

# TDP005 Projekt: Objektorienterat system

## Designspecifikation

Författare

Alexander Jonsson, [alejo720@student.liu.se](mailto:alejo720@student.liu.se)

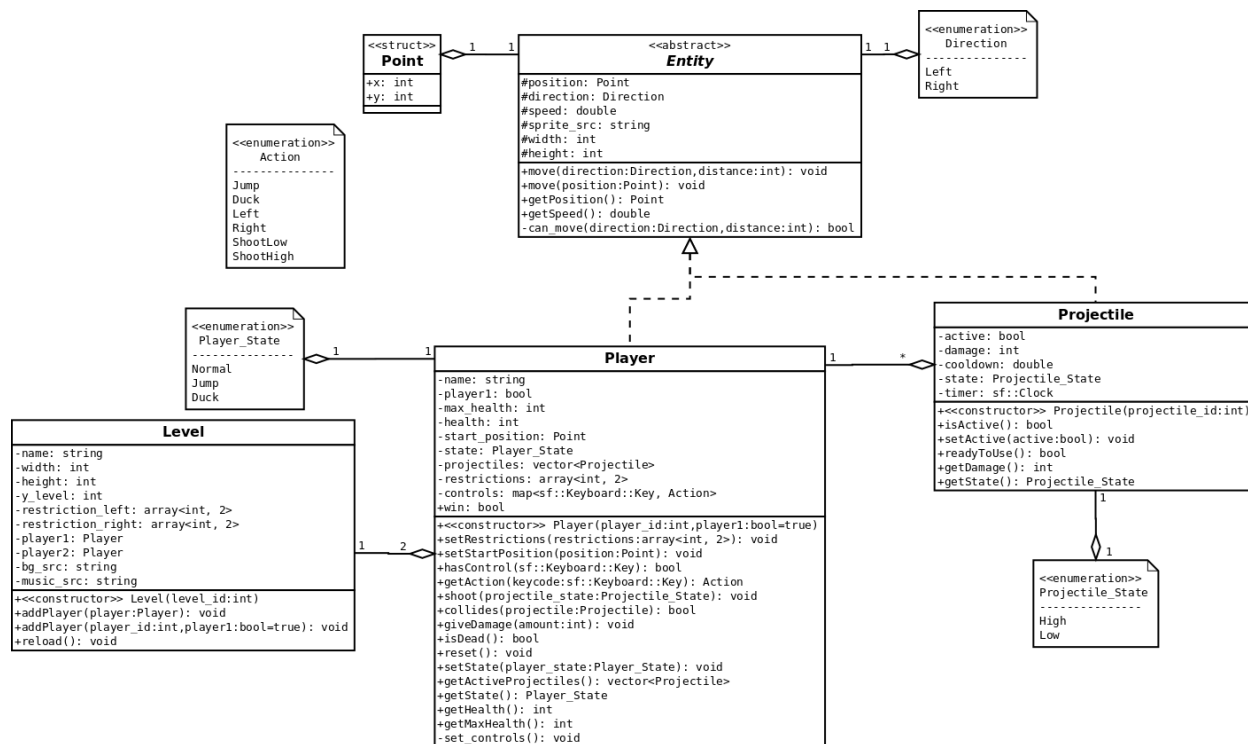
Joakim Johansson, [joajo229@student.liu.se](mailto:joajo229@student.liu.se)

## 1 Revisionshistorik

Ver.	Revisionsbeskrivning	Datum
1.0	Första utkast med klassdiagram, externa filer samt diskussion om valet av design.	2016-11-25

