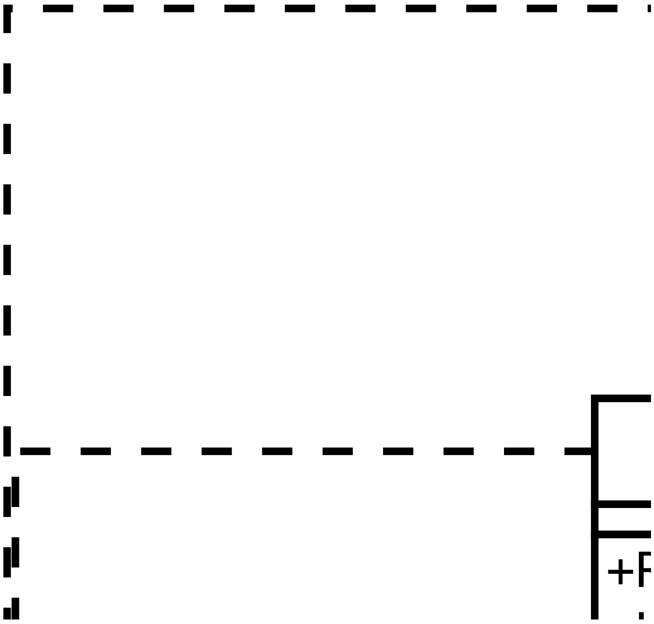
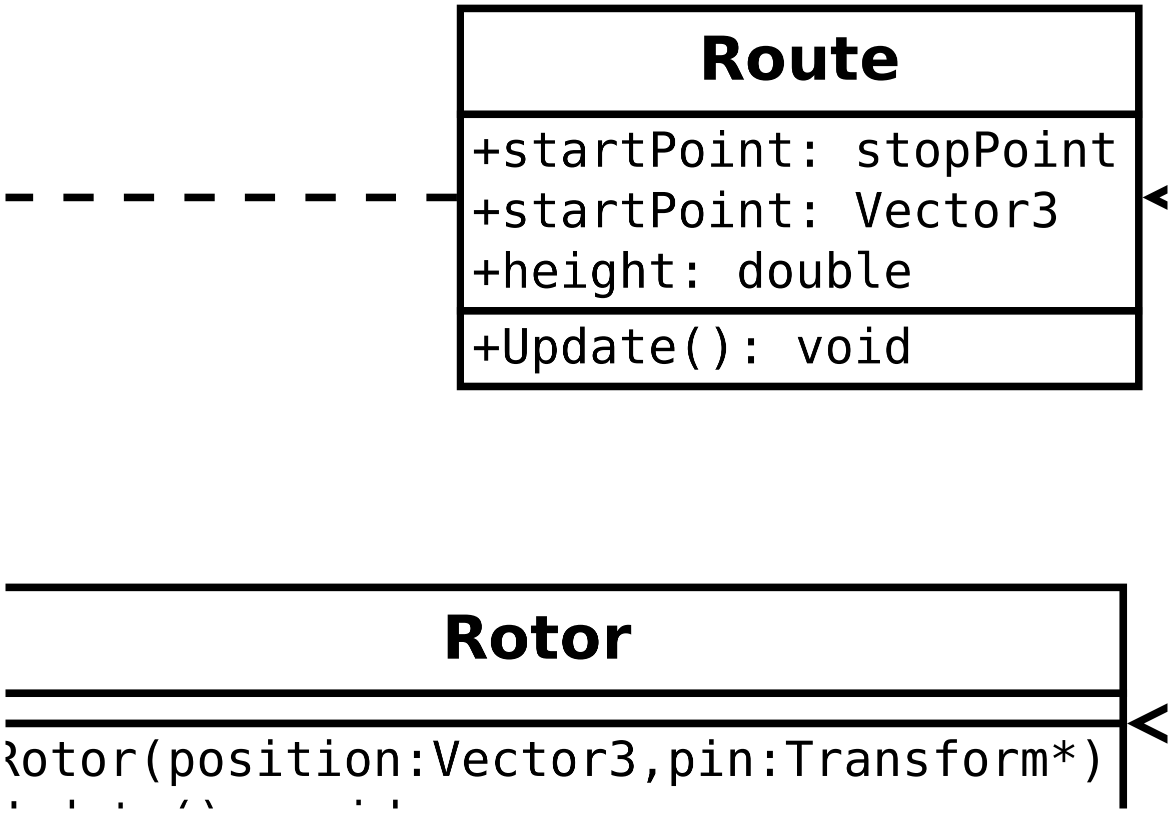
