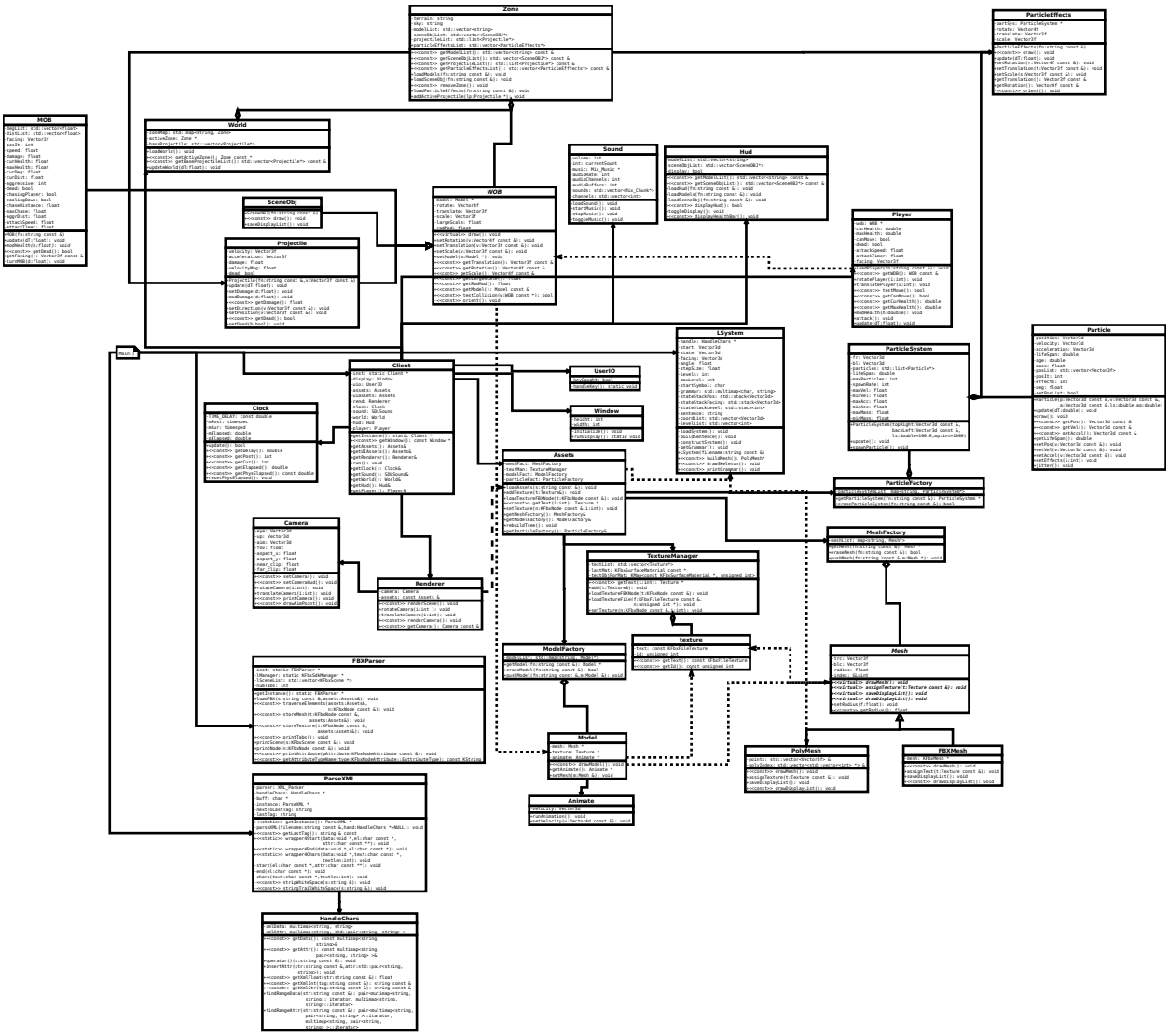
Zone
-terrain: string
-sky: string
-modelList: std::vector<string>
-sceneObjList: std::vector<SceneOBJ*>
-projectileList: std::vector<Projectile*>
-particleEffectsList: std::vector<ParticleEffects*>
+<<const>> getModelList(): std::vector<string> const &
+<<const>> getSceneOBJList(): std::vector<SceneOBJ*> const &
+<<const>> getProjectileList(): std::list<Projectile*> const &
+<<const>> getParticleEffectsList(): std::vector<ParticleEffects*> const &
+loadModel(fn:string const &): void
+loadSceneOBJ(fn:string const &): void
+<<const>> removeDone(): void
+loadParticleEffects(fn:string const &): void
+addActiveProjectile(p:Projectile *): void
ParticleEffects
-partSys: ParticleSystem *
-translate: Vector3f
-rotate: Vector3f
-scale: Vector3f
+ParticleEffects(fn:string const &)
+<<const>> draw(): void
+update(df:float): void
+setRotation(r:Vector4f const &): void
+setTranslate(t:Vector3f const &): void
+setScale(s:Vector3f const &): void
+getRotation(): Vector4f const &
+getTranslate(): Vector3f const &
+<<const>> orient(): void
MOB
-dmgList: std::vector<float>
-distList: std::vector<float>
-facing: Vector3f
-pos: Vector3f
-speed: float
-damage: float
-curHealth: float
-maxHealth: float
-curDist: float
-curDist: float
-aggressive: int
-dead: bool
-chasingPlayer: bool
-coolingDown: bool
-chaseDistance: float
-maxDist: float
-wapDist: float
-attackSpeed: float
-attackTimer: float
+MOB(fn:string const &)
+update(df:float): void
+modHealth(h:float): void
+<<const>> getDead(): bool
+<<const>> getPos(): Vector3f const &
+getMaxDist(): float
World
-loadMap: std::map<string, Zone *>
-activeZone: Zone *
-baseProjectile: std::vector<Projectile*>
+World(): void
+<<const>> getActiveZone(): Zone const *
+<<const>> getBaseProjectileList(): std::vector<Projectile*> const &
+updateWorld(df:float): void
Sound
-volume: int
-sfx: currentSound
-music: Mix_Music *
-audioRate: int
-audioChannels: int
-audioBuffers: int
-audioFormat: enum<Mix_Chunk*>
-channels: std::vector<int>
+loadSound(): void
+startMusic(): void
+stopMusic(): void
+togleMusic(): void
Hud
-modelList: std::vector<string>
-sceneObjList: std::vector<SceneOBJ*>
-Player *
+<<const>> getModelList(): std::vector<string> const &
+<<const>> getSceneOBJList(): std::vector<SceneOBJ*> const &
+loadModel(fn:string const &): void
+loadModel(fn:string const &): void
+loadSceneOBJ(fn:string const &): void
+<<const>> displayMap(): bool
+togglemap(p): void
+<<const>> displayHealthBar(): void
SceneObj
-pos: Vector3f
+SceneObj(fn:string const &)
+<<const>> draw(): void
+displayList(): void
Projectile
-velocity: Vector3f
-acceleration: Vector3f
-lifetime: int
-velocityMag: float
-dead: bool
+Projectile(fn:string const &,v:Vector3f const &)
+update(df:float): void
+setDamage(d:float): void
+modDamage(d:float): void
+setSpeed(s:float): void
