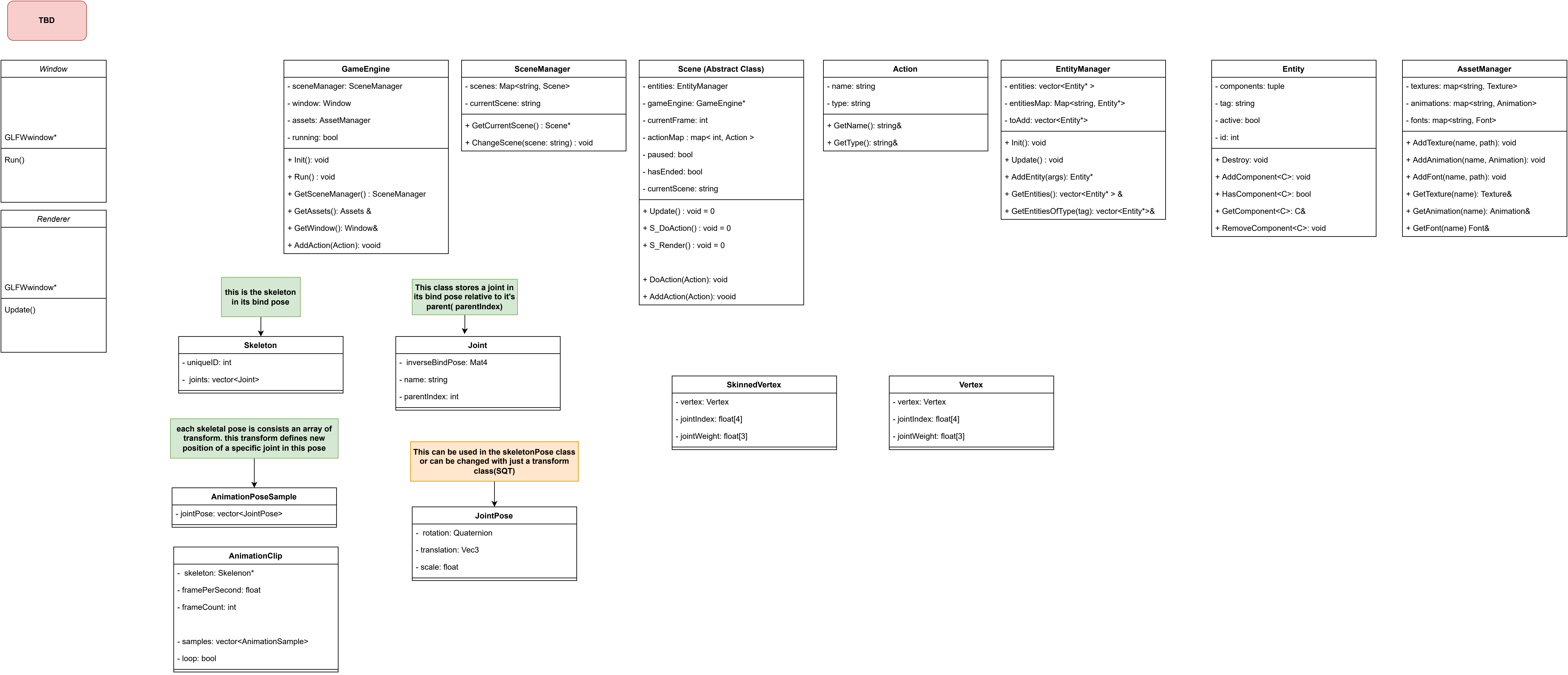
TBD
Window
GLFWwindow*
Run()
Renderer
GLFWwindow*
Update()
GameEngine
- sceneManager: SceneManager
- window: Window
- assets: AssetManager
- running: bool
+ Init(): void
+ Run() : void
+ GetSceneManager() : SceneManager
+ GetAssets(): Assets &
+ GetWindow(): Window&
+ AddAction(Action): vooid
SceneManager
- scenes: Map<string, Scene>
- currentScene: string
+ GetCurrentScene() : Scene*
+ ChangeScene(scene: string) : void
Scene (Abstract Class)
- entities: EntityManager
- gameEngine: GameEngine*
- currentFrame: int
- actionMap : map< int, Action >
- paused: bool
- hasEnded: bool
- currentScene: string
+ Update() : void = 0
+ S_DoAction() : void = 0
+ S_Render() : void = 0
+ DoAction(Action): void
+ AddAction(Action): vooid
Action
- name: string
- type: string
+ GetName(): string&
+ GetType(): string&
EntityManager
- entities: vector<Entity* >
- entitiesMap: Map<string, Entity*>
- toAdd: vector<Entity*>
+ Init(): void
+ Update() : void
+ AddEntity(args): Entity*
+ GetEntities(): vector<Entity* > &
+ GetEntitiesOfType(tag): vector<Entity*>&
Entity
- components: tuple
- tag: string
- active: bool
- id: int
+ Destroy: void
+ AddComponent<C>: void
+ HasComponent<C>: bool
+ GetComponent<C>: C&
+ RemoveComponent<C>: void
AssetManager
- textures: map<string, Texture>
- animations: map<string, Animation>
- fonts: map<string, Font>
+ AddTexture(name, path): void
+ AddAnimation(name, Animation): void
+ AddFont(name, path): void
+ GetTexture(name): Texture&
+ GetAnimation(name): Animation&
+ GetFont(name) Font&
this is the skeleton in its bind pose
Skeleton
- uniqueID: int
- joints: vector<Joint>
This class stores a joint in its bind pose relative to it's parent(parentIndex)
Joint
- inverseBindPose: Mat4
- name: string
- parentIndex: int
SkinnedVertex
- vertex: Vertex
- jointIndex: float[4]
- jointWeight: float[3]
Vertex
- vertex: Vertex
- jointIndex: float[4]
- jointWeight: float[3]
each skeletal pose is consists an array of transform. this transform defines new position of a specific joint in this pose
AnimationPoseSample
- jointPose: vector<JointPose>
This can be used in the skeletonPose class or can be changed with just a transform class(SQT)
JointPose
- rotation: Quaternion
- translation: Vec3
- scale: float
AnimationClip
- skeleton: Skelenon*
- framePerSecond: float
- frameCount: int
- samples: vector<AnimationSample>
- loop: bool