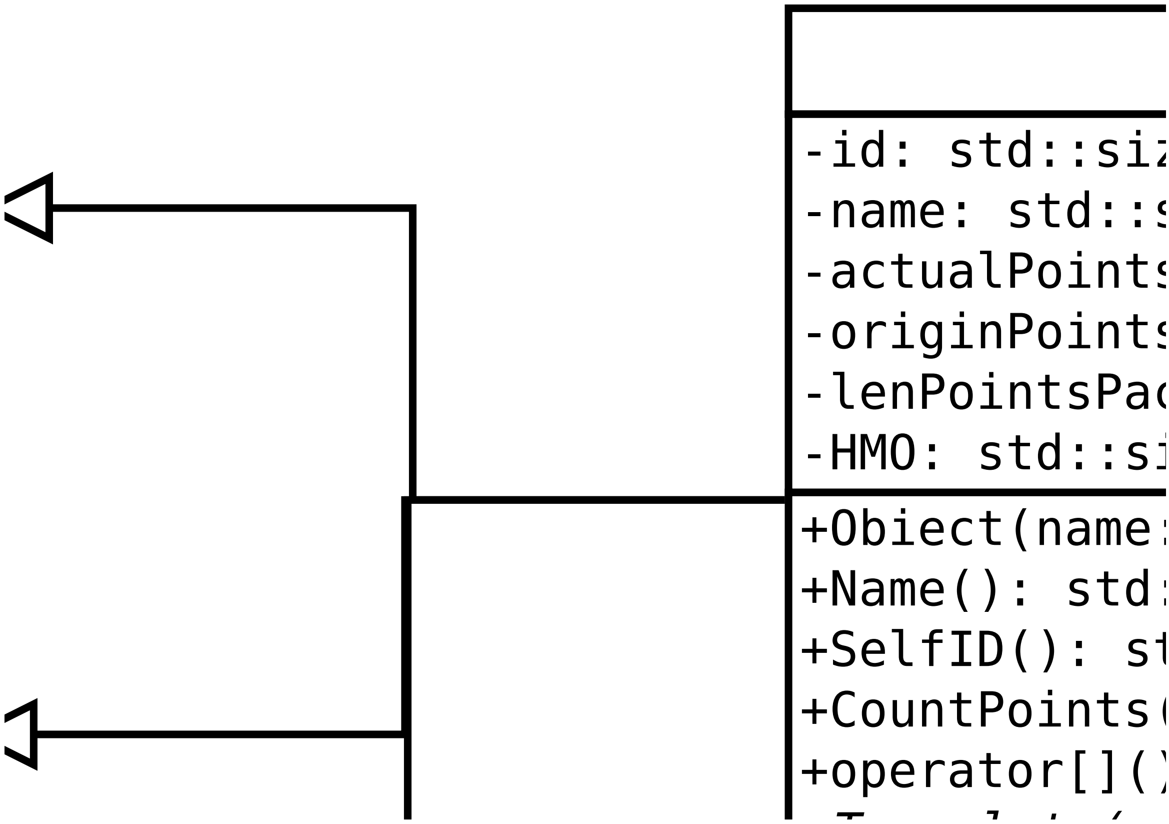
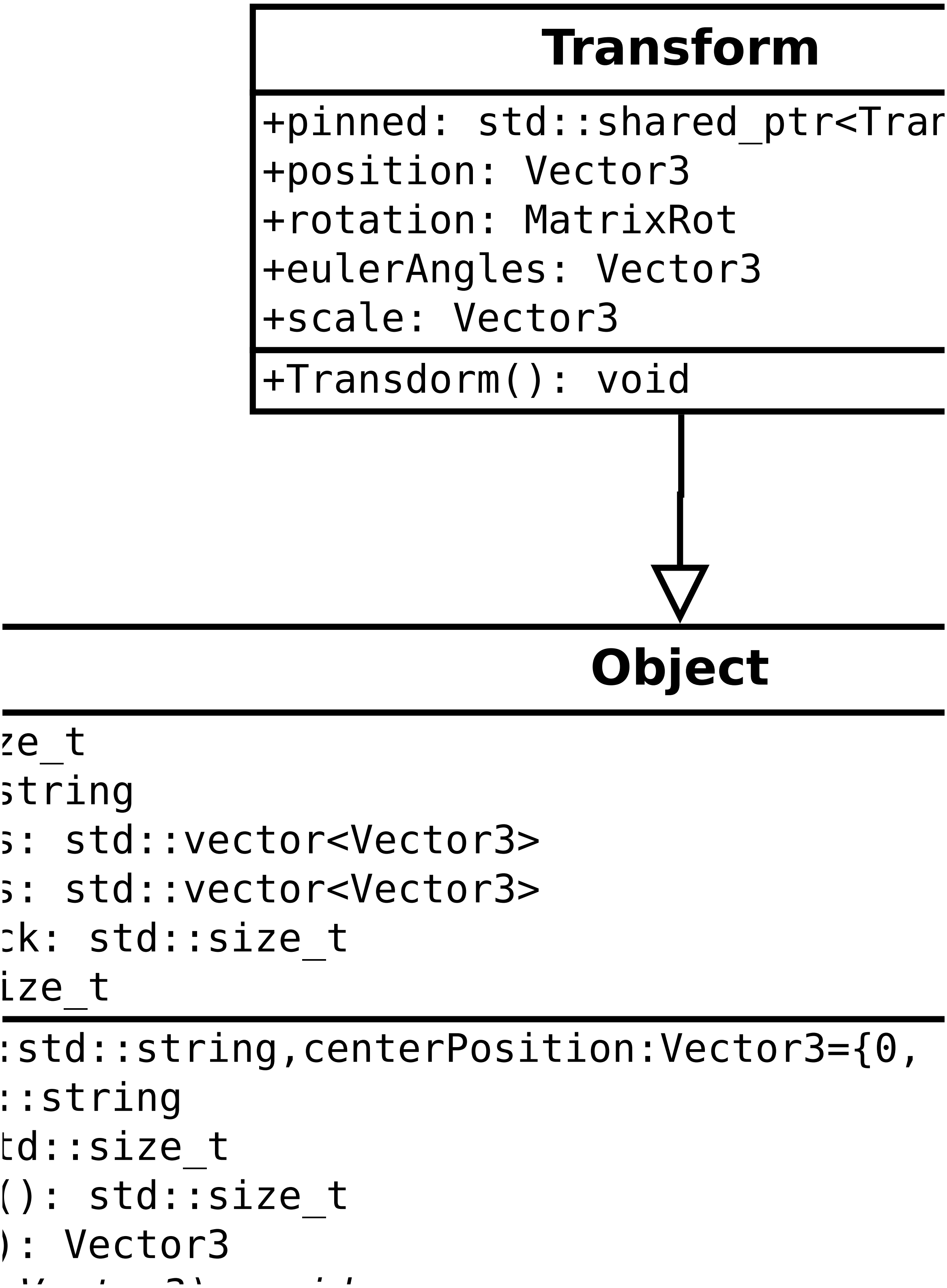

