```
Entity Manager
               struct PathInfo
Game
                                          Functions
               j1EntityManager()
               virtual ~j1EntityManager()
               bool Awake(xml node&)
               bool Start()
               bool PreUpdate()
               bool Update(dt)
               void UpdateEntity(dt)
               bool PostUpdate(dt)
               bool CleanUp()
               j1Entity * const CreateEntity(const char*, entity type)
               void DestroyEntity(j1Entity*)
               void OnCollision(Collider1*, Collider2*)
               Animation* LoadAnimation(const char*, const char*)
               Playerdata& GetPlayerData() { return playerinfo }
               SlimeData& GetSlimeData() { return slimeinfo }
               BatData& GetBatData() { return batinfo }
               bool Load(pugi::xml_node&)
               bool Save(pugi::xml_node&) const
                                          Variables
               p2List <j1Entity*> entities
               bool do logic
                      logic updates per second
               float update ms cycle
               float accumulated_time
                      entityID
               int
```

```
Entity
enum entity state
enum class entity_type
                        Functions
class j1Entity
j1Entity(const char*,entitytype) : manager, entitytype
void Init(j1EntityManager* manager)
virtual bool Start()
virtual bool Load(xml node&)
virtual bool Save(xml node&) const
virtual void FixedUpdate(foat dt)
virtual void LogicUpdate(dt)
virtual bool CleanUp()
virtual void OnCollision(Collider1* , Collider2*)
                        Variables
p2SString
fPoint
             position
fPoint
             Velocity
Collider*
             entitycoll
             entitycollrect
SDL Rect
float
             colliding_offset
float
             gravity
entity type entitytype
entity state entitystate
             entityID
int
Animation*
            CurrentAnimation
SDL Texture* spritesheet
j1EntityManager*
                     manager
```

## Assumptions

Entity manager was a struct with pathfinding information. We have two enums with entity types and states. Every entity has a struct with variables we do not list on the UML, those variables are loaded on the entity awake. Only the relevant variables to understand the functionality are shown.

Player

Playerdata playerinfo

SlimeData slimeinfo BatData batinfo

```
struct Playerdata
                     Functions
class j1Player
                                   void FixedUpdate(dt)
j1Player()
                                   void LogicUpdate(foat dt)
~j1Player()
                                   void OnCollision(Collider1*,
bool Start()
                                   Collider2*)
bool Update(dt)
                                   bool Load(xml node&)
bool PostUpdate(dt)
                                   bool Save(xml node&) const
bool CleanUp()
                     Variables
bool playercolliding
                                        bool wasRight
bool colliding_roof
                                        bool dead
bool colliding_floor
                                        bool initialmoment
bool double_jump
                                        bool first move
bool must fall
                                        bool god_mode
bool going_right
                                        int parallaxflow
bool going_left
                                        int previousflow
```

## Slime

0..2

## struct SlimeData **Functions** class j1Slime :public j1Entity j1Slime() bool Load(xml node&) ~j1Slime() bool Save(xml node&) const bool Start() bool ReestablishVariables bool Update(dt) bool CreatePathfinding(const iPoint) bool PostUpdate(dt) bool Pathfind(dt) bool CleanUp() void UpdateMovement(dt) void FixedUpdate(dt) void LogicUpdate(dt) void OnCollision(Collider1\*, Collider2\*) Variables bool dead const p2DynArray<iPoint>\* bool going\_right last pathfinding bool going left uint pathfinding\_index bool must\_fall uint pathfinding\_size bool slimecolliding PathInfo\* path\_info SlimeData Slimeinfo

0..2

```
Bat
struct BatData
                     Functions
class j1Bat
j1Bat()
                         bool Load(xml node&)
~j1Bat()
                         bool Save(xml node&) const
bool Start()
                         bool ReestablishVariables
bool Update(dt)
bool PostUpdate(dt)
                         bool CreatePathfinding(const iPoint)
bool CleanUp()
                         bool Pathfind(dt)
void FixedUpdate(dt)
                         void UpdateMovement(dt)
void LogicUpdate(dt)
void OnCollision(Collider1*, Collider2*)
                     Variables
bool dead
                        const p2DynArray<iPoint>*
bool going_right
                        last pathfinding
bool going_left
                        uint pathfinding index
bool going up
                        uint pathfinding size
bool going down
                        PathInfo* path info
bool batcolliding
BatData BatInfo
```