

# TDP005 Projekt: Objektorienterat system

## Designspecifikation

Författare

Elliot Johansson, [elljo130@student.liu.se](mailto:elljo130@student.liu.se)

Lukas Freyland, [lukfr510@student.liu.se](mailto:lukfr510@student.liu.se)

Nadim Lakrouz, [nadla777@student.liu.se](mailto:nadla777@student.liu.se)

## 1 Revisionshistorik

| Ver. | Revisionsbeskrivning | Datum      |
|------|----------------------|------------|
| 1.0  | Första utkast        | 2022-11-25 |

## 2 Player

Syftet med Playerklassen är att representera den karaktär spelaren styr. Spelaren styr vilka formler player-karaktären använder. Spelaren kommer inte förflytta sig på planen. Vissa formler kommer behöva riktas för att träffa fienderna medan andra inte behöver det. Funktionen `regenerate_mana` kommer sakta återställa manapoäng med `sf::Clock`.

### 2.1

- `int hp` - privat variabel som tilldelas ett heltal.
- `double mana` - privat variabel som tilldelas ett flyttal.
- `sf::Clock clock` - sfml variabel för att hantera tiden.
- `int set_hp()` - ändrar privata variabeln `hp`.
- `int get_hp()` - hämtar privata variabeln `hp`.
- `int get_mana()` - hämtar privata variabeln `mana`.
- `void regenerate_mana()` - ökar privata variabeln `mana` över tid.

## 3 Enemies

"Enemies" klassens syfte är att representera fienderna som i intervaller kommer förflytta sig mot spelaren på spelplanen. Enemies har en `check_collision` funktion som anropas när fiender och spelare kolliderar.

### 3.1

- `double hp` - privat variabel som tilldelas ett flyttal.
- `double resistance` - privat variabel som tilldelas ett flyttal.
- `void check_collision()` - kollar kollision med spelaren och magiska formler.
- `int set_hp()` - ändrar privata variabeln `hp`.
- `int get_hp()` - hämtar privata variabeln `hp`.
- `int get_resistance()` - hämtar privata variabeln `resistance`.
- `void ai_movement()` - styr fiende rörelse.

## 4 Spells\_Handler

- `pair<string, int> mana_per_spell` - lista med hur mycket mana varje magiska formel kostar.
- `void handle_input()` - hanterar spelarens inmatning.
- `void handle_combination()` - hanterar kombinationer av magiska formler.

- void handle\_\_damage() - beräknar och hanterar damage.
- void control\_spell() - navigerar spells med hjälp av mus.

## 5 Diagram

