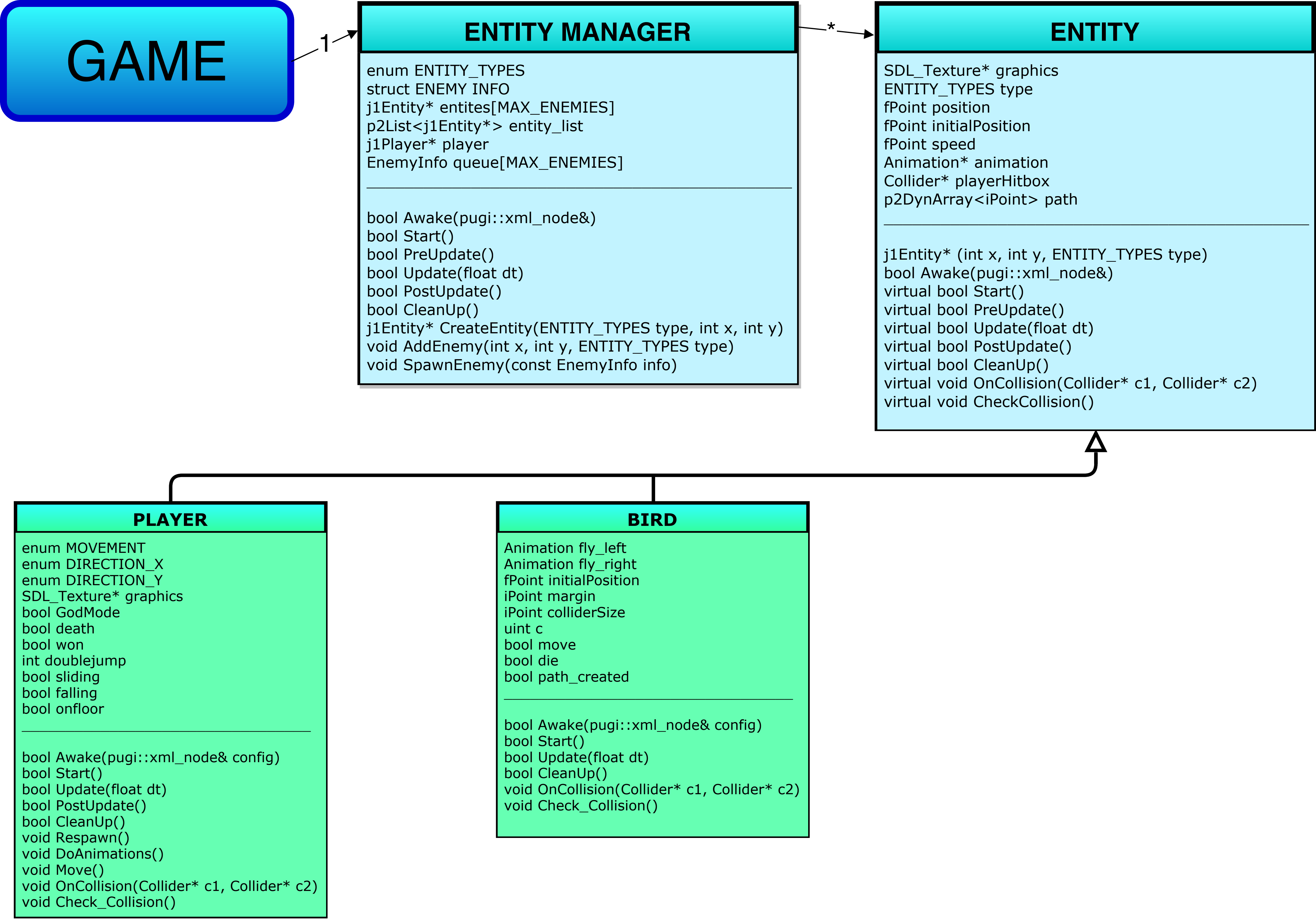

ENTITY MANAGER

enum ENTITY_TYPES
struct ENEMY INFO
j1Entity* entites[MAX_ENEMIES]
p2List<j1Entity*> entity_list
j1Player* player
EnemyInfo queue[MAX_ENEMIES]

bool Awake(pugi::xml_node&)
bool Start()
bool PreUpdate()
bool Update(float dt)
bool PostUpdate()
bool CleanUp()
j1Entity* CreateEntity(ENTITY_TYPES type, int x, int y)
void AddEnemy(int x, int y, ENTITY_TYPES type)
void SpawnEnemy(const EnemyInfo info)

ENTITY

SDL_Texture* graphics
ENTITY_TYPES type
fPoint position
fPoint initialPosition
fPoint speed
Animation* animation
Collider* playerHitbox
p2DynArray<iPoint> path

j1Entity* (int x, int y, ENTITY_TYPES type)
bool Awake(pugi::xml_node&)
virtual bool Start()
virtual bool PreUpdate()
virtual bool Update(float dt)
virtual bool PostUpdate()
virtual bool CleanUp()
virtual void OnCollision(Collider* c1, Collider* c2)
virtual void CheckCollision()

PLAYER

enum MOVEMENT
enum DIRECTION_X
enum DIRECTION_Y
SDL_Texture* graphics
bool GodMode
bool death
bool won
int doublejump
bool sliding
bool falling
bool onfloor

bool Awake(pugi::xml_node& config)
bool Start()
bool Update(float dt)
bool PostUpdate()
bool CleanUp()
void Respawn()
void DoAnimations()
void Move()
void OnCollision(Collider* c1, Collider* c2)
void Check_Collision()

BIRD

Animation fly_left
Animation fly_right
fPoint initialPosition
iPoint margin
iPoint colliderSize
uint c
bool move
bool die
bool path_created

bool Awake(pugi::xml_node& config)
bool Start()
bool Update(float dt)
bool CleanUp()
void OnCollision(Collider* c1, Collider* c2)
void Check_Collision()