

relevants ones to understand how the

game entities are done.

bool PostUpdate()
bool CleanUp()

bool Load(pugi::xml\_node&)
bool Save(pugi::xml\_node&) const

void OnCollision(Collider\* c1, Collider\* c2)

Entity enum entity state enum entity type entity state state entity\_type type iPoint position fPoint virtualPosition int pos\_relCam fPoint velocity bool colliding (bottom, left, right, up) SDL Texture\* graphics Animation\* animation Collider\* collider Collider\* collidingFloor float speed: float jump force; uint max jump value bool flying float scale iPoint sprite\_pos, collider\_pos, collider\_size Entity(const char\* name) bool Entity Update(float dt) void Entity\_OnCollision(Collider\* c1, Collider\* c2) virtual bool specificAnimation() Animation\* LoadAnimation(const char\* animationPath, const char\* animationName) void LoadLogic(const char\* animationPath) void setAnimation()

Player Bat bool god mode float step\_time Animation jump\_cloud bool Awake(pugi::xml node&) iPoint cloud offset bool Start() iPoint cloud pos bool Update(float dt) SDL\_Texture\* graphics\_god bool PostUpdate() Animation SSJ aura bool CleanUp() iPoint aura offset void OnCollision(Collider\* c1, Collider\* c2) bool double\_jump bool Load(pugi::xml\_node&) bool won bool Save(pugi::xml node&) const bool Start() bool Update(float dt)

bool Awake(pugi::xml\_node&)
bool Start()
bool Update(float dt)
bool PostUpdate()
bool CleanUp()
void OnCollision(Collider\* c1, Collider\* c2)
bool Load(pugi::xml\_node&)
bool Save(pugi::xml\_node&) const