1 **Entities** App j1Player* player uint player_life iPoint loadPositionPlayer private: pugi::xml_node config EntityInfo queue[MAX ENTITIES] Entity* entities[MAX_ENTITIES] SDL_Texture* sprites p2DynArray<iPoint> flyPositions p2DynArray<iPoint> walkPositions j1Entities() ~j1Entities() bool Start() bool PreUpdate() bool Awake(pugi::xml node&) bool Update(float dt) bool PostUpdate() bool CleanUp() bool ResetEntities() bool Load(pugi::xml node&) bool Save(pugi::xml node&) const bool AddEntity(ENTITY_TYPES type, int x, int y, SDL_Rect colliderRect) void SpawnEntity(const EntityInfo& info)

```
Entity
float gravityf
float speed jump
fPoint speed
fPoint position
Animation* animation
bool death
uint enemy
Collider* collider
bool isPlayer
bool movement[]
int now
Entity ()
Entity (int x, int y)
virtual ~Entity()
virtual bool Awake(pugi::xml node&)
virtual bool Start()
virtual bool Update(float dt)
virtual bool CleanUp()
virtual bool Load(pugi::xml node&)
virtual bool Save(pugi::xml_node&) const
const Collider* GetCollider() const
virtual void NormalizeAnimations(float dt)
virtual void CalculateGravity(float dt)
virtual void Move(float dt)
virtual void Draw(SDL Texture* sprites)
virtual void OnCollision(Collider* collider)
virtual void Dead()
```

0...*

J1Player

```
j1Player()
j1Player(int x, int y, SDL Rect colliderRect)
~j1Player()
state istate
SDL_Texture * graphics
SDL RendererFlip flip
Animation* current animation
iPoint OriginPos
float velocityX
Timer timer
bool IsGodMode
int jumpCount
int dashCount
int godmodeCount
Animation animations
const char* audio name
uint audio uint
Collider* c_player
Collider* offSet
```

```
bool Awake(pugi::xml_node& conf)
bool Start()
bool Update(float dt)
bool CleanUp()
bool Load(pugi::xml_node&)
bool Save(pugi::xml_node&) const
state SetStateFromInt(int state_as_int)
void ResetPlayer()
```

Enemy_Walk

uint zombiesound
Animation animations
fPoint originalpos
bool isDead
int i
int now
bool current_in_path
p2DynArray<iPoint> enemy_path
Timer timer
bool movingLeft

Enemy_Walk(int x, int y, SDL_Rect colliderRect)
bool Awake(pugi::xml_node& config)
bool Save(pugi::xml_node& config) const
bool Load(pugi::xml_node& config)
void Move(float dt)
void Dead()
void NormalizeAnimations(float dt)
void CalculateGravity(float dt)

Enemy_Fly

bool Agro
Animation fly
Animation attack
iPoint originalpos
bool isDead =
int i
bool current_in_path
p2DynArray<iPoint> enemy_path
int now
Timer timer
bool movingLeft
uint flysound

Timer soundtimer

Enemy_Fly(int x, int y, SDL_Rect colliderRect)
bool Awake(pugi::xml_node& config)
bool Save(pugi::xml_node& config) const
bool Load(pugi::xml_node& config)
void Move(float dt)
void Dead()
void DrawPath()
void NormalizeAnimations(float dt)