```
Game_Object
 # sf::Vector2f position
 # int width
 # int height
 # sf::Sprite sprite
 + Game_Object(float
  x, float y, int w,
  int h)
  + virtual ~Game_Object
 ()=default
 + virtual void update()
  + virtual void render
 (sf::RenderWindow &window,
   Camera &camera)
 + virtual sf::FloatRect
  get rect() const
 + sf::Vector2f get_position
 () const
                Δ
         Moving_Object
# bool jumping
# sf::Vector2f velocity
# sf::Vector2f starting
 position
# bool alive
# sf::Clock animation
 timer
# int animation_frame
# Animation_State current
 animation state
# int animation frame rate
+ Moving_Object(float
x, float y, int w, int h)
+ virtual void update
(float friction, float
gravity)=0
+ sf::Vector2f get_velocity
() const
+ sf::FloatRect get_rect
() const override
+ sf::Vector2f get_starting
 _position() const
+ void update_position()
+ virtual void reset()
+ bool is_alive() const
+ void kill()
+ void push(float force)
+ void blow(float force)
 + virtual void change
 _direction()
# virtual void update
 _animation_state()
# virtual void update
 texture()
                Δ
             Enemy
# float speed
+ Enemy(float x, float y)
+ Enemy(float x, float
y, int w, int h)
  void render(sf::RenderWindow
&window, Camera &camera)
override
+ void update(float friction,
float gravity) override
+ void spawn()
# void update_animation
state() override
                Δ
         Rolling_Stone
     Rolling_Stone(float
    x, float y)
     void update_texture
   () override
    + sf::FloatRect get_rect
   () const override
     void stop_vertical
   () override
   + void change_direction
   () override
   + void update animation
    _state() override
```