

 fieldNode: SKFieldNode fieldShape: SKShapeNode + showField: Bool + radius: CGFloat + fallof: CGFloat + categoryBitMask: UInt32 + region: SKRegion? + strength: CGFloat

+ fieldColor: UIColor

+ didAddToEntity() -> Void

+ init(radius: CGFloat, showField: Bool)

- fieldNode: SKFieldNode fieldShape: SKShapeNode - frictionFunction: SKFieldForceEvaluator + showField: Bool - angle: CGFloat + radius: CGFloat radius: CGFloat + fallof: CGFloat - phase: CGFloat + categoryBitMask: UInt32 + region: SKRegion? + init(shapeType: MGFrictionFieldShapeType: CGFloat, showField: Bool) + didAddToEntity() -> Void

- ORBIT_LINE_NAME: String angularVelocity: CGFloat - orbiting Entity: GKEntity? orbitPoint: CGPoint timeElapsed: TimeInterval - orbitLine: SKShapeNode? + showingOrbitLine: Bool + didAddToEntity() -> Void

+ update(deltaTime seconds: TimeInterval) -> Void

+ releaseFromOrbit(makeDynamic: Bool)

+ soundNode: SKAudioNode + generator: UllmpactFeedbackGenerator + impact: UllmpactFeedbackGenerator.FeedbackStyle

+ audioData: MGAudioUnit + init(sound: MGAudioUnit, loopAudio: Bool) + init(impact: UIImpactFeedbackGenerator.FeedbackStyle) + didAddToEntity() -> Void + didAddToEntity() -> Void + run(impactType: UIImpactFeedbackGenerator.FeedBackStyle?) + setVolume(value: CGFloat)

- handleOrbit(point: CGPoint, overrideRadius: CGFloat?, overrideSpeed: CGFloat?) + fileName: String + orbitAround(point: CGPoint, overrideRadius: CGFloat?, overrideSpeed: CGFloat?) + type: AudioType + orbitAround(entity: GKEntity, overrideRadius: CGFloat?, overrideSpeed: CGFloat?)

+ data: Dictionary<String, AnyObject>

+ init(data: Dictionary<String, AnyObject>?) + addAnimation(actions: [SKAction], withName: String) -> Void + runAnimation(withName: String,

- commandBank: Dictionary<String, [SKAction]>

orderOfSecond: ActionAnimationOrderType,

nameOfResult: String) -> Bool

orderType: ActionAnimationOrderType, target: SKNode?, onCompletion: @escaping () -> Void) -> Void + getAnimationNames() -> [String] + pairAnimations(nameOfFirst: String, nameOfSecond: String, orderOfFirst: ActionAnimationOrderType,

 tileMap: SKTileMapNode + emitterNode: SKEmitterNode + init(tileSetName: String, size: CGSize) + init(emitterName: String) + didAddToEntity() -> Void + init(tileSetName: String, columns: Int, rows: Int) + addAnimation(action: SKAction, withName: String) -> Void + emit(at relPoint: CGPoint, emitTime: TimeInterval) -> Void + didAddToEntity() -> Void

+ fill(withTileGroupName: String) -> Void - anchorPoint: CGPoint + isActive: Bool

+ init(strength: CGFloat, camera: MGCameraNode) + didAddToEntity() -> Void

+ camera: MGCameraNode?

+ update(deltaTime seconds: TimeInterval) -> Void

+ strength: CGFloat