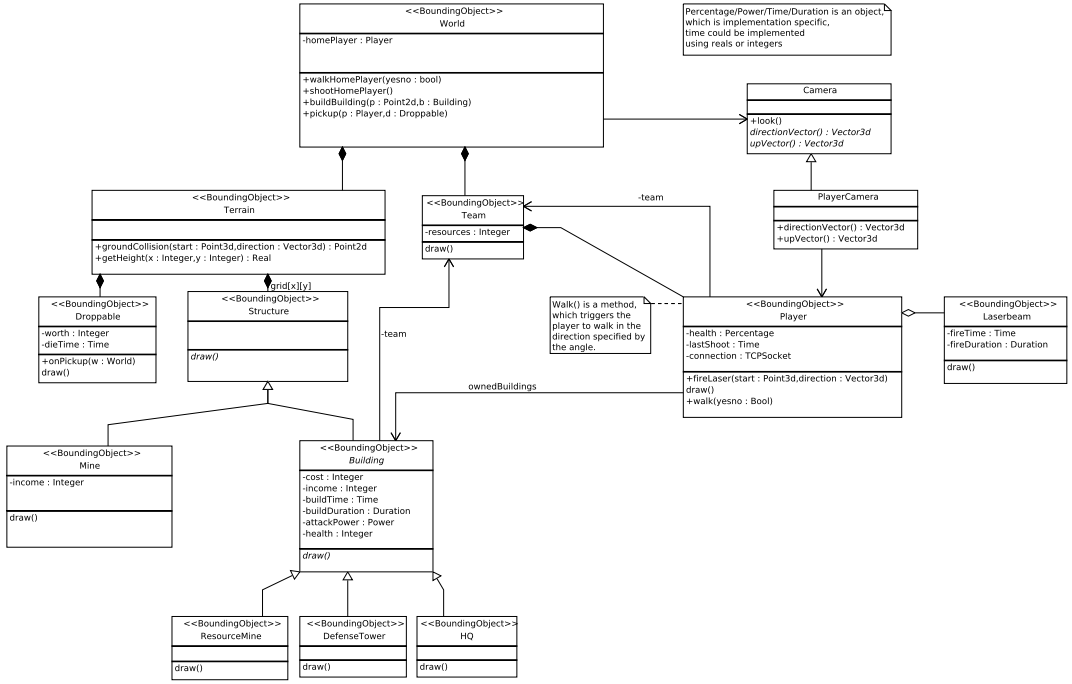
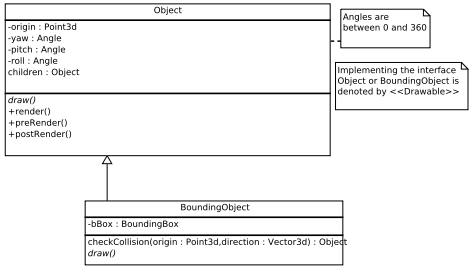
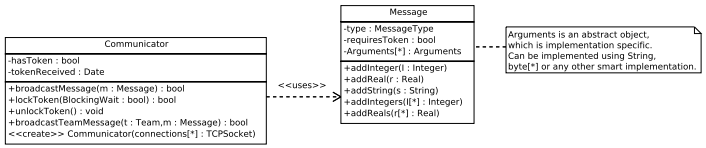

<<BoundingObject>>
World
-homePlayer : Player
+walkHomePlayer(yesno : bool)
+shootHomePlayer()
+buildBuilding(p : Point2d,b : Building)
+pickup(p : Player,d : Droppable)
Percentage/Power/Time/Duration is an object, which is implementation specific, time could be implemented using reals or integers
Camera
+look()
directionVector() : Vector3d
upVector() : Vector3d
<<BoundingObject>>
Terrain
+groundCollision(start : Point3d,direction : Vector3d) : Point2d
+getHeight(x : Integer,y : Integer) : Real
<<BoundingObject>>
Team
-resources : Integer
draw()
-team
PlayerCamera
+directionVector() : Vector3d
+upVector() : Vector3d
grid[x][y]
<<BoundingObject>>
Droppable
-worth : Integer
-dieTime : Time
+onPickup(w : World)
draw()
<<BoundingObject>>
Structure
draw()
-team
Walk() is a method, which triggers the player to walk in the direction specified by the angle.
<<BoundingObject>>
Player
-health : Percentage
-lastShoot : Time
-connection : TCPSocket
+fireLaser(start : Point3d,direction : Vector3d)
draw()
+walk(yesno : Bool)
<<BoundingObject>>
Laserbeam
-fireTime : Time
-fireDuration : Duration
draw()
ownedBuildings
<<BoundingObject>>
Mine
-income : Integer
draw()
<<BoundingObject>>
Building
-cost : Integer
-income : Integer
-buildTime : Time
-buildDuration : Duration
-attackPower : Power
-health : Integer
draw()
<<BoundingObject>>
ResourceMine
draw()
<<BoundingObject>>
DefenseTower
draw()
<<BoundingObject>>
HQ
draw()


Object
-origin : Point3d
-yaw : Angle
-pitch : Angle
-roll : Angle
children : Object
draw()
+render()
+preRender()
+postRender()
Angles are between 0 and 360
Implementing the interface Object or BoundingObject is denoted by <<Drawable>>
BoundingObject
-bBox : BoundingBox
checkCollision(origin : Point3d,direction : Vector3d) : Object
draw()


Communicator
-hasToken : bool
-tokenReceived : Date
+broadcastMessage(m : Message) : bool
+lockToken(BlockingWait : bool) : bool
+unlockToken() : void
+broadcastTeamMessage(t : Team,m : Message) : bool
<<create>> Communicator(connections[*] : TCPSocket)
<<uses>>
Message
-type : MessageType
-requiresToken : bool
-Arguments[*] : Arguments
+addInteger(i : Integer)
+addReal(r : Real)
+addString(s : String)
+addIntegers(i[*] : Integer)
+addReals(r[*] : Real)
Arguments is an abstract object, which is implementation specific. Can be implemented using String, byte[*] or any other smart implementation.