+setDirection(v:Vector3f const &): void
+setPosition(v:Vector3f const &): void
+<<const>> getDead(): bool
+setDead(b:bool): void
WOB
-model: Model *
-rotate: Vector4f
-translate: Vector3f
-scale: Vector3f
-largeScale: float
-radius: float
+<<const>>draw(): void
+setRotation(v:Vector4f const &): void
+setTranslation(v:Vector3f const &): void
+setScale(v:Vector3f const &): void
+setModel(m:Model *): void
+<<const>> getTranslation(): Vector3f const &
+<<const>> getRotation(): Vector4f const &
+<<const>> getScale(): Vector4f const &
+<<const>> getLargeScale(): float
+<<const>> getRadius(): float
+<<const>> getModel(): Model const &
+<<const>> testWithLook(w: WOB const *): bool
+<<const>> print(): void
Player
-wob: WOB *
-curHealth: double
-maxHealth: double
-canMove: bool
-dead: bool
-attackRadius: float
-attackTimer: float
-facing: Vector3f
+loadPlayer(fn:string const &): void
+<<const>> getWOB(): WOB const &
+rotatePlayer(i:int): void
+translatePlayer(i:int): void
+<<const>> testMove(): bool
+<<const>> getCanMove(): bool
+<<const>> getMaxHealth(): double
+<<const>> getMaxHealth(): double
+modHealth(h:double): void
+attack(): void
+updateTimer(df:float): void
Particle
-position: Vector3d
-velocity: Vector3d
-acceleration: Vector3d
-lifetime: double
-age: double
-mass: float
-gaussist: std::vector<Vector3f>
-posIt: int
-effects: int
-deg: float
-dead: bool
+Particle(p:Vector3d const &,v:Vector3d const &,a:Vector3d const &,l:double,ag:double)
+update(df:double): void
+draw(): void
+<<const>> getPos(): Vector3d const &
+<<const>> getVel(): Vector3d const &
+<<const>> getAccel(): Vector3d const &
+getLifeSpan(): double
+setPos(v:Vector3d const &): void
+setVel(v:Vector3d const &): void
+setAccel(v:Vector3d const &): void
+setEffects(i:int): void
+jitter(): void
Client
-sock: TcpClientSocket *
-display: Window
-uio: UserIO
-assets: Assets
-classes: Assets
-rend: Renderer
-hud: Hud
-sound: SDLSound
-world: World
-hud: Hud
-player: Player
+Client()
+getInstance(): static client *
+<<const>> getWindow(): const Window *
+getUIO(): UserIO &
+getAllAssets(): Assets &
+getRenderer(): Renderer&
+run(): void
+getClock(): Clock&
+getSound(): SDLSound&
+getWorld(): World&
+getHud(): Hud&
+getPlayer(): Player&
UserIO
-keyStatus: bool
+keyCheck(): static void
Window
-height: int
-width: int
+initWindow(): void
+setDimension(): static void
Clock
-FRAME_DELAY: const double
-mTest: timespec
-mCur: timespec
-mElapsed: double
-mPrevTime: double
+update(): bool
+<<const>> getDelay(): double
+<<const>> getPoint(): int
+<<const>> getCur(): int
+<<const>> getElapsed(): double
+<<const>> getPrevElapsed(): const double
+setPrevElapsed(): void
LSystem
-handle: HandleChars *
-start: Vector3d
-state: Vector3d
-facing: Vector3d
-angle: float
-stepSize: float
-levels: int
-iterations: int
-startSymbol: char