+F


Route
+startPoint: stopPoint
+startPoint: Vector3
+height: double
+Update(): void
Rotor
Rotor(position:Vector3,pin:Transform*)


-id: std::siz
-name: std::s
-actualPoints
-originPoints
-lenPointsPac
-HMO: std::si
+Obiect(name:
+Name(): std:
+SelfID(): st
+CountPoints(
+operator[]()



Transform

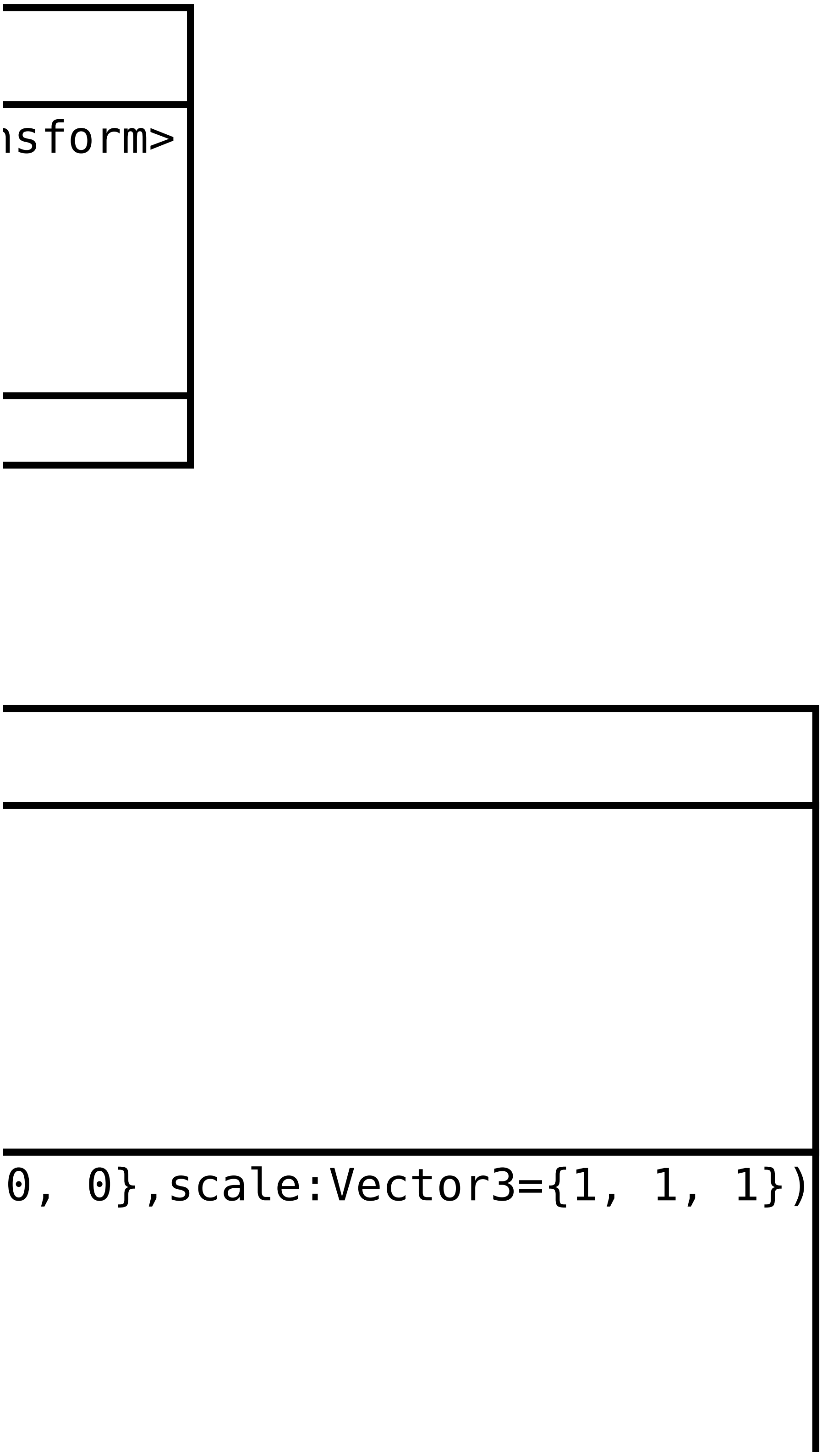
+pinned: std::shared_ptr<Trar
+position: Vector3
+rotation: MatrixRot
+eulerAngles: Vector3
+scale: Vector3

