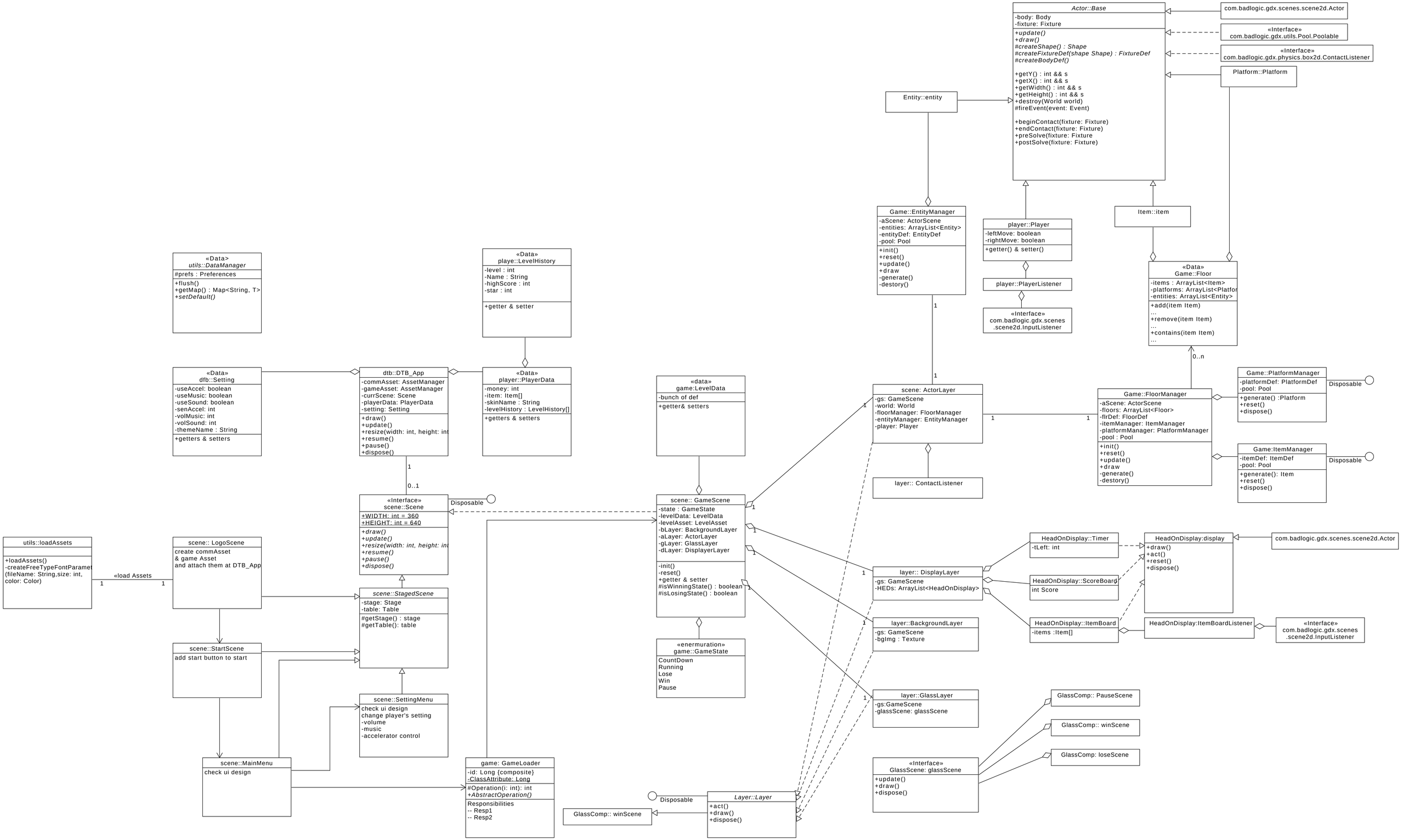
«Data»
utils::DataManager
#prefs : Preferences
+flush()
+getMap() : Map<String, T>
+setDefault()
«Data»
playe::LevelHistory
-level : int
-Name : String
-highScore : int
-star : int
+getter & setter
«Data»
dfb::Setting
-useAccel: boolean
-useMusic: boolean
-useSound: boolean
-senAccel: int
-volMusic: int
-volSound: int
-themeName : String
+getters & setters
dtb::DTB_App
-commAsset: AssetManager
-gameAsset: AssetManager
-currScene: Scene
-playerData: PlayerData
-setting: Setting
+draw()
+update()
+resize(width: int, height: int
+resume()
+pause()
+dispose()
«Data»
player::PlayerData
-money: int
-item: Item[]
-skinName : String
-levelHistory : LevelHistory[]
+getters & setters
«data»
game:LevelData
-bunch of def
+getter& setters
1
0..1
«Interface»
scene::Scene
+WIDTH: int = 360
+HEIGHT: int = 640
+draw()
+update()
+resize(width: int, height: int
+resume()
+pause()
+dispose()
Disposable
scene:: GameScene
-state : GameState
-levelData: LevelData
-levelAsset: LevelAsset
-bLayer: BackgroundLayer
-aLayer: ActorLayer
-gLayer: GlassLayer
-dLayer: DisplayerLayer
-init()
-reset()
+getter & setter
#isWinningState() : boolean
#isLosingState() : boolean
utils::loadAssets
+loadAssets()
-createFreeTypeFontParamet
(fileName: String,size: int,
color: Color)
«load Assets
scene:: LogoScene
create commAsset
& game Asset
and attach them at DTB_App
scene::StagedScene
-stage: Stage
-table: Table
#getStage() : stage
#getTable(): table
scene::StartScene
add start button to start
scene::SettingMenu
check ui design
change player's setting
-volume
-music
-accelerator control
«enermuration»
game::GameState
CountDown
Running
Lose
Win
Pause
scene::MainMenu
check ui design
game: GameLoader
-id: Long {composite}
-ClassAttribute: Long
#Operation(i: int): int
+AbstractOperation()
Responsibilities
-- Resp1
-- Resp2
Disposable
GlassComp:: winScene
Layer::Layer
+act()
+draw()
+dispose()
Actor::Base
-body: Body
-fixture: Fixture
+update()
+draw()
#createShape() : Shape
#createFixtureDef(shape Shape) : FixtureDef
#createBodyDef()
+getY() : int && s
