

MÖTE 2018/04/19

- Fixa utvecklingsmiljön i git (se Carls bild).
- Vill ej bygga in libgdx i vår modell, ska vara helt "ren" → inget ska vara beroende av libgdx...
- Det är okej att view är libgdx-baserad, detta är även okej för controller.
- Toplevel-container som håller koll på aktuellt fönster.
- Hur byter vi scen? Via Toplevel scen? Som är osynlig och håller reda på vilken scen som visas. Ändra scen genom att ändra pekare hos TopLevel? Den kan känna av tangentnedtryckningar via en Listener..
 - ELLER ha olika stages - 3 stages i en övergripande stage?
 - Dessa kommer köras i en klass som extends ApplicationAdapter.
 - Varje stage ska ha en render metod?
 - Ha en optionsclass som alla ser (i view).
 - Allt det här tillhör view.
 - main-metoden ingår i view - dras ingång via libgdx (behöver ej tänka på).
- **view** → **controller** → **model**
- Ha en Factory-klass som bygger hela modellen?
- Vad ska vi göra med alla override metoder som vi inte använder?
 - Gör en Adapter?
 - Lägg alla override metoder i en klass...
- Gör klass för varje typ av komponent, till exempel en egen knapp klass där all kod för en knapp finns. Knapp klassen ska finnas i vyn.

UML sekvensdiagram

- shoot
 - skottet ska ha en färdig formel?
 - koda in med hjälp av gdx när vi testar
- move
- aim

TILLS NÄSTA MÖTE (26/4)

- körbart use case med grafik
 - tex en tank som kan skjuta och röra på sig.
- så ett körbart GUI