Association — e.g : a use b: a — use b b wstring ToWSTR(string st) Aggregation ——— **CGameObject** # vx, vy: float # nx: int # state: int # isDeleted: bool +SetPosition(float x, float y): void +SetSpeed(float vx, float vy): void +GetPosition(float &x, float &y): void +GetSpeed(float &vx, float &vy): void +GetState(): int +Delete(): virtual void +IsDeleted(): bool +RenderBoundingBox(): void CGameObject() CGameObject(float x, float y) +GetBoundingBox(float &I;, float &t, float &r, float &b): virtual void = 0 +Update(DWORD dt, vector<LPGAMEOBJECT>\* coObjects = NULL): virtual void +Render(): virtual void = 0 +SetState(int state): virtual void +lsCollidable(): virtual int +OnNoCollision(DWORD dt): virtual void +OnCollisionWith(LPCOLLISIONEVENT e): virtual void +IsBlocking(): virtual int ~CGameObject() +IsDeleted(const LPGAMEOBJECT &o): static bool **CPlatform** CGoomba CCoin **CPortal** CInvisiblePlatform # ax, ay: float -isSitting: BOOLEAN -width, height: int -scene\_id: int // target scene to switch to # cellWidth, cellHeight: float # die\_start: ULONGLONG -maxVx: float -width, height: float -ax, ay: float // accelaration on x, y # spriteIdBegin, spriteIdMiddle, spriteIdEnd: int +CInvisiblePlatform(int x, int y, int width, int + CBrick(float x, float y): CGameObject(x, y) + Ccoin(float x, float y): CGameObject(x, y) # GetBoundingBox(float &left, float &top, float &right, float height):CGameObject(x, y) + Render(): void +CPortal(float I, float t, float r, float b, int + Render(): void -level: int &bottom): virtual void + CPlatform(float x, float y, float cell\_width, float cell\_height, int + Update(DWORD dt): void + Update(DWORD dt): void scene\_id) -untouchable: int +Render(): void length, int sprite\_id\_begin, int sprite\_id\_middle, int sprite\_id\_end) + GetBoundingBox(float& I, float& t, float& r, float& b): + GetBoundingBox(float& I, float& t, float& r, float& b): +Render(): virtual void -untouchable\_start: ULONGLONG # Update(DWORD dt, vector<LPGAMEOBJECT> +Update(WORD dt): void +GetBoundingBox(float &I, float &t, float &r, :CGameObject(x, y) -isOnPlatform: BOOLEAN +GetBoundingBox(float& I, float& t, float& r, float& + IsBlocking(): int \*coObjects): virtual void float &b): virtual void -coin: int + Render(): void +GetSceneId(): int # Render(): virtual void +RenderBoundingBox(): void + Update(DWORD dt): void +IsBlocking(): ing + GetBoundingBox(float& I, float& t, float& r, float& b): void -OnCollisionWithGoomba(LPCOLLISIONEVENT e): void # IcCollidable(): virtual int + RenderBoundingBox(): void -OnCollisionWithCoin(LPCOLLISIONEVENT e): void # IsBlocking():: virtual int -OnCollisionWithPortal(LPCOLLISIONEVENT e): void # OnNoCollision(DWORD dt): virtual void -GetAniIdBig(): int # OnCollisionWith(LSCOLLISIONEVENT e) -GetAniIdSmall(): int +CGoomba(float x, float y) +CMario(float x, float y) : CGameObject(x, y) +SetState(int state): virtual void +Update(DWORD dt, vector<LPGAMEOBJECT>\* coObjects): void() +Render(): void +SetState(int state): void +IsCollidable(): int +IsBlocking(): int +OnNoCollision(DWORD dt): void +OnCollisionWith(LPCOLLISIONEVENT e): void +SetLevel(int I): void +StartUntouchable(): void +GetBoundingBox(float& left, float& top, float& right, float& bottom): void CTextures -\_\_instance: static CTextures textures: unordered\_map<int, LPTEXTURE> +CTextures() +Add(int id, LPCWSTR filePath) +Get(unsigned int i): LPTEXTURE +GetInstance(): static CTextures\*

Inheritance \_\_\_\_\_>