+Transdorm(): void

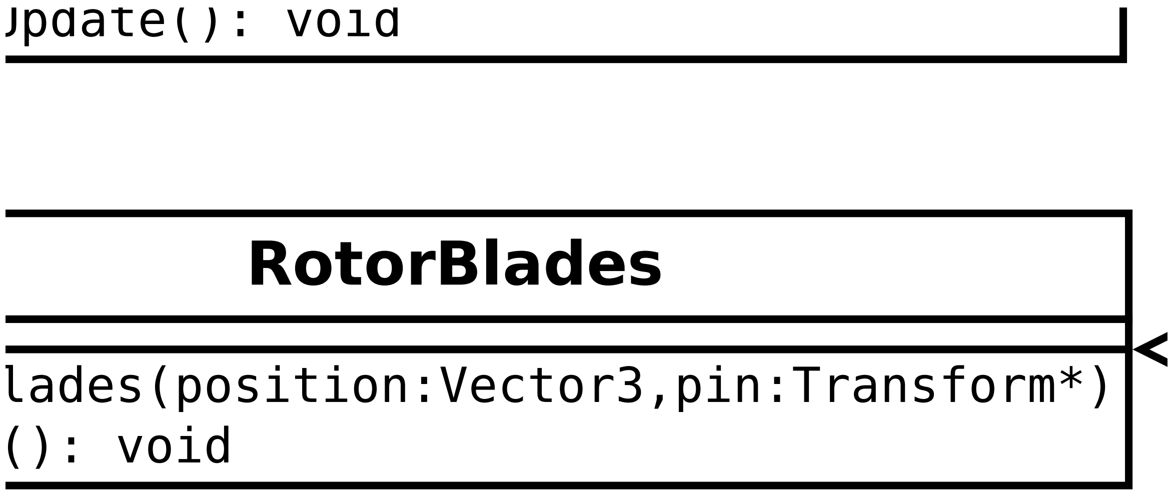
Object

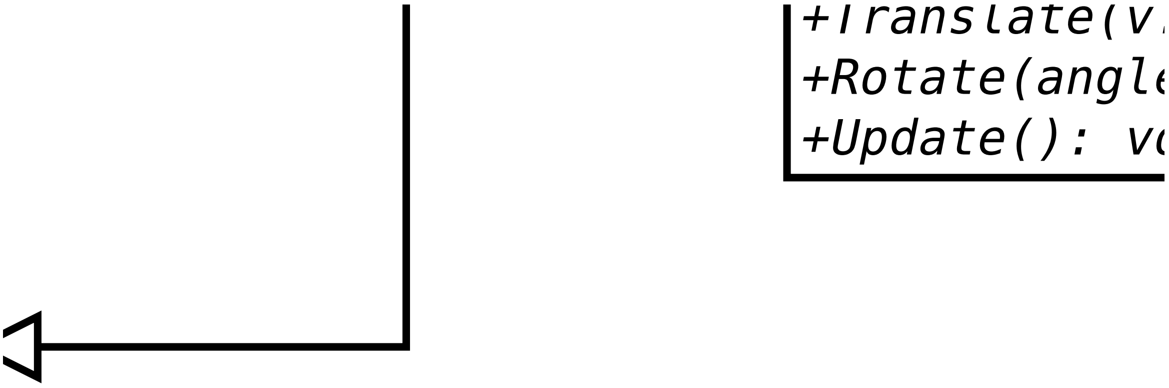
ze_t
string
s: std::vector<Vector3>
s: std::vector<Vector3>
