

# skadedyrkontrollørene

skadedyrkontrollørene

Controller

- buttons
  - menu

- **IMenuButtons**

- High coupling, rigid interface.
- Enhver klasse som implementerer dette interfacet må ha alle de spesifiserte knappene. Hva hvis man tenker i fremtiden at man vil ha færre knapper, f.eks ikke en speed-knapp. Jeg hadde foretrukket et mer generelt interface, og så kan heller klasser være mer konkret.
- Jeg tenker også at hvis man finner et behov for en annen meny i kodebasen senere, så kan man ikke gjenbruke dette interfacet, fordi det er så spesifikt. Så et mer generelt interface hadde tillatt gjenbruk av kode.

- **MenuButtons**

- fjern ubrukte imports
- Liker bruk av factory
- støtter ikke scaling
  - magiske tall
- updateButtonAppearance
  - her burde det være forskjellige navn på de to metodenene. F.eks så kunne den som er public hatt navnet updateAppearanceOfPauseAndSpeedButtons eller at det hadde vært lagt til dokumentasjon som forteller hvilke knapper som blir oppdatert.

- upgrade

- **IUpgradeButtons**

- Hvis IMenuButtons var et mer generelt knappeinterface, så kunne man bare gjenbrukt det, istedenfor å lage enda et interface. F.eks så har begge interfaces updateButtonAppearance metoden.

- **UpgradeButton**

- updateButtonAppearance kunne hatt en kommentar om hvilke knapper som oppdateres.

- **ButtonFactory**

- God ide med en buttonfactory!

- **SkadedyrController**

- handleUserInput

- magiske tall

main

- **SkadedyrGame**
  - savner javadoc
  - Clean og enkel klasse!
- **SkadedyrMain**

model

- catmenu
  - **ICatMenu**
    - Siden dette også er en meny, så kunne man brukt Button her også
  - **CatMenu**
    - Array<T> fra Libgdx istedenfor ArrayList.
    - Bra at konstanter er definert, istedenfor magiske tall.
- entities
  - cat
    - **ICat**
      - Interfacet bryter med Interface segregation principle fra SOLID
        - Det inkorporerer alt for mye og burde heller vært delt opp i flere mindre interfaces.
        - Ved å omfavne så mye, så blir interfacet mye mer konkret og begrenser utviklers frihet til hvordan man skal implementere. Det setter også begrensninger for code reusability ved å være så spesifikt.
    - **Cat**
      - Siden ICat spesifiserer så mye, så blir det på en måte dobbelt opp med informasjon i Cat. Det hadde vært bedre at ICat deklarte det essensielle med en katt, mens Cat gir en generell implementasjon som så blir mer spesifikk i subclassene. Dermed kunne man fjernet informasjonen fra ICat som nå er dobbelt opp i Cat.
    - **BasicCat**
      - fin og grei
    - **FreezeCat**
      - fin og grei
    - **ShotgunCat**
      - fin og grei
  - rat
    - **IRat**
      - Feedbacken her blir egentlig samme som ICat

- **Rat**
  - kanskje man kunne gruppert noen av feltvariablene i egne objekter, for å korte ned på listen. F.eks det som har med det grafiske som sprite og textures kunne vært i et eget objekt for å skille det litt fra resten.
  - I createPath() så er det masse magiske tall. Skjønner at det er koordinater, men de kunne kanskje vært deklarerert i et path-objekt eller lignende.
  - Jeg tenker at enums burde vært deklarerert i sin egen klasse, istedenfor å være bortgjemt i Rat-klassen.
  - Bra at dette er en superklasse, slik at underklassene blir mye kortere og mer ryddig.
- **BasicRat**
  - fin og grei
- **FastRat**
  - fin og grei
- **StrongRat**
  - fin og grei
- **RatFactory**
  - fin og grei

view

- states
  - **State**
    - Burde kanskje hatt et interface
    - Ellers fin og grei!
  - **GameOverState**
    - Fjern feltvariabel model. Ellers fin og grei.
  - **GameStateManager**
    - fin og grei
  - **HelpState**
    - fin og grei
  - **MenuState**
    - fin og grei
  - **PlayState**
    - Noen magiske tall, ellers fin og grei.
  - **WinState**
    - fin og grei
- **GameResourceFactory**
  - Savner dokumentasjon. Ellers fin og grei.

- **GdxTimeSource**
  - Savner dokumentasjon. Ellers fin og grei.
- **LibGDXResourceFactory**
  - Savner dokumentasjon. Ellers fin og grei.
- **SkadedyrView**
  - fin og grei
- **TimeSource**
  - Savner dokumentasjon. Ellers fin og grei.
- 

#### Generelt

- Synes det er bra at det er blitt brukt interfaces, abstrakte klasser og arv, men synes at man kan reflektere litt mer over hva et interface burde være og om det er i henhold til Solid.