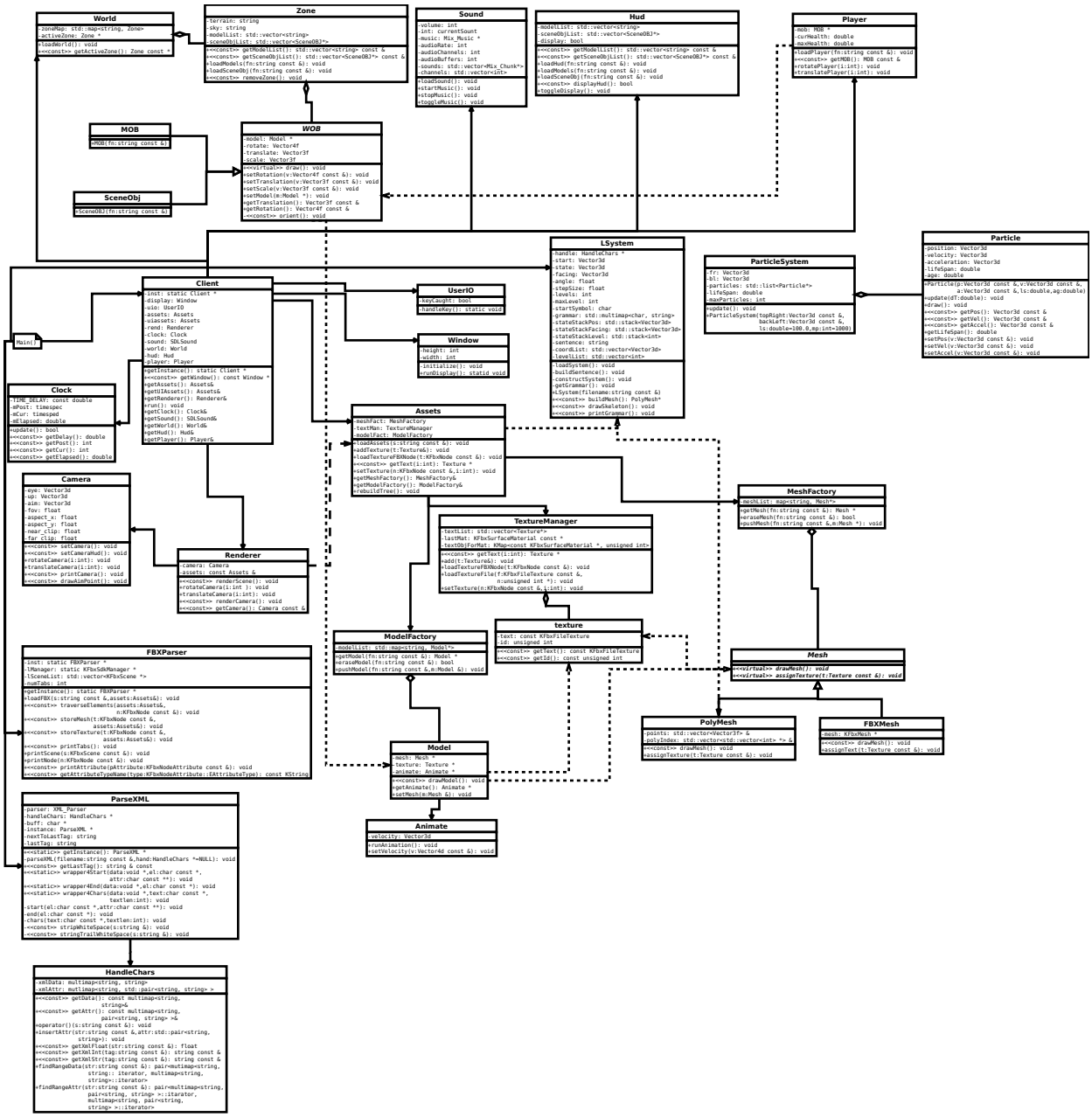
World
-zoneMap: std::map<string, Zone*>
-activeZone: Zone *
+loadWorld(): void
+<<const>> getActiveZone(): Zone const *
Zone
-terrain: string
-sky: string
-modelList: std::vector<string>
-sceneObjList: std::vector<SceneOBJ*>
+<<const>> getModelList(): std::vector<string> const &
+<<const>> getSceneObjList(): std::vector<SceneOBJ*> const &
+loadModels(fn:string const &): void
+loadScenesObj(fn:string const &): void
+<<const>> removeZone(): void
Sound
-volume: int
-int: currentSount
-music: Mix_Music *
-audioRate: int
-audioChannels: int
-audioBuffers: int
-sounds: std::vector<Mix_Chunk*>
-channels: std::vector<int>
+loadSound(): void
+startMusic(): void
+stopMusic(): void
+toggleMusic(): void
Hud
-modelList: std::vector<string>
-sceneObjList: std::vector<SceneOBJ*>
-display: bool
+<<const>> getModelList(): std::vector<string> const &
+<<const>> getSceneObjList(): std::vector<SceneOBJ*> const &
+loadHud(fn:string const &): void
+loadModels(fn:string const &): void
+loadScenesObj(fn:string const &): void
+<<const>> displayHud(): bool
+toggleDisplay(): void
Player
-mob: MOB *
-curHealth: double
-maxHealth: double
+loadPlayer(fn:string const &): void
+<<const>> getMOB(): MOB const &
+rotatePlayer(i:int): void
+translatePlayer(i:int): void
MOB
+MOB(fn:string const &)
SceneObj
+SceneOBJ(fn:string const &)
WOB
-model: Model *
-rotate: Vector4f
-translate: Vector3f
-scale: Vector3f
+<<virtual>> draw(): void
+setRotation(v:Vector4f const &): void
+setTranslation(v:Vector3f const &): void
+setScale(v:Vector3f const &): void
+setModel(m:Model *): void
+getTranslation(): Vector3f const &
+getRotation(): Vector4f const &
+<<const>> orient(): void
LSystem
-handle: HandleChars *
-start: Vector3d
-state: Vector3d
-facing: Vector3d
-angle: float
-stepSize: float
-levels: int
-maxLevel: int
-startSymbol: char
-grammar: std::multimap<char, string>
-stateSStackPos: std::stack<Vector3d>
-stateStackFacing: std::stack<Vector3d>
-stateSStackLevel: std::stack<int>
-sentence: string
-coordList: std::vector<Vector3d>
-levelList: std::vector<int>
-loadSystem(): void
-buildSentance(): void
-constructSystem(): void
-getGrammar(): void
+LSystem(filename:string const &)
+<<const>> buildMesh(): PolyMesh*
+<<const>> drawSkeleton(): void
+<<const>> printGrammar(): void
ParticleSystem
-fr: Vector3d
-bl: Vector3d
-particles: std::list<Particle*>
-lifeSpan: double
-maxParticles: int
+update(): void
+ParticleSystem(topRight:Vector3d const &,