ck: std::size_t
ize_t

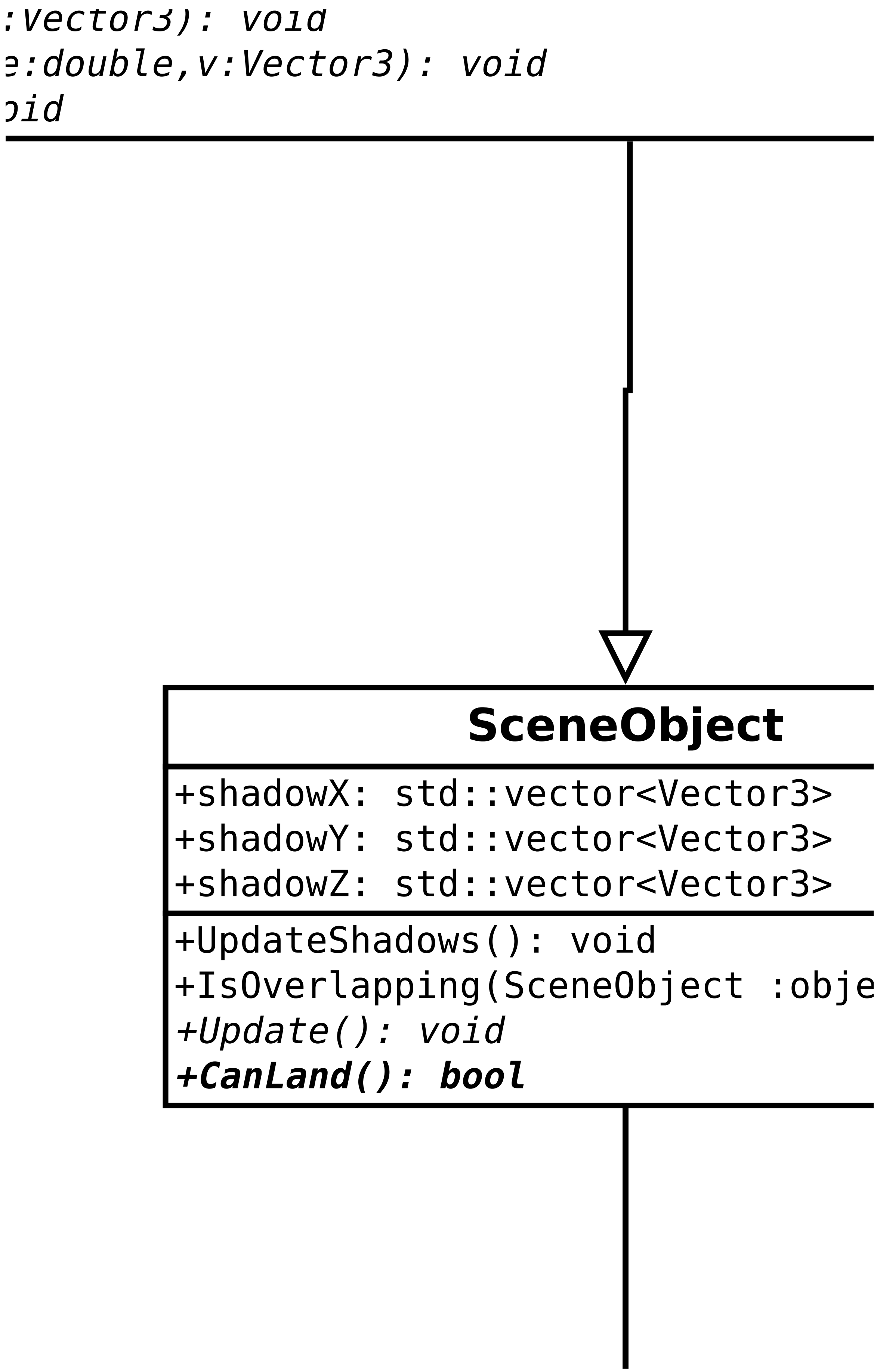
:std::string,centerPosition:Vector3={0,
::string
td::size_t
(): std::size_t
): Vector3
Vector3): void

