GAME

j1EntityManager

j1Entity* EntityFactory (ENTITY_TYPE type, float x = 0, float y = 0);
void CreateEntity (ENTITY_TYPE type, float x = 0, float y = 0);
void AddEnemy (float x, float y, ENTITY_TYPE type);
void SpawnEnemy (const EntityInfo& info);

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

---

p2List<j1Entity*> entityList;

j1Player* player;
j1Chest* chest;
j1MovingObstacle* obstacle;
j1Door* door;
j1Bat* bat;
j1Demon* demon;
j1BatEnemy* bat_e;
j1LifeItem* lifeItem;
j1JetpackItem* jetpackItem;

EntityInfo queue[MAX_ENTITIES];

j1Entity

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

virtual void BlitEntity (SDL_Rect r, bool flip = false, float x = 0, float y = 0);

ENTITY_TYPE type;

fPoint position;

Collider* collider = nullptr;

Animation* animation = nullptr;

SDL_Texture* sprites = nullptr;

ENTITY_TYPE

PLAYER,
CHEST,
OBSTACLE,
DOOR,
BAT,
DEMON,
BAT_E,
LIFE_ITEM,
JETPACK_ITEM,
DAMAGE_ITEM,
POINTS_ITEM

ENTITY_INFO

ENTITY_TYPE type;

fPoint position;

j1Player

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

void PlayerMovement (float dt);
void Jumping();
void Jetpack();
void Die();
void Shooting(float x, float y, float dt);
void ChangeWeapon();
void GodMode(float dt);

void LoadInfo();

Animation idle;
Animation run;
Animation jetpack;
Animation jump;
Animation godAnim;
Animation deathAnim;

PARTICLE_TYPE currentType = BASIC_SHOOT;

fPoint position, speed;
fPoint margin_particles, speed_particles;

iPoint hitbox;

float jumpForce, jetForce, godSpeed;
float gravity, jumpForce_xml, jetForce_xml;

int lifes = 3, double_jump = 2;

bool playerCanMove = false;
bool openingChest = false;
bool itemPicked = false;
bool doorOpened = false;
bool godMode = false;
bool isDead = false;
bool playerCreated = false;
bool onFloor = false;
bool isJumping = false;
bool jetpackActive = false;
bool changedFloor = false;
bool touchingWin = false;
bool ColRight = false;
bool ColLeft = false;
bool ColUp = false;
bool ColDown = false;
bool flip = true;
bool playedFx = false;
bool playedFx2 = false;

j1Bat

void MoveHorizontal (float x);
void MoveVertical (float y);

Animation idle;

fPoint bat_position;

float inertia = 0;
float vertical_speed = 0;

bool redirect_horizontal = false;
bool moving_vertical = false;
bool movement = false;

j1MovingObstacle

void Properties();

Animation idle;

fPoint speed, limit;
fPoint movement;

int topHeight, lowHeight;
int variation, adjust;
int hitbox;

bool movingDown = false;
bool movingUp = false;
bool active = false;
bool playedFx = false;

j1Demon

void Move (p2DynArray<iPoint>& path, float dt);

void PathFind (float dt);
void Shoot (float dt);
void MoveBack (float dt);
void Jump (float dt);

Animation idleAnim;
Animation runAnim;
Animation attackAnim;
Animation deathAnim;
Animation hurtAnim;

PATH_MOVEMENT direction;

j1Timer timerShot;

p2DynArray<iPoint>* path;

fPoint demon_position, speed;

float jump_force, gravity;

uint lastShot = 0, cooldownShot;

bool ColLeft = false;
bool ColRight = false;
bool ColDown = false;
bool ColUp = false;
bool dead = false;

bool jumping = false;
bool move_back = false;bool flip = false;
bool path_created = false;

j1BatEnemy

void Move(p2DynArray<iPoint>& path, float dt);

Animation idleAnim;
Animation pathAnim;
Animation dyingAnim;
Animation deathAnim;

PATH_MOVEMENT direction;

p2DynArray<iPoint>* path;

fPoint speed;

float gravity;

bool ColLeft = false;
bool ColRight = false;
bool ColDown = false;
bool ColUp = false;
bool fall = false;

bool dead = false;
bool flip = false;
bool path_created = false;

j1Door

Animation idle;
Animation openingAnim;

fPoint door_position;

bool opened = false;
bool playedFx = false;

j1Item

Animation idleAnim;
Animation destroyAnim;

j1Platform

void Move(float dt);
void MoveVertical(float dt);
void Properties();

Animation idle;

fPoint speed;
float initial_position;
float limit_left, right, up, down;

iPoint limits;
iPoint hitbox;
int type_platform;

bool isVertical = false;
bool arrived_X, arrived_Y;