

# Mandatory Assignment in Advanced Software Construction

## Ide:

At designe og implementere et bibliotek (library) med klasser som kan udgøre et miniframework for et tur-baseret 2D spil. Du skal benytte forskellige teknikker og værktøjer du har lært i undervisningen. Dette framework skal dog **IKKE understøtte nogen form for GUI**.

I dette bibliotek skal du fokusere på at skabe en verden(world) med forskellige skabninger (creatures), der kan have forskellige forsvar (protection armor), som skjold, magi, støvler eller lign. samt forskellige angrebs våben (attack possibilities) som økse, sværd, boomerang, magi eller lign.

Du skal lave et 2D spil, fordi det er lettere at håndtere end 3D, ligesom det skal ikke være et realtidsspil (realtime game), igen for at gøre det lettere.

Baseret på de emner vi har gennemgået i faget 'Advanced Software Construction', skal du **ikke** håndtere nogen form for User Interface i frameworket.

*Som inspiration:*

*Du kan blive inspireret af at kigge på Greenfoot se [main page](#) eller deres [API references](#) . De arbejder med en verden (**world**) og med aktører (**actors**), De arbejder desuden med GUI (som altså er uden for denne opgave).*

## Aflevering (form og tidspunkt):

Denne opgave er individual og skal aflevere opgaven individuelt i wiseflow – men du må gerne snakke med andre og udveksle ideer og løsningsformer.

**Arbejdsperioden for denne opgave er 5 marts til 16 april**

Tirsdag den 16 April

Du skal demonstrere dit framework til underviserne mellem 9:10-15(16). Du skal desuden uploade en url til dit GitHub repository i wiseflow.

Du skal til den 16 april forberede en demo/præsentation. En fælles plan for præsentation på dagen kommer i begyndelsen af april.

## Detaljeret Beskrivelse af opgaven:

*(ændringer og præciseringer kan forekomme undervejs i opgaveforløbet)*

Du skal lave et bibliotek (Library) med klasser der udgør et miniframework for et tur-baseret 2D-spil, hvor der er en verden(world), skabninger(creatures) og objekter(armor,våben,skatte eller lign).

Du **skal ikke skrive nogen rapport**, Du bør dog lave kommentarer i koden.

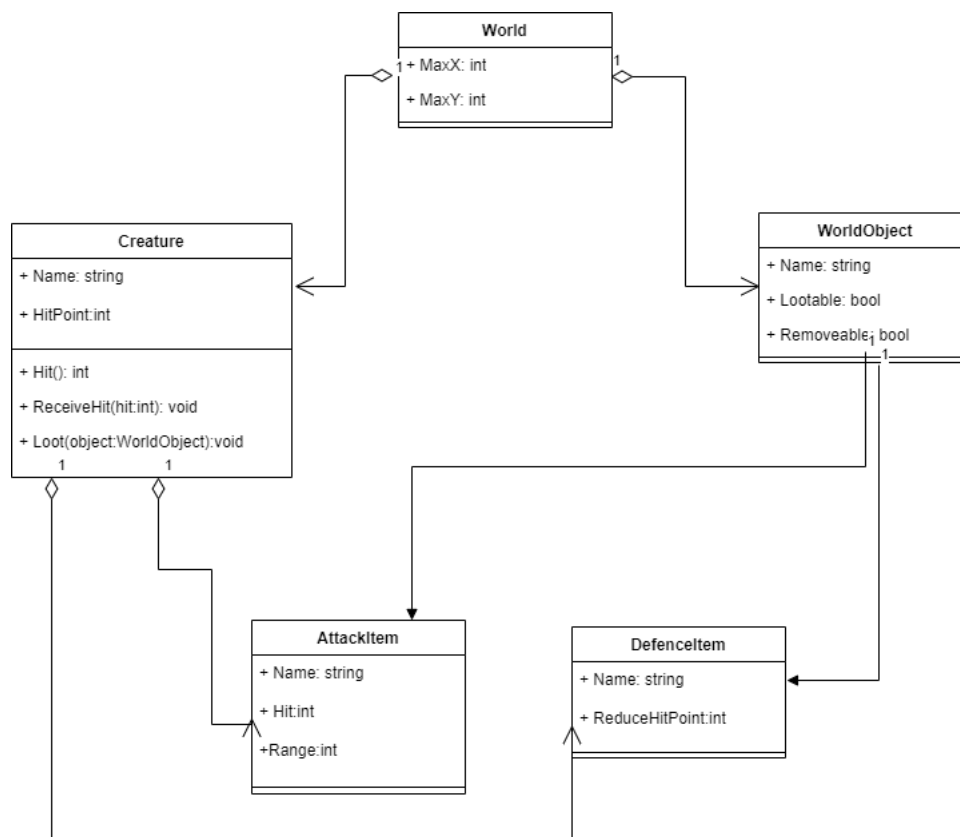
Du skal designe og implementere **et framework**, dvs. et bibliotek(Library), der kan understøtte andre spil-udviklere i deres arbejde med at lave et turbaseret 2D spil.