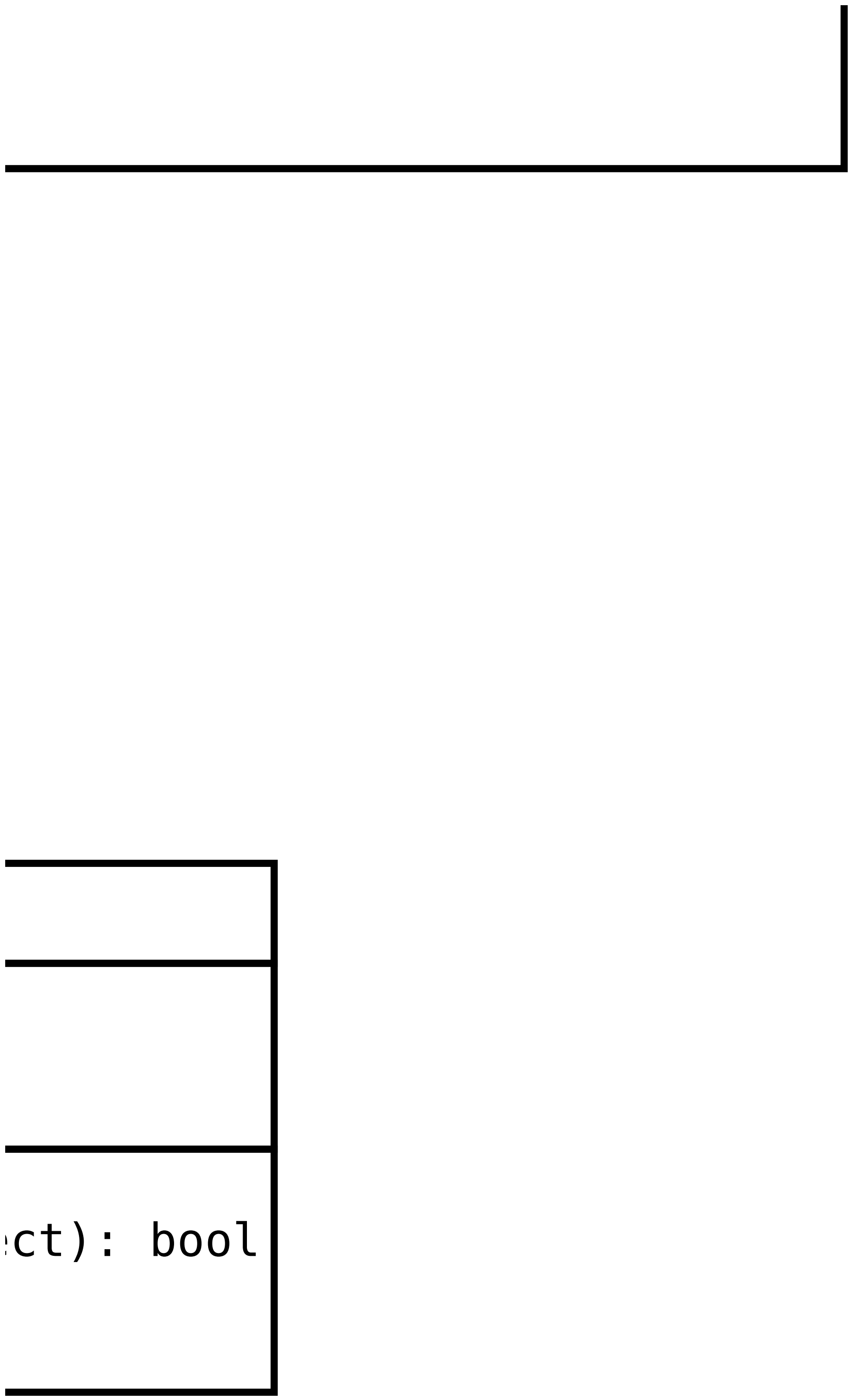

nsform>
0, 0},scale:Vector3={1, 1, 1})

