GAME

1 →

Entity Manager

enum class entityType

p2SString player;
SDL_Texture* player_graphics;

p2SString ghost;
SDL_Texture* ghost_graphics;

p2SString slime;
SDL_Texture* slime_graphics;

p2List<j1Entity*> entities;

---

bool Awake(pugi::xml_node& config);

bool Start();

bool Update(float dt);

bool PostUpdate(float dt);

bool CleanUp();

j1Entity* CreateEntity(j1Entity::entityType type, int posx = 0, int posy = 0);

bool save(pugi::xml_node& data);

bool load(pugi::xml_node& data);

bool DeleteEntity(j1Entity* entity);

j1Entity* InfoPlayer();

Assumptions: We have an enum with all entity types. There are some functions and variables that haven't been included. This is a general representation of how our Entity System works.

*

Entity

enum entityType

entityType type;

int width;

int high;

iPoint position;

int fixBlit;

bool godMode;

bool blit;

Collider* collider;

bool to_delete;

---

j1Entity(entityType type);

bool Awake(pugi::xml_node&);

bool CleanUp();

bool save(pugi::xml_node& data);

Player

enum states

Animation* current_animation;
states actualState;
Animation idle;

p2SString jumpFx;

float gravity;
float acceleration;

float jumpF;
float energyJump;

float jumpG;
float energyGrab;
bool grabFinish;
float gGravity;
int timeGrabDelay;
int slipping;

---

bool Start();

bool Update(float dt);

bool PostUpdate(float dt);

bool CleanUp();

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

bool LoadConfigInfo();

void CheckInputState(float dt);

void CheckAnimation(float dt);

Flying Enemy

enum states_slime

states_ghost   actualState;
Animation* current_animation = nullptr;
Animation idle;

float speedx;
float speedy;

p2SString debug_tex;
SDL_Texture* graphics_debug_tex;

iPoint p;
iPoint origin;
iPoint pos;
const p2DynArray<iPoint>* lastpath;
const p2DynArray<iPoint>* path;

float wave;
bool going_up = true;
int start;

---

bool Start();

bool Update(float dt);

bool PostUpdate(float dt);

bool CleanUp();

void CheckAnimation(float dt);

bool LoadConfigInfo();

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

void Pathfinding(float dt);
void Follow_path(float dt);

Land Enemy

states_slime

float gravity;
float gGravity;

states_slime actualState;
Animation* current_animation = nullptr;
Animation idle;

p2SString debug_tex;
SDL_Texture* graphics_debug_tex;

iPoint p;
iPoint origin;
iPoint pos;
const p2DynArray<iPoint>* lastpath;
const p2DynArray<iPoint>* path;

int start;

---

bool Start();

bool Update(float dt);

bool PostUpdate(float dt);

bool CleanUp();

void CheckAnimation(float dt);

bool LoadConfigInfo();

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

void Pathfinding(float dt);
void Follow_path(float dt);