

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.

Du skal besvare opgave gennem de emner vi har gennemgået i faget 'Advanced Software Construction'.

Aflevering (form og tidspunkt):

Denne opgave er individuel og skal aflevere opgaven individuelt I Wiseflow – men du må gerne snakke med andre og udveksle ideer og løsningsformer.

Arbejdsperioden for denne opgave er 7 oktober til 11 november

[Tirsdag den 11 november](#)

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

Du skal til den 11. november forberede en demo/præsentation. En fælles tidsplan for præsentation kommer senere.

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**.

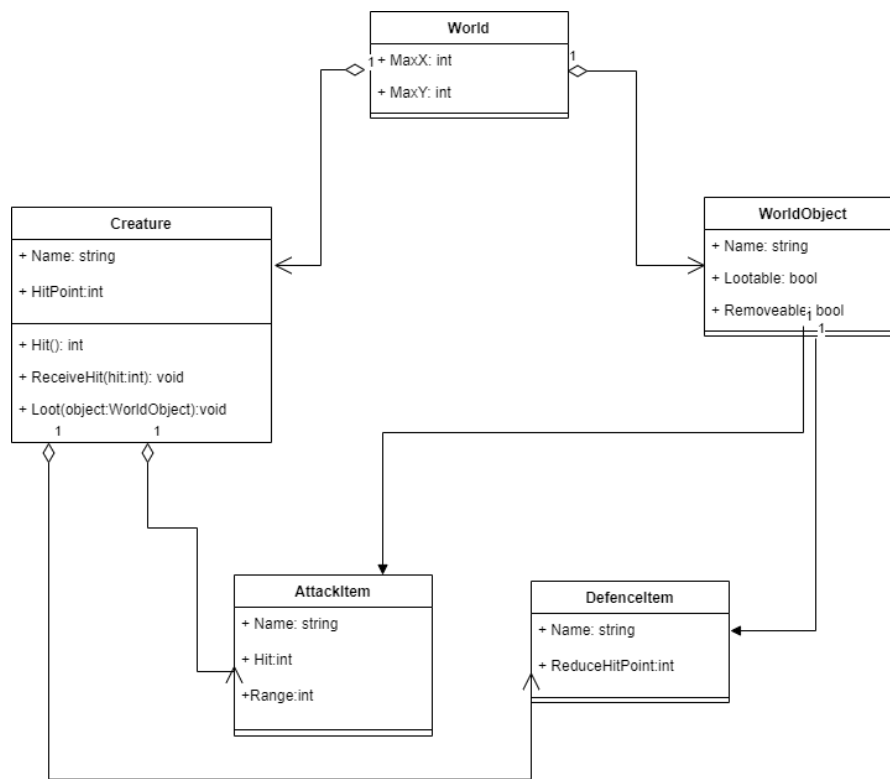
Du skal designe og implementere **et framework**, dvs. et bibliotek (Library), der kan understøtte **andre** spil-udviklere i deres arbejde med at lave et turbaseret 2D spil.

Din løsning skal kunne tilgås gennem en NuGet-pakke.

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, en skattekiste eller en eliksir
Nogle objekter skal kunne samles op og evt. 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 benytte design klassediagrammet herunder
(download et library med klasserne : <https://pele-easj.dk/2025e-ASWC/exercises/Mandatory2DGameFramework.zip>):



- **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.

Den egentlige Opgave: Krav til forbedringer af framework'et

Du har lavet det basis arbejdet (initial work) eller benyttet zip-filen ovenover, hvor klasserne fra klassediagrammet herover er implementeret.

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).

Du kan derfor ændre på klasserne, lave nye klasser eller det du nu har brug for.

- Opgaven skal løses i C#
- Framework'et skal kunne **Konfigureres** fra en konfiguration-fil, dvs. indlæse værdier fra en **XML-fil** (uge 37). Du skal som minimum kunne indlæse world size dvs. MaxX og MaxY, samt spil niveau (begynder, normal, ekspert.)
- Framework'et skal kunne understøtte **tracing/logging** af beskeder, dvs. du skal lave en klasse MyLogger, der er en **singleton** og hvor du kan tilmelde hhv. afmelde TraceListeners, samt logge forskellige informationer (uge 37). Du skal sikre at loggingen er fleksibel, og du skal bl.a. kunne tilføje og fjerne TraceListeners. Ingen Console.WriteLine i Librarie'et.
- Framework'et skal være **dokumenteret** med ///-kommentarer, mindst 3 klasser Samt generer en index.html-fil lavet med doxygen (uge 36)
- Framework'et skal følge principperne fra **SOLID** (uge 43), alle 5 principper.
- Framework'et skal indeholde Forskellige **Design Pattern** (uge 36, uge 39, uge 44):
 - 'MyLogger' skal være en **singleton**
 - Creature skal være en **Template**
 - Creature skal understøtte **Observer**, til at notificere hver gang det bliver ramt (Hit)
 - Creature skal implementere mindst én metode som **Strategy**
 - Defenceltem skal have en **Decorator**-klasse til f.eks. at booste forsvar eller svække forsvar
 - Defenceltem skal have en **Composite** klasse
 - Du skal benytte **Factory** til at generere forskellige game-objekter
 - Desuden kan du overveje at anvende **State** eller andre Patterns

- Framework'et skal forberedes og deployes som NuGet-pakke til NuGet-store (uge 45)
- Du skal mindst et sted i Framework'et vise Operator overload (uge 41)

Desuden kan (*ikke skal*) Framework'et benytte **Reflection** | **Regular expression**

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 – kort sagt hvor mange våben kan en creature have ad gangen.
- Når en skabning modtager et slag, bliver det så reduceret med en række forsvarsvåben eller er der kun et. Lidt det samme kan en creature f.eks. have både hjelm og brynje på samme tid.
- 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 en Console-application, der gør brug af dit Framework, for at prøve meget simpelt om det ser ud til at virke.
- 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 må gerne vise resultaterne med Console.WriteLine() i console-applikationen
- **Du skal IKKE lave et helt spil !!! Du skal fokusere på Framework'et.**

Til dem som hurtigt kommer langt – så er der flere udfordringer (Ekstra opgaver)

- 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.