-grammar: std::multimap<char, string>
-stateStackPos: std::stack<Vector3d>
-stateStackFacing: std::stack<Vector3d>
-stateStackLevel: std::stack<int>
-sentence: string
-lineList: std::vector<Vector3d>
-lineFlist: std::vector<int>
+LSystem():: void
+buildSentence(): void
+constructSystem(): void
+getGrammar(): void
+LSystem(fl:lameme:string const &)
+<<const>> drawMesh(m:PolyMesh*)
+<<const>> drawDeleteLin(): void
+<<const>> printGrammar(): void
ParticleSystem
-fr: Vector3d
-bl: Vector3d
-liveParticles: std::list<Particle*>
-lifeSpan: double
-maxParticles: int
-timeRate: int
-maxVel: float
-minVel: float
-maxAccl: float
-minAccl: float
-maxMass: float
-minMass: float
+ParticleSystem(maxRight:Vector3d const &, backLeft:Vector3d const &, l:double=100.0,mp:int=3000)
+update(): void
+timePartic(df): void
ParticleFactory
-particleSystemMap: std::map<string, ParticleSystem*>
+getParticleSystem(fn:string const &):ParticleSystem *
+createParticleSystem(fn:string const &): bool
Assets
-meshFact: MeshFactory
-texMan: TextureManager
-modelFact: ModelFactory
-particleFact: ParticleFactory
+loadAssets(fn:string const &): void
+addTexture(t:Texture&): void
+<<const>> getTexture(s:string const &): void
+<<const>> getTex(i:int): Texture *
+setTexture(m:KFbxNode const &,i:int): void
+getMeshFactory(): MeshFactory&
+getModelFactory(): ModelFactory&
+modUsfree(): void
+getParticleFactory(): ParticleFactory&
Camera
-pos: Vector3d
-up: Vector3d
-aim: Vector3d
-fov: float
-aspect_x: float
-aspect_y: float
-near_clip: float
-far_clip: float
+<<const>> setCamera(): void
+<<const>> setCameraAbs(): void
+rotateCamera(i:int): void
+translateCamera(i:int): void
+<<const>> printCamera(): void
+<<const>> drawLinPoint(): void
Renderer
-camera: Camera
-assets: const Assets &
-drawLinPoint: Vector3d
+rotateCamera(i:int): void
+translateCamera(i:int): void
+<<const>> renderCamera(): void
+<<const>> getCamera(): Camera const &
TextureManager
-textList: std::vector<Texture*>
-textMap: KFbxSurfaceMaterial const *
+loadTexture(m:KFbxSurfaceMaterial *, unsigned int)
+<<const>> getTex(i:int): Texture *
+add(t:Texture&): void
+loadTexture(KFbxNode(t:KFbxNode const &): void
+loadTextureFile(f:KFbxFileTexture const &, n:unsigned int *): void
+getTexture(n:KFbxNode const &,i:int): void
MeshFactory
-meshList: std::map<string, Mesh*>
+getMesh(fn:string const &): Mesh *
+eraseMesh(fn:string const &): bool
+pushMesh(fn:string const &,m:Mesh *): void
texture
-text: const KFbxFileTexture *
-textID: int
+<<const>> getText(): const KFbxFileTexture *
+<<const>> getID(): const unsigned int
ModelFactory
-modelList: std::map<string, Model*>
+getModel(fn:string const &): Model *
+drawModel(fn:string const &): bool
+pushModel(fn:string const &,m:Model &): void
Mesh
-bln: Vector3f
-blr: Vector3f
-radius: float
-center: Vector3f
+Mesh(): void