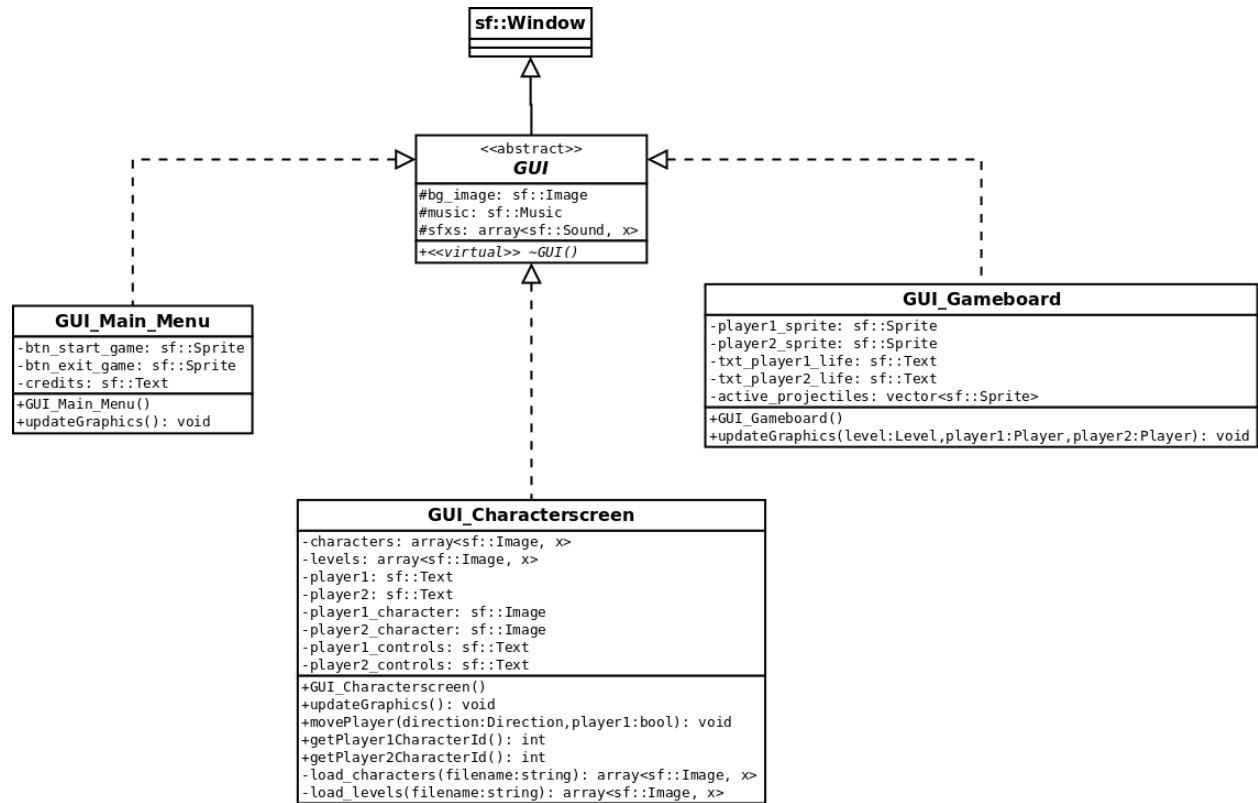
## 2 UML klassdiagram av systemet

Här visas ett UML klassdiagram för den logiska delen av systemet:



Figur 1: Klassdiagram: logisk del

Här visas ett UML klassdiagram för den grafiska delen av systemet:



Figur 2: Klassdiagram: grafisk del

## 3 Detaljb beskrivning

### 3.1 Player

Kommer snart...

### 3.2 Level

Kommer snart...

## 4 Diskussion av valet om design

### 4.1 Fördelar

- Klasser ärver andra klasser. Exempelvis ärver 'Player' och 'Projectile'-klasserna utav 'Entity'-klassen vilket leder till återanvändning av kod. Det gör det även lättare att skapa en liknande klass senare.
- De grafiska klasserna innehåller endast grafiska komponenter vilket leder till separation mellan det grafiska och logiska. Detta gör det möjligt att det grafiska eller logiska ska kunna användas till andra system.