Din løsning skal kunne tilgås fra et GitHub-repository eller lign.

I første trin skal du implementere følgende objekter (- se også næste side):

- En verden (A World) (en 2D playground)
- Skabninger (Creatures, som har en position i verdenen)
- Objekter der har en fast position fx. En mur  
Nogle objekter skal kunne samles op/fjernes fra verdenen, mens andre kan give bonus (xp eller lign) eller skade.  
(nye våben, armor, hitpoints, ...)
- Våben (Attack) objekter (våben som bue, dolk eller magi, ...)
- Forsvars våben (armor) (skjolde, støvler, ...)

Som start kan du implementer design klasse diagrammet herunder:



- **World**  
(f.eks: properties størrelse af verdenen maxX, maxY + liste af Creatures/WorldObjects)
- **Creature**  
(f.eks: properties list of attack/defence items, + metoderne Hit, ReceiveHit, Loot)
- **WorldObject**  
(f.eks: hvor er de placeret i verdenen (world), og kan de fjernes(removable))
- **Attack Item**  
(f.eks: hvordan er de placeret hos Creature, Kan de findes I 'skattekister' eller I bonus bokse; evt. properties hit point, range, name)
- **Defence Object**  
(f.eks: hvordan er de placeret hos Creature, Kan de findes I 'skattekister' eller I bonus bokse; evt. properties reduceHitPoint, name)

### Nogle funktioner

- En skabning (Creature) kan ramme/slå (hit) andre skabninger(creatures)
- En skabning (Creature) kan opsamle et objekt (hvis man altså kan samle det op (lootable)), herved kan skabninger (creature) få flere eller bedre våben (skjolde, magi, flere hitpoint el.lign.).
- En skabning (Creature) kan modtage et angreb(hit) og evt. dø hvis life-point(hitpoint) bliver nul eller negativ.

## Krav til forbedringer af framework'et

Du har lavet det basis arbejdet (initial work), hvor du har implementeret klasserne fra klassediagrammet herover – gør det simpelt.

Nu skal du trin for trin (agilt) refaktorere framework'et så det bliver gradvis mere og mere fleksibelt for brug og ændringer gennem brug af forskellige teknikker og værktøjer du har lært i dette valgfag. (for hvert emne refereres til den uge det er gennemgået)

- Framework'et skal kunne **Konfigureres** fra en konfiguration-fil (uge 6)
- Framework'et skal kunne understøtte **tracing/logging** af beskeder (uge 6)
- Framework'et skal være **dokumenteret** f.eks. ///-kommentarer (uge 5)
- Framework'et skal følge principperne fra **SOLID** (uge 11)
- Framework'et skal gøre brug af iterationer og **LINQ** (uge 14)
- Framework'et skal indeholde mindst tre **Design Pattern** (uge 5, uge 10, uge 12) f.eks. blandt disse:
  - Template
  - State
  - Composite
  - Observer
  - Decorator
  - Strategy
  - Factory (Abstract Factory)
- Desuden kan (ikke skal) Framework'et indeholde
  - **Reflection** (uge 9)
  - **Regular expression** (uge 7)
  - **Operator overload** (uge 14)
  - **Path finding** (uge 15)

### Desuden overvej følgende emner:

- Placering skal 'playground' I verdenen være en matrix af x og y koordinater eller skal all creatures og world-objects have en x og y koordinat.
- Når en skabning slår kan det så være summen af en række våben eller er der kun et våben
- Når en skabning modtager et slag, bliver det så reduceret med en række forsvarsvåben eller er der kun et.
- Kan der være specielle tilstande, hvor skabningen kan f.eks. have super kræfter eller er svækket (Ligesom Mario).
- Skal alle objekter og skabninger (creatures) have et unikt Id.
- Er der en grænse for hvad en skabning kan bærer af våben og forsvarsvåben.
- Kan en skabning (Creature) skifte mellem de forskellige våben den bærer.
- Skal der være forskellige niveauer/kategorier af skabninger (creatures).

### Når du HAR lavet dit Framework – prøv det MEGET simpelt

- Du skal udover designe og implementere Framework'et I et Library prøve meget simpelt om det ser ud til at virke. (noget med en consol-applikation)
- Du skal I program.cs oprette nogle konkrete klasser som arver/implementere interface og eller klasser fra dit Framework. Herved kan du vise styrken ved dit Framework og hvor let det er at få noget til at virke.
- **Du skal IKKE** lave et helt spil !!! **Du skal fokusere på Framework'et.**

### *Til dem som hurtigt kommer langt – så er der flere udfordringer*

- Understøt et game loop, hvor hvert loop indebærer at en skabning (Creature) bevæger sig, samler objekter op eller kæmper mod en anden skabning.
- Skabninger (Creatures) kan bevæge sig, hvor de altid tager den korteste vej f.eks. X retning=3 frem mens Y retning = -10 tilbage, du kan tjekke om der er forhindringer (obstacles) undervejs i bevægelsen.