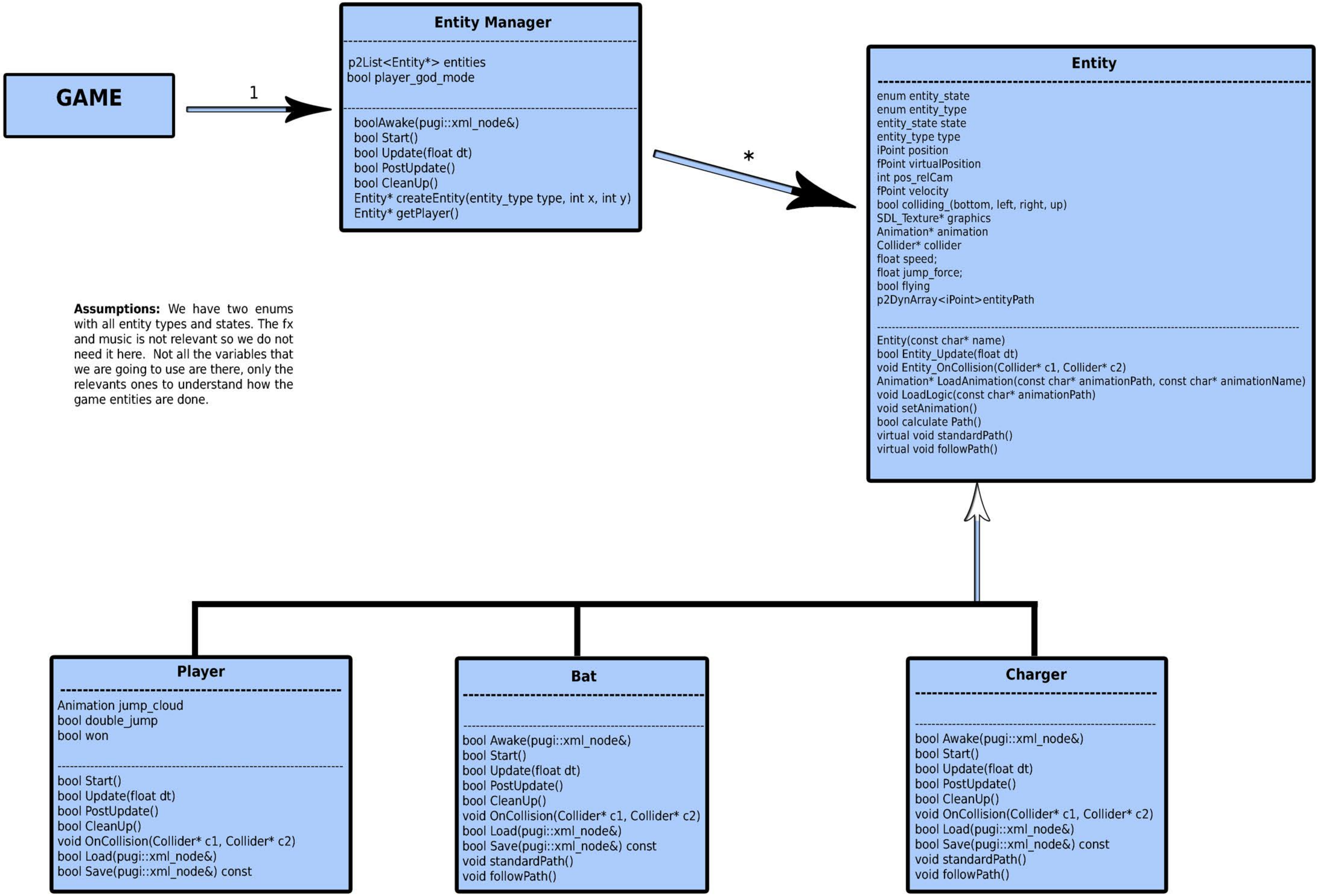

GAME

Entity Manager

p2List<Entity*> entities
bool player_god_mode

boolAwake(pugi::xml_node&)
bool Start()
bool Update(float dt)
bool PostUpdate()
bool CleanUp()
Entity* createEntity(entity_type type, int x, int y)
Entity* getPlayer()

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
float speed;
float jump_force;
bool flying
p2DynArray<iPoint>entityPath

Entity(const char* name)
bool Entity_Update(float dt)
void Entity_OnCollision(Collider* c1, Collider* c2)
Animation* LoadAnimation(const char* animationPath, const char* animationName)
void LoadLogic(const char* animationPath)
void setAnimation()
bool calculate Path()
virtual void standardPath()
virtual void followPath()

Assumptions: We have two enums with all entity types and states. The fx and music is not relevant so we do not need it here. Not all the variables that we are going to use are there, only the relevants ones to understand how the game entities are done.

Player

Animation jump_cloud
bool double_jump
bool won

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

Bat

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
void standardPath()
void followPath()

Charger

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
void standardPath()
void followPath()