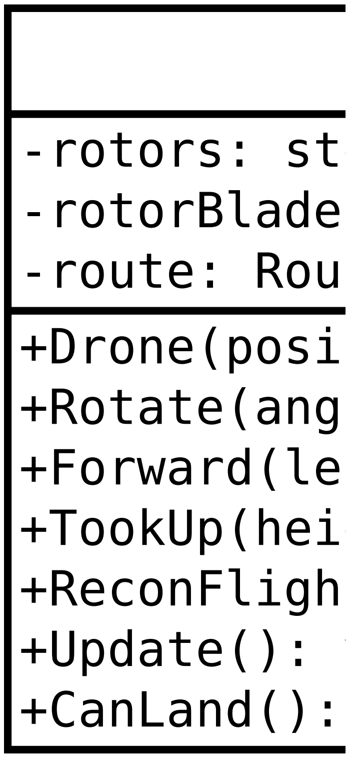
+U
+RotorB
+Update

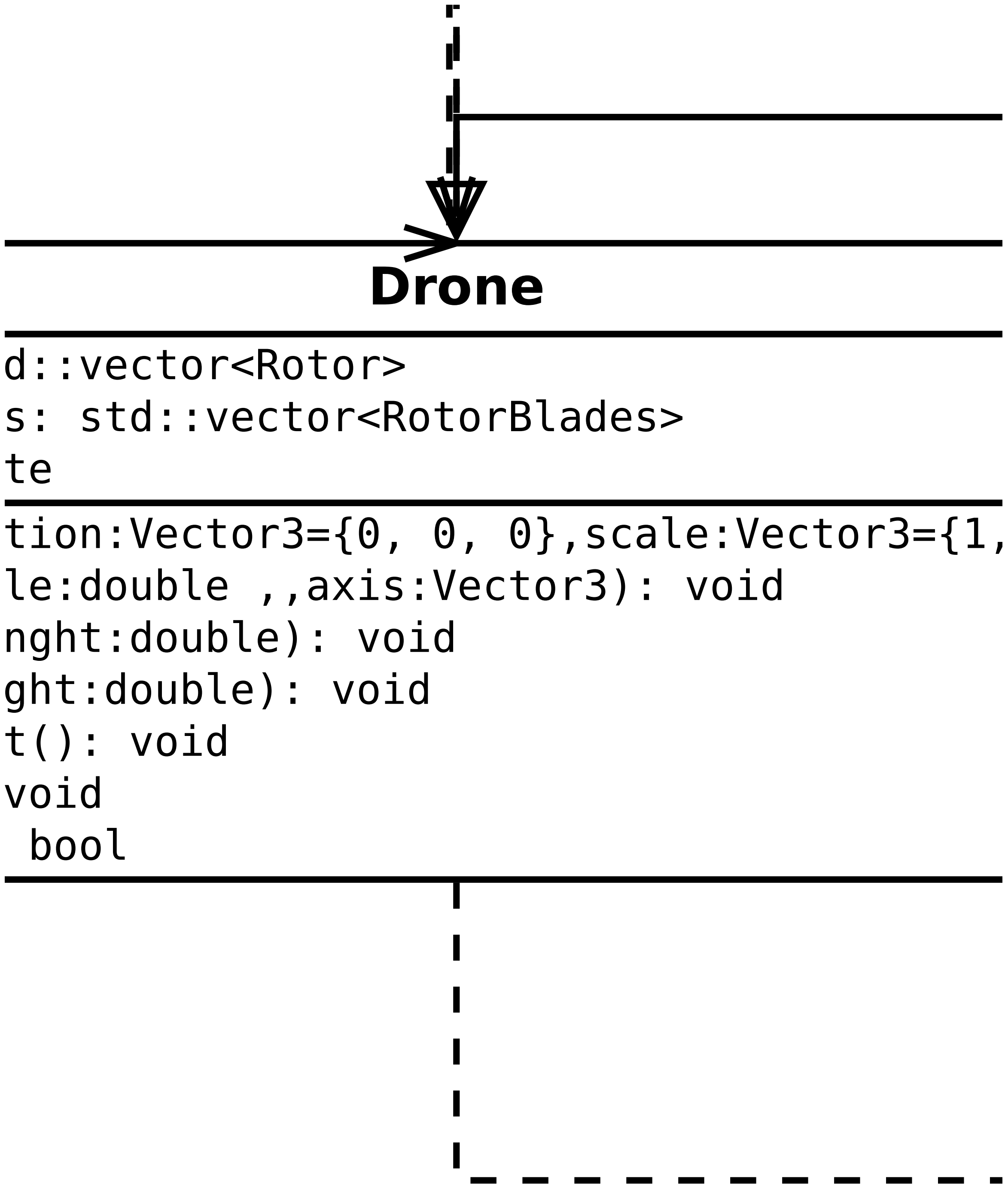

Jpdate(): void
RotorBlades
lades(position:Vector3,pin:Transform*)
(): void


+Translate(v
+Rotate(angle
+Update(): vo


:Vector3): void
e:double,v:Vector3): void
oid
SceneObject
+shadowX: std::vector<Vector3>
+shadowY: std::vector<Vector3>
+shadowZ: std::vector<Vector3>
+UpdateShadows(): void
+IsOverlapping(SceneObject :obje
+Update(): void
+CanLand(): bool


