Aggregation ———

CGameObject # vx, vy: float # nx: int # state: int # isDeleted: bool # type: int +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(float x, float y, int type) +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 +IsCollidable(): virtual int +OnNoCollision(DWORD dt): virtual void +OnCollisionWith(LPCOLLISIONEVENT e): virtual void +IsBlocking(): virtual int ~CGameObject() +IsDeleted(const LPGAMEOBJECT &o): static bool CPortal **CPlatform** CMushroom CInvisiblePlatform # ax, ay: float # ax, ay: float -width, height: int -isSitting: BOOLEAN -scene_id: int // target scene to switch to # cellWidth, cellHeight: float # die_start: ULONGLONG -maxVx: float -width, height: float # spriteIdBegin, spriteIdMiddle, spriteIdEnd: int -ax, ay: float // accelaration on x, y +CInvisiblePlatform(float x, float y, int type, int + CBrick(float x, float y, int type): CGameObject(x, y, + Ccoin(float x, float y, int type): CGameObject(x, y, # GetBoundingBox(float &left, float &top, float &right, float # GetBoundingBox(float &left, float &top, float &right, float width, int height):CGameObject(x, y, type) +CPortal(float I, float t, float r, float b, int -level: int &bottom): virtual void &bottom): virtual void - CPlatform(float x, float y, int type, float cell_width, float cell_height, + Render(): void + Render(): void scene_id, int type) -untouchable: int +Render(): void int length, int sprite_id_begin, int sprite_id_middle, int sprite_id_end) + Update(DWORD dt): void + Update(DWORD dt): void +Render(): virtual void -untouchable_start: ULONGLONG # Update(DWORD dt, vector<LPGAMEOBJECT> # Update(DWORD dt, vector<LPGAMEOBJECT> +Update(WORD dt): void + GetBoundingBox(float& I, float& t, float& r, float& b): + GetBoundingBox(float& I, float& t, float& r, float& b): +GetBoundingBox(float &I, float &t, float &r, :CGameObject(x, y, type) -isOnPlatform: BOOLEAN +GetBoundingBox(float& I, float& t, float& r, float& *coObjects): virtual void *coObjects): virtual void float &b): virtual void -coin: int + IsBlocking(): int - Render(): void + SetState(int state): void +GetSceneId(): int # Render(): virtual void # Render(): virtual void +RenderBoundingBox(): void - Update(DWORD dt): void +IsBlocking(): ing -isStopUpdate: bool GetBoundingBox(float& I, float& t, float& r, float& b): void # IsCollidable(): virtual int + RenderBoundingBox(): void # IsCollidable(): virtual int -OnCollisionWithGoomba(LPCOLLISIONEVENT e): void # IsBlocking():: virtual int # IsBlocking():: virtual int # OnNoCollision(DWORD dt): virtual void # OnNoCollision(DWORD dt): virtual void -OnCollisionWithCoin(LPCOLLISIONEVENT e): void -OnCollisionWithPortal(LPCOLLISIONEVENT e): void # OnCollisionWith(LSCOLLISIONEVENT e) # OnCollisionWith(LSCOLLISIONEVENT e) | -GetAniIdBig(): int +CGoomba(float x, float y, int type) -GetAniIdSmall(): int +CMushroom(float x, float y, int type) +SetState(int state): virtual void +SetState(int state): virtual void +CMario(float x, float y, int type) : CGameObject(x, y, CQuestionBrick +Update(DWORD dt, vector<LPGAMEOBJECT>* coObjects): void() -default_pos_y: float +Render(): void +SetState(int state): void -itemInsideId: int +IsCollidable(): int + CQuestionBrick(float x, float y, int type, int item_id): CBrick(x, y, +IsBlocking(): int + Render(): void +OnNoCollision(DWORD dt): void + Update(DWORD dt, vectoc<LPGAMEOBJECT>* coObjects): void +OnCollisionWith(LPCOLLISIONEVENT e): void + GetBoundingBox(float& I, float& t, float& r, float& b): void CWingGoomba + SetState(int state): void +SetLevel(int I): void -walkingDistance: float +StartUntouchable(): void -startWalkingLocation: float +GetBoundingBox(float& left, float& top, float& right, float& bottom): void # GetBoundingBox(float &left, float &top, float &right, float &bottom): virtual void +IsStopUpdate(): bool # Update(DWORD dt, vector<LPGAMEOBJECT> *coObjects): virtual void # Render(): virtual void # IsCollidable(): virtual int # IsBlocking():: virtual int # OnNoCollision(DWORD dt): virtual void # OnCollisionWith(LSCOLLISIONEVENT e) +CWingGoomba(float x, float y, int type) +SetState(int state): virtual void

