

1.

# GAME: SHEN

0...1

## J1EntityManager:

```

j1EntityManager();
~j1EntityManager();
bool Awake(pugi::xml_node& config);
bool Start();
bool PreUpdate();
bool Update(float dt);
bool PostUpdate();
bool CleanUp();
j1Entity* Get_Player();

bool Load(pugi::xml_node&);
bool Save(pugi::xml_node& const);

j1Entity* Summon_Entity(j1Entity::Types type, iPoint pos);
void Destroy_Entity(j1Entity* entity);
void Destroy_Entities();

pugi::xml_node node;
p2List<j1Entity*> entities;

```

0...N

## J1Entity:

```

enum class Types
{
    PLAYER,
    SKELETON,
    FLYING_ENEMY,
    NULL_ENTITY
};

Types entity_type;
j1Entity(Types type);
virtual ~j1Entity();
bool Load(pugi::xml_node& data);
bool Save(pugi::xml_node& data) const;
virtual void Pushbacks();
virtual void OnCollision(Collider* c1, Collider* c2);
virtual void GetPosition();

iPoint position;
int SpawnPointX, SpawnPointY;

SDL_Texture* Character_tex = nullptr;
bool Looking_Forward = true;
bool Can_input = true;
bool Alive = true;
bool Grounded = true;

//PATHFINDING
bool Following_Player = false;
const p2DynArray<iPoint*> path = nullptr;

p2SString entity_name;

```

0..N

## j1Player

```

//PLAYER, AND WORLD MAX VELOCITIES
int Spawn_X;
int Spawn_Y;
float World_Gravity;
float Speed_X;
float Speed_Y;
float Dash_Speed;
float Reduction_Speed;
float MAX_X;
float MAX_Y;
float MAX_JUMP;
float velocity;

bool Looking_Forward = true;
bool Can_Input = true;
bool Alive = true;
bool Dash = false;
bool Grounded = true;
bool God_Mode = false;

uint32 dash_timer;
int dashTime;
bool dashTimer = false;

//Attack timer
uint32 attack_timer;
bool attacking = false;
int attackTime;
bool attackTimer = false;

uint32 death_timer;
int deathTime;
bool deathTimer = false;

//PLAYER GOD-MODE
bool GodMode = false;
bool GodModeForward = false;
bool GodModeBackward = false;
bool GodModeUp = false;
bool GodModeDown = false;

Collider* playerhead = nullptr;
Collider* playerbody = nullptr;
Collider* playerfeet = nullptr;
Collider* playerattack = nullptr;

//PLAYER STATE MANAGER
void Player_Position();
void Player_State_Machine();
void Set_Player_State(states state);
void Restart();

public:
p2SString folder;
pugi::xml_node node;
PlayerInfo playerInfo;
input Input;
p2SString file;
bool keyup = true;
SDL_Texture* graphics = nullptr;
//STATES DON'T WORK INSIDE PLAYERINFO. SO THATS WHY ITS IN HERE
states state;

```

0... N

## j1Skeleton

```

bool pathfinding_ground();
iPoint position;
SDL_Texture* skeletonTex = nullptr;
bool Looking_Forward = true;
bool Alive = true;
bool Grounded = true;
bool Near_Player = false;

//ANIMATIONS
Animation* current_animation = nullptr;
Animation walking;
Animation walking2;
Animation death;
Animation death2;

//SPEED
float SpeedX = 1;
float SpeedY = 1;
float gravity;
float vel;

//COLLIDERS
Collider* skeletonCollider = nullptr;
void Pushbacks();
Skeleton_States state = SKELETON_FORWARD;

p2SString folder;
p2SString texture_path;
pugi::xml_node node;

```

0...N

## j1Flying\_Enemy

```

bool pathfinding();
void Flying_Enemy_State(Flying_Enemy_States state);
void Flying_Enemy_Position();

iPoint position;
SDL_Texture* Flying_Enemy_Tex = nullptr;
bool Looking_Forward = true;
bool Alive = true;
bool Grounded = true;

//ANIMATIONS
Animation* current_animation = nullptr;
Animation walking;
Animation walking2;
Animation death;
Animation death2;

//SPEED
float Speedx = 1;
float Speedy = 1;
float gravity;
float vel;

//COLLIDERS
Collider* Flying_Enemy_Collider = nullptr;
void Pushbacks();
Flying_Enemy_States state = FLYING_ENEMY_FORWARD;

p2SString folder;
p2SString texture_path;
pugi::xml_node node;

```