

# TDP005 Projekt: Objektorienterat system

## Designspecifikation

Författare

Mohammad Omar, [mohom373@student.liu.se](mailto:mohom373@student.liu.se)

## Revisionshistorik

Ver.	Revisionsbeskrivning	Datum
0.1	Skapade första utkast av designspecifikation	191129
0.2	Redigering efter kommentarer av labbassistent	191204
0.3	Sista inlämning	191220

## 1 Detaljbeskrivning

### 1.1 Player

Player-klassen ser till att spelaren har ett spelarobjekt som hen kan styra. Spelarobjektet tar form av en bräda och styrningen sker endast i horisontell led med tangentbordsinmatning. Syftet med spelarobjektet är att försöka stoppa ett Ball-objekt från att träffa DeadZone som är en klass som ser bland annat till spelaren förlorar liv. När spelarobjektet når höger eller vänster vägg uppstår en kollision och brädan hindras från att längre kunna förflytta sig i samma riktning som väggen den kolliderar med.

Player-klassen ärver av Movable-klassen och Entity-klassen.

Dessa variabler och metoder ärver Player-klassen av Entity-klassen:

- width : int - En variabel som anger brädans nuvarande bredd.
- height : int - En variabel som anger brädans nuvarande höjd.
- collision(Playstate &playstate) = 0: void - En funktion som kallas när en kollision har uppstått.
- update() = 0: void - En funktion som updaterar entity.

Dessa variabler och metoder ärver Player-klassen av Movable-klassen:

- physics\_object: Physics\_Object
- movement(sf::Vector2f &position) override: void

## 1.2 Ball

Syftet med Ball-klassen, vilket i det här fallet är en projektil som tar formen av en kvadrat, är att samla poäng genom att träffa poängzonerna som finns på spelplanen. Ball-klassen har också som ett syfte att vara ett hinder för spelaren då hen förlorar ett liv när ett Ball-objekt kolliderar med en Deadzone. Bollen styrs inte av spelaren. Dess riktning bestäms av infallsvinkeln vid kollision med olika objekt.

Ball-klassen ärver av Movable-klassen och Entity-klassen.

Dessa variabler och metoder ärver Ball-klassen av Entity-klassen:

- width : int - En variabel som anger brädans nuvarande bredd.
- height : int - En variabel som anger brädans nuvarande höjd.
- collision(Playstate &playstate) = 0: void - En funktion som kallas när en kollision har uppstått.
- update() = 0: void - En funktion som updaterar entity.

Dessa variabler och metoder ärver Ball-klassen av Movable-klassen:

- physics\_object: Physics\_Object
- movement(sf::Vector2f &position) override: void

Ball-klassens egna funktioner:

- handle\_collision(sf::Vector2f &position) : void

## 2 Diskussion av designen

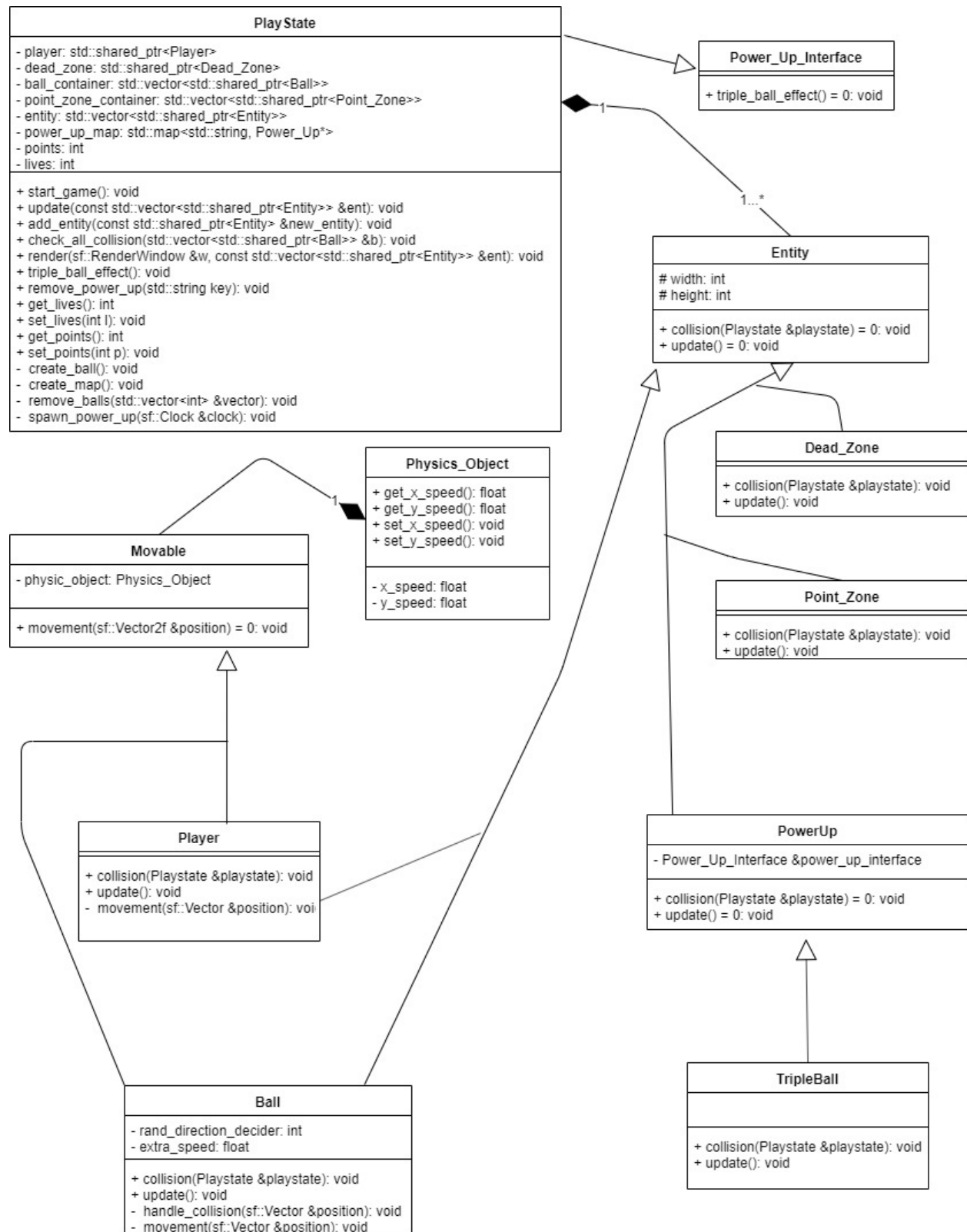
Nuvarande klassstruktur ser till att det är enkelt att utveckla vidare med nya klasser. Klassen PowerUp som endast innehåller TripleBall för tillfället, kan exempelvis utvecklas med en ny power-up utan större problem genom att låta den nya klassen ärva av PowerUp klassen.

## 3 Externa filformat

Det externa filformatet som kommer användas för inläsning av spelplan är en textfil(.txt). Innehållet av textfilen är dock inte bestämt vid det här stadiet.

## 4 Klassdiagram

Nedan visas klassdiagramet.



Figur 1: UML - klassdiagram