backLeft:Vector3d const &,
ls:double=100.0,mp:int=1000)
Particle
-position: Vector3d
-velocity: Vector3d
-acceleration: Vector3d
-lifeSpan: double
-age: double
+Particle(p:Vector3d const &,v:Vector3d const &,
a:Vector3d const &,ls:double,ag:double)
+update(dT: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
Client
-inst: static Client *
-display: Window
-uio: UserIO
-assets: Assets
-uiassets: Assets
-rend: Renderer
-clock: Clock
-sound: SDLSound
-world: World
-hud: Hud
-player: Player
+getInstance(): static Client *
+<<const>> getWindow(): const Window *
+getAssets(): Assets&
+getUIAssets(): Assets&
+getRenderer(): Renderer&
+run(): void
+getClock(): Clock&
+getSound(): SDLSound&
+getWorld(): World&
+getHud(): Hud&
+getPlayer(): Player&
UserIO
-keyCaught: bool
-handleKey(): static void
Window
-height: int
-width: int
-initialize(): void
-runDisplay(): static void
Main()
Clock
-TIME_DELAY: const double
-mPost: timespec
-mCur: timespec
-mElapsed: double
+update(): bool
+<<const>> getDelay(): double
+<<const>> getPost(): int
+<<const>> getCur(): int
+<<const>> getElapsed(): double
Assets
-meshFact: MeshFactory
-textMan: TextureManager
-modelFact: ModelFactory
+loadAssets(s:string const &): void
+addTexture(t:Texture&): void
+loadTextureFBXNode(t:KFbxNode const &): void
+<<const>> getText(i:int): Texture *
+setTexture(n:KFbxNode const &,i:int): void
+getMeshFactory(): MeshFactory&
+getModelFactory(): ModelFactory&
+rebuildTree(): void
Camera
-eye: Vector3d
-up: Vector3d
-aim: Vector3d
-fov: float
-aspect_x: float
-aspect_y: float
-near_clip: float
-far_clip: float
+<<const>> setCamera(): void
+<<const>> setCameraHud(): void
+rotateCamera(i:int): void
+translateCamera(i:int): void
+<<const>> printCamera(): void
+<<const>> drawAimPoint(): void
Renderer
-camera: Camera
-assets: const Assets &
+<<const>> renderScene(): void
+rotateCamera(i:int): void
+translateCamera(i:int): void
+<<const>> renderCamera(): void
+<<const>> getCamera(): Camera const &
MeshFactory
-meshList: map<string, Mesh*>
+getMesh(fn:string const &): Mesh *
+eraseMesh(fn:string const &): bool
+pushMesh(fn:string const &,m:Mesh *): void
TextureManager
-textList: std::vector<Texture*>
-lastMat: KFbxSurfaceMaterial const *
-textObjForMat: KMap<const KFbxSurfaceMaterial *, unsigned int>
+<<const>> getText(i:int): Texture *
+add(t:Texture&): void
+loadTextureFBXNode(t:KFbxNode const &): void
+loadTextureFile(f:KFbxFileTexture const &,
n:unsigned int *): void