+getX() : int && s
+getWidth() : int && s
+getHeight() : int && s
+destroy(World world)
#fireEvent(event: Event)
+beginContact(fixture: Fixture)
+endContact(fixture: Fixture)
+preSolve(fixture: Fixture
+postSolve(fixture: Fixture)
com.badlogic.gdx.scenes.scene2d.Actor
«Interface»
com.badlogic.gdx.utils.Pool.Poolable
«Interface»
com.badlogic.gdx.physics.box2d.ContactListener
Platform::Platform
Entity::entity
Game::EntityManager
-aScene: ActorScene
-entities: ArrayList<Entity>
-entityDef: EntityDef
-pool: Pool
+init()
+reset()
+update()
+draw
-generate()
-destory()
player::Player
-leftMove: boolean
-rightMove: boolean
+getter() & setter()
player::PlayerListener
«Interface»
com.badlogic.gdx.scenes
.scene2d.InputListener
Item::item
«Data»
Game::Floor
-items : ArrayList<Item>
-platforms: ArrayList<Platfor
-entities: ArrayList<Entity>
+add(item Item)
...
+remove(item Item)
...
+contains(item Item)
...
0..n
scene: ActorLayer
-gs: GameScene
-world: World
-floorManager: FloorManager
-entityManager: EntityManager
-player: Player
Game::FloorManager
-aScene: ActorScene
-floors: ArrayList<Floor>
-flrDef: FloorDef
-itemManager: ItemManager
-platformManager: PlatformManager
-pool : Pool
+init()
+reset()
+update()
+draw
-generate()
-destory()
Game::PlatformManager
-platformDef: PlatformDef
-pool: Pool
+generate() :Platform
+reset()
+dispose()
Disposable
Game:ItemManager
-itemDef: ItemDef
-pool: Pool
+generate(): Item
+reset()
+dispose()
Disposable
layer:: ContactListener
layer:: DisplayLayer
-gs: GameScene
-HEDs: ArrayList<HeadOnDisplay>
HeadOnDisplay::Timer
-tLeft: int
HeadOnDisplay::ScoreBoard
int Score
HeadOnDisplay::ItemBoard
-items :Item[]
HeadOnDisplay:display
+draw()
+act()
+reset()
+dispose()
com.badlogic.gdx.scenes.scene2d.Actor
HeadOnDisplay:ItemBoardListener
«Interface»
com.badlogic.gdx.scenes
.scene2d.InputListener
layer::BackgroundLayer
-gs: GameScene
-bgImg : Texture
layer::GlassLayer
-gs:GameScene
-glassScene: glassScene
«Interface»
GlassScene: glassScene
+update()
+draw()
+dispose()
GlassComp:: PauseScene
GlassComp:: winScene
GlassComp: loseScene