

```
#include "stdafx.h"

#include "Enemy.h"

Enemy::Enemy(sf::Sprite* sp)
{
    srand(time(NULL));
    m_counter = 0;

    m_visible = false;

    m_sprite = sp;

    m_speed = 20.f;

    m_clock;
    m_Time = m_clock.getElapsedTime();

    m_TimerSpeed = 0.0f;

    m_x = 0.0f;
    m_y = 0.0f;

    m_collider = new Collider(m_x, m_y);
    m_collider->SetParent(this);
    m_collider->SetWidthHeight(m_sprite->getGlobalBounds().width, m_sprite->getGlobalBounds().height);

    m_type = ENTITY_ENEMY;
    SetVisible();
}

Enemy::~Enemy()
{
}

void Enemy::Update(float deltatime)
{
    m_Time = m_clock.getElapsedTime();
    m_TimerSpeed = m_Time.asSeconds() / 10;

    m_speed = 20.0f + m_TimerSpeed;

    if (m_visible)
    {
        m_y += m_speed * deltatime;
    }

    m_collider->Refresh();

    m_sprite->setPosition(m_x, m_y);

    //m_TimerSpeed++;
}

sf::Sprite* Enemy::GetSprite()
{
    return m_sprite;
}

Collider* Enemy::GetCollider()
{
}
```

```
    return m_collider;
}

float Enemy::GetX()
{
    return m_x;
}
float Enemy::GetY()
{
    return m_y;
}

void Enemy::SetInvisible()
{
    m_visible = false;
}
void Enemy::SetVisible()
{
    if (m_visible == false)
    {
        m_visible = true;

        Positions* pos = ServiceLocator<Positions>::GetService();

        m_pos = rand()% 5 ;

        SetPosition(pos->GetLanes(m_pos).x, pos->GetLanes(m_pos).y - rand()% 220 - m_sprite->getGlobalBounds
        (.height);
    }
}

bool Enemy::IsVisible()
{
    return m_visible;
}

EEntityType Enemy::GetType()
{
    return m_type;
}

void Enemy::SetType(int entity)
{
    m_type = (EEntityType)entity;
}

void Enemy::SetPosition(float x, float y)
{
    m_x = x;
    m_y = y;
    m_collider->Refresh();
}

/* bool IsGrounded() = 0;
void SetGrounded(int i) = 0;*/

bool Enemy::IsHolding()
{
    return false;
}

void Enemy::Hold(bool yes)
{
}

Weakness Enemy::GetWeakness()
{
    return (Weakness)1;
}
```