- Att läsa objekts egenskaper från filer leder till mindre kod då vi slipper hårdkoda objektens egenskaper inuti koden.

### 4.2 Nackdelar

- Karaktärens projektiler skapas och lagras inuti spelarobjektet. I detta fall fungerar det bra då det är få projektiler som ska lagras. Om det skulle behöva lagras flera skulle detta vara minneskrävande. Det skulle då vara bättre att skapa objektet endast då det behövs.

- I spelarobjektet lagras det internt om spelaren är spelare 1 eller 2. Detta begränsar spelarklassen ifall det skulle behövas fler än två spelare.

## 5 Externa filer

### 5.1 Mappstruktur

```
resources
  img
    level
      *__portrait.png
      *.png
    character
      *__portrait.png
      *.png
    projectile
      *.png
  data
    levels.json
    characters.json
    projectiles.json
    *.json
  music
    *.mp3
  sfx
    *.wav
*.cc
*.h
settings.config
```

I 'root'-mappen ligger alla 'C++'-filer, konfigurationsfiler och en mapp som innehåller alla statiska filer som spelet ska ladda information från. Exempel på denna information är bilder, musik, ljudfiler och data som krävs för de olika objekten. '\_\_portrait'-bilderna är de bilder som ska visas vid val av karaktär. Alla data filer som ska användas är i 'JSON'-format vilket kräver att vi laddar ner en modul för att hantera 'JSON'-filer i C++.

## 5.2 Nivåer

Detta är ett exempel på hur en 'JSON'-fil innehållandes de olika nivåernas data kan se ut:

```
[
  {
    "id": 1,
    "name": "Avalanche Leftovers",
    "width": 800,
    "height": 640,
    "y_level": 100,
    "restriction_left": [20, 350],
    "restriction_right": [400, 780],
    "bg_src": "avalanche_leftovers.png",
    "music": "icy_win.wav"
  },
  {
    "id": 2,
    "name": "Firelands",
    "width": 1024,
    "height": 1280,
    "y_level": 20,
    "restriction_left": [100, 200],
    "restriction_right": [824, 924],
    "bg_src": "firelands.png",
    "music": "scorching_battle.wav"
  }
]
```

## 5.3 Karaktärer

Detta är ett exempel på hur en 'JSON'-fil innehållandes de olika karaktärernas data kan se ut:

```
[
  {
    "id": 1,
    "name": "Crazy guy",
    "max_health": 1000,
    "speed": 1.2,
    "projectiles": [1, 2],
    "sprite_src": "crazy_guy.png",
    "img": "crazy_guy_portrait.png"
  },
  {
    "id": 1,
    "name": "Less crazy guy",
    "max_health": 750,
    "speed": 1.5,
    "projectiles": [3, 4],
    "sprite_src": "less_crazy_guy.png",
    "img": "less_crazy_guy_portrait.png"
  }
]
```

## 5.4 Projektiler

Detta är ett exempel på hur en 'JSON'-fil innehållandes de olika projektilernas data kan se ut:

```
[
  {
    "id": 1,
    "projectile_name": "Amazing projectile",
    "projectile_state": "low",
    "damage": 25
    "speed": 3.0,
    "cooldown": 3,
    "sprite_source": "projectile_name.png"
  },
  {
    "id": 2,
    "projectile_name": "Awesome projectile",
    "projectile_state": "high",
    "damage": 100,
    "speed": 1.0,
    "sprite_source": "projectile_name.png",
    "cooldown": 6
  }
]
```

## 5.5 Konfigurationsfil

Detta är strukturen i en konfigurationsfil:

```
player1_jump 'w'
player1_left 'a'
player1_duck 's'
player1_right 'd'
player1_low_shoot 'r'
player1_high_shoot 't'
player2_jump 'up-arrow'
player2_left 'left-arrow'
player2_duck 'down-arrow'
player2_right 'right-arrow'
player2_low_shoot 'o'
player2_high_shoot 'p'
```