+setTexture(n:KFbxNode const &,i:int): void
texture
-text: const KFbxFileTexture
-id: unsigned int
+<<const>> getText(): const KFbxFileTexture
+<<const>> getId(): const unsigned int
ModelFactory
-modelList: std::map<string, Model*>
+getModel(fn:string const &): Model *
+eraseModel(fn:string const &): bool
+pushModel(fn:string const &,m:Model &): void
Mesh
+<<virtual>> drawMesh(): void
+<<virtual>> assignTexture(t:Texture const &): void
FBXParser
-inst: static FBXParser *
-lManager: static KFbxSdkManager *
-lScenelist: std::vector<KFbxScene *>
-numTabs: 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>> getAttributeTypeName(type:KFbxNodeAttribute::EAttributeType): const KString
Model
-mesh: Mesh *
-texture: Texture *
-animate: Animate *
+<<const>> drawModel(): void
+getAnimate(): Animate *
+setMesh(m:Mesh &): void
PolyMesh
-points: std::vector<Vector3f> &
-polyIndex: std::vector<std::vector<int> *> &
+<<const>> drawMesh(): void
+assignTexture(t:Texture const &): void
FBXMesh
-mesh: KFbxMesh *
+<<const>> drawMesh(): void
+assignText(t:Texture const &): void
Animate
-velocity: Vector3d
+runAnimation(): void
+setVelocity(v:Vector4d const &): void
ParseXML
-parser: XML_Parser
-handleChars: HandleChars *
-buff: char *
-instance: ParseXML *
-nextToLastTag: string
-lastTag: string
+<<static>> getInstance(): ParseXML *
-parseXML(filename:string const &,hand:HandleChars *=NULL): void
+<<const>> getLastTag(): string & const
+<<static>> wrapper4Start(data:void *,el:char const *,
attr:char const **): void
+<<static>> wrapper4End(data:void *,el:char const *): void
+<<static>> wrapper4Chars(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>> stripWhiteSpace(s:string &): void
-<<const>> stringTrailWhiteSpace(s:string &): void
HandleChars
-xmlData: multimap<string, string>
-xmlAttr: mutlimap<string, std::pair<string, string> >
+<<const>> getData(): const multimap<string,
string>&
+<<const>> getAttr(): const multimap<string,
pair<string, string> >&
+operator()(s:string const &): void
+insertAttr(str:string const &,attr:std::pair<string,
string>): void
+<<const>> getXmlFloat(str:string const &): float
+<<const>> getXmlInt(tag:string const &): string const &
+<<const>> getXmlStr(tag:string const &): string const &
+findRangeData(str:string const &): pair<mutimap<string,
string:: iterator, multimap<string,
string>::iterator>
+findRangeAttr(str:string const &): pair<multimap<string,
pair<string, string> >::iterator,
multimap<string, pair<string,
string> >::iterator>