+virtual void drawMesh(): void
+virtual void drawMesh(v:Texture const &): void
+virtual void saveDisplayList(): void
+virtual void drawDisplayList(): void
+setRadius(f:float): void
+<<const>> getRadius(): float
FBXParser
-lsdk: static FbxParser *
-lManager: static KFbxSdkManager *
-lScene: static std::vector<KFbxScene *>
-lnumTabs: int
+getInstance(): static FbxParser *
+loadFBX(s:string const &,assets:Assets&): void
+<<const>> traverseElements(assets:Assets&, n:KFbxNode const &): void
+<<const>> storeMesh(t:KFbxNode const &, assets:Assets&): void
+<<const>> storeTexture(t:KFbxNode const &, assets:Assets&): void
+<<const>> printTabs(): void
+printScene(s:KFbxScene const &): void
+printNode(n:KFbxNode const &): void
+<<const>> printAttribute(pAttribute:KFbxNodeAttribute const &): void
+<<const>> getAttrStr(pAttribute:KFbxNodeAttribute.EAttrType): const KString *
Model
-mesh: Mesh *
-texture: Texture *
-animate: Animate *
+<<const>> drawMesh(): void
+getAnimate(): Animate *
+setMesh(m:Mesh &): void
PolyMesh
-points: std::vector<Vector3f> *
-polyIndex: std::vectorvector<int>vector<int> *
+<<const>> drawMesh(): void
+assignTexture(t:Texture const &): void
+saveDisplayList(): void
+<<const>> drawDisplayList(): void
FBXMesh
-mesh: KFbxMesh *
+<<const>> drawMesh(): void
+assignText(t:Texture const &): void
+saveDisplayList(): void
+<<const>> drawDisplayList(): void
ParseXML
-parser: XML_Parser
-handleChars: HandleChars *
-depth: int
-instance: ParseXML *
-mostToLastTag: string
-lastTag: string
+<<static>> getInstance(): ParseXML *
+parseXML(fl:lameme:string const &,hand:HandleChars *=NULL): void
+<<const>> getLastTag(): string & const
+<<static>> wrapperStart(data:void *,el:char const *, attr:char const **): void
+<<static>> wrapperEnd(data:void *,el:char const *): void
+<<static>> wrapperChars(data:void *,text:char const *, textlen:int): void
+start(el:char const *,attr:char const **): void
+end(el:char const *): void
+chars(text:char const *,textlen:int): void
+<<const>> stringMerge(s:string &): void
+<<const>> stringTrim(s:string &): void
HandleChars
-pcdata: multimap<string, string>
-pcdatr: multimap<string, std::pair<string, string>>
+<<const>> getData(s): const multimap<string, string>&
+<<const>> getAttr(): const multimap<string, std::pair<string, string>>&
+operator()(s:string const &): void
+insertAttr(str:string const &,attr:std::pair<string, string>): void
+<<const>> getNdFloat(str:string const &): float
+<<const>> getNdInt(tag:string const &): string const &
+<<const>> getNdStr(tag:string const &): string const &
+findRangeData(str:string const &,tag:multimap<string, string>, iterator, multimap<string, string>::iterator)
+findRangeAttr(str:string const &): pair<multimap<string, pair<string, string> >::iterator, multimap<string, pair<string, string>>::iterator>
+setDataFlag(s:string &): void