# Схема наследования классов



# Описание полей и методов

## Класс “GameObject”

### Функции:

#### virtual void Update(Game\* const scene);

Пустая виртуальная функция

##### Параметры:

**scene** - сцена, в которой происходят события

##### Возвращаемое значение:

Нет

#### virtual void onPositionUpdate\_();

Функция, срабатывающая на изменение размеров и позиции

##### Параметры:

Нет

##### Возвращаемое значение:

Нет

#### void SetSpriteX\_(float x);

Задаёт координату X у спрайта

##### Параметры:

**x** - расстояние от левой границы экрана

##### Возвращаемое значение:

Нет

#### void SetSpriteY\_(float y);

Задаёт координату Y у спрайта

##### Параметры:

**y** - расстояние от нижней границы экрана

##### Возвращаемое значение:

Нет

### Поля:

**sprite\_** - Спрайт, с которым связан объект

**x\_** - Расстояние объекта от отсчитываемого края

**y\_** - Расстояние объекта от низа окна

**w\_** - Ширина объекта

**h\_** - Высота объекта

**frameWidth\_** - Ширина окна (для отсчёта от правого края)

**isRightAlignment\_** - Отсчёт от правого края?

**prevX\_, prevY\_, prevW\_, prevH\_, prevFrameWidth\_, prevIsRightAlignment\_** - Вспомогательные переменные для обнаружения изменений

## Класс “CastleObject”

### Функции:

#### virtual void Update(Game\* const scene) override;

Вызывает Update родительского класса

##### Параметры:

**scene** - сцена, в которой происходят события

##### Возвращаемое значение:

Нет

#### void Damage(float power);

Наносит урон данному объекту

##### Параметры:

**power** – урон, который необходимо нанести объекту

##### Возвращаемое значение:

Нет

#### virtual void onPositionUpdate\_();

Функция, срабатывающая на изменение размеров и позиции

##### Параметры:

Нет

##### Возвращаемое значение:

Нет

#### void InitLabels\_();

Инициализирует надписи

##### Параметры:

Нет

##### Возвращаемое значение:

Нет

#### void UpdateLabelsPosition\_();

Перемещает надписи в соответствии с размерами и позицией объекта

##### Параметры:

Нет

##### Возвращаемое значение:

Нет

### Поля:

**health\_** - Количество жизней

**maxHealth\_** - Максимальное количество жизней

**armor\_** - Количество брони

**maxArmor\_** - Максимальное количество брони

**healthLabel\_** - Надпись, отображающая количество жизней

**armorLabel\_** - Надпись, отображающая количество брони

## Класс “DynamicObject”

### Функции:

#### virtual void Update(Game\* const scene) override;

Изменяет позицию объекта в соответствии со скоростью

##### Параметры:

**scene** - сцена, в которой происходят события

##### Возвращаемое значение:

Нет

### Поля:

**hSpeed\_** - Горизонтальная скорость

**vSpeed\_** - Вертикальная скорость

## Класс “AttackerObject”

### Функции:

#### virtual void Update(Game\* const scene) override;

Вызывает Update родительского класса и отрабатывает столкновение с другими объектами

##### Параметры:

**scene** - сцена, в которой происходят события

##### Возвращаемое значение:

Нет

#### void Attack(Game\* const scene);

Атакует другие объекты с помощью функции Damage(), если данный объект находится на достаточно малом расстоянии от них

##### Параметры:

**scene** - сцена, в которой происходят события

##### Возвращаемое значение:

Нет

#### void Damage(float power);

Наносит урон данному объекту

##### Параметры:

**power** – урон, который необходимо нанести объекту

##### Возвращаемое значение:

Нет

#### virtual void onPositionUpdate\_();

Функция, срабатывающая на изменение размеров и позиции

##### Параметры:

Нет

##### Возвращаемое значение:

Нет

#### void InitLabel\_();

Инициализирует надпись

##### Параметры:

Нет

##### Возвращаемое значение:

Нет

#### void UpdateLabelPosition();

Перемещает надпись в соответствии с размерами и позицией объекта

##### Параметры:

Нет

##### Возвращаемое значение:

Нет

### Поля:

**health\_** - Количество жизней

**maxHealth\_** - Максимальное количество жизней

**power\_** - Урон, который наносит данный объект

**label\_** - Надпись, отображающая количество жизней

## Аналогичные функции различных классов

### Функции:

#### static size\_t GetClassCost();

Статический константный метод

##### Параметры:

Нет

##### Возвращаемое значение:

Стоимость объекта данного класса

#### virtual size\_t GetCost() const override;

Вызывает своего GetClassCost() класса

##### Параметры:

Нет

##### Возвращаемое значение:

Стоимость объекта