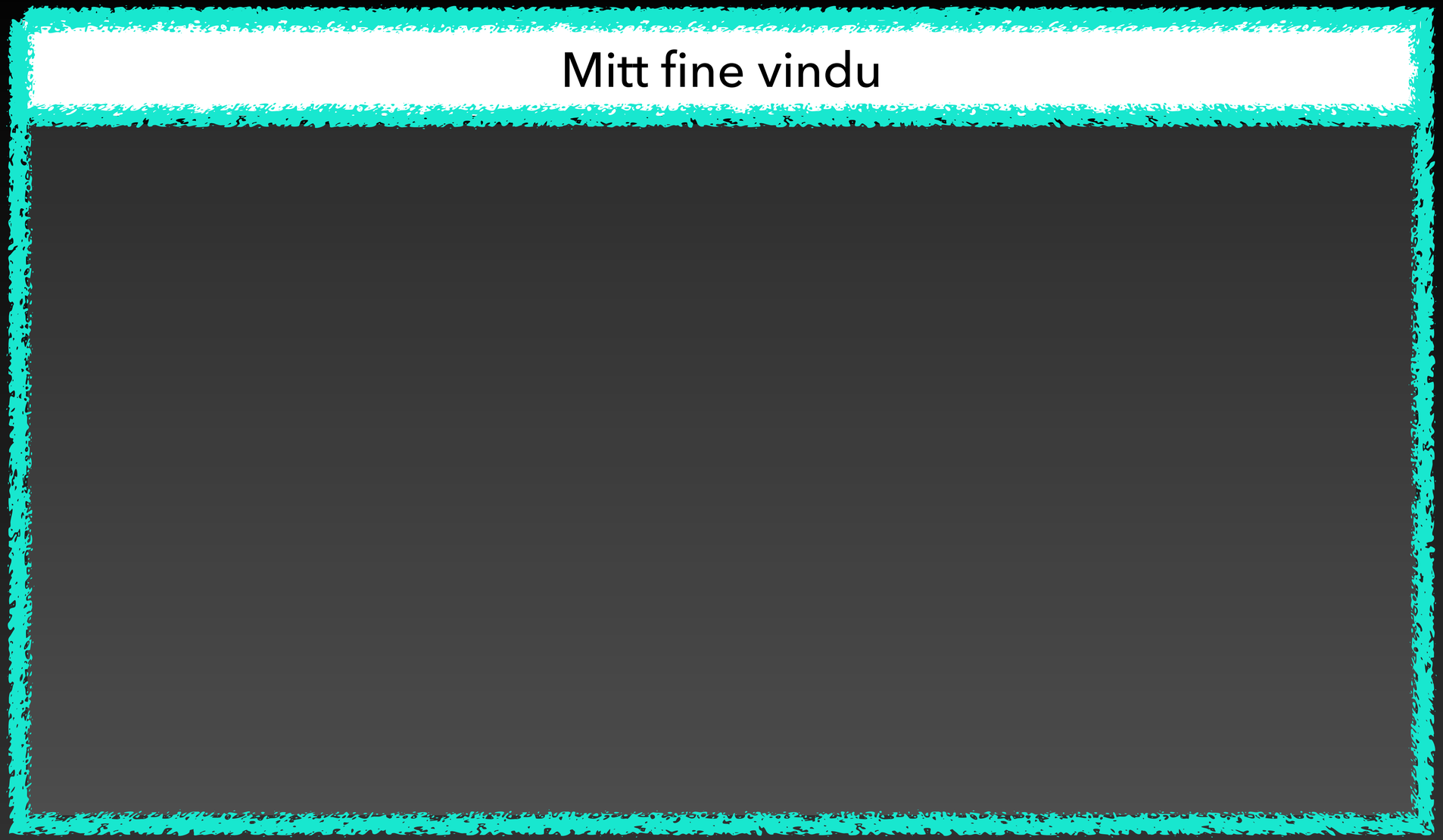


Til slutt så gjør vi vinduet synlig.

Hvis du kjører det vi har nå, så ser det nok ganske lite ut.



Men dette ser jo litt kjedelig ut, heldigvis har vi mange ulike typer komponenter vi kan fylle på for å gjøre vinduet mer nyttig



Vi kan f.eks. sette inn tekst. En måte å gjøre det på er ved å bruke JLabel



Da kunne det blitt noe slikt.

I praksis, så vil det etter at vi kjører .*pack()*-metoden, være et vindu like stor som teksten



Kanskje vil vi ha noen knapper som vi kan trykke på? Til det så har vi JButton

Mitt fine vindu

Hallo! Har du det bra?

Ja 😄

Nei 😢

Nå begynner det bli mer interessant, men hvis vi trykker på de, så skjer det jo ingenting



For å fikse det så kan vi lage en klasse som implementere en lytter (listener)

Mitt fine vindu

Hallo! Har du det bra?

Ja 😄

Nei 😢

Så kobler vi lytterne til knappene vi ønsker at de skal lytte til

Event Driven Programming

Eller hendelses-drevet programmering på norsk.

Det betyr at programmene vi lager svarer på hendelser, eller sender ut hendelser.

I GUI så er det den mest vanligste måten bibliotekene er bygget opp, og hver interaksjon fra brukeren vil generere hendelser.

Event Driven Programming



Måten det er bygget opp på er at når vi oppretter GUI-komponenter, så lager Java en egen tråd, *event dispatch thread*.

Denne går i en kontinuerlig løkke, og består av en kø som alle hendelsene blir lagt til.

Hver hendelse blir så, hvis det eksisterer lyttere, sendt til det programmet som de tilhører.

Event Driven Programming



Det viktige her er å være klar over at vi nå har to tråder, *main* og *event*.

Alle hendelser vil kjøres av event-tråden.

Det betyr også at hvis lytteren vi lager bruker lang tid, så blir ikke de andre hendelsene prosessert før den er ferdig.

For en bruker vil det føles som programmet har stoppet.

Event Driven Programming



Da må vi også passe på at vi er trådsikre.

All GUI bør kjøres av event-tråden.

Men skal vi kjøre noe som tar tid og er tungt, så må vi passe på at vi oppretter en egen tråd.