ect): bool

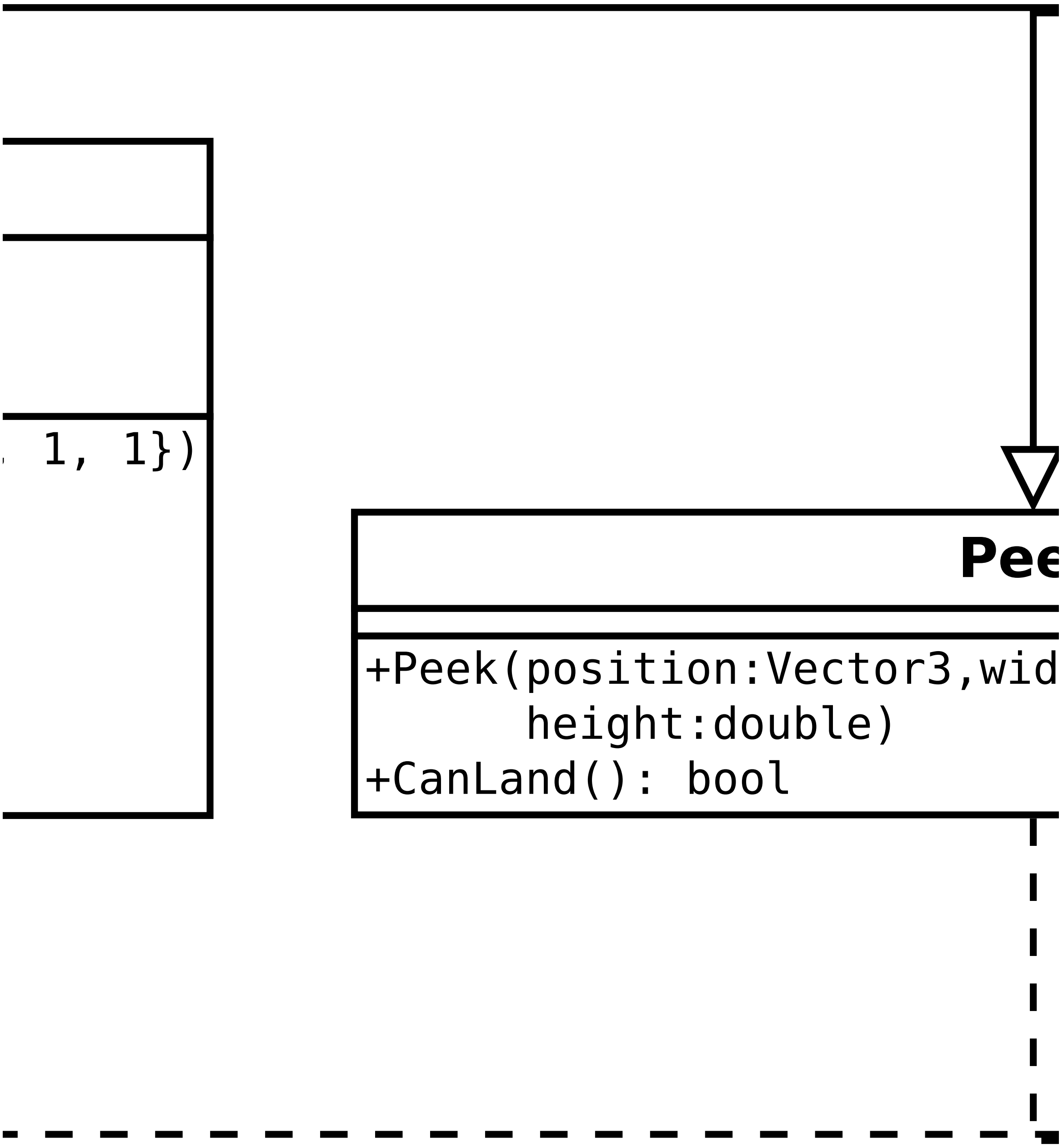

-rotors: st
-rotorBlade
-route: Rou
+Drone(posi
+Rotate(ang
+Forward(le
+TookUp(hei
+ReconFligh
+Update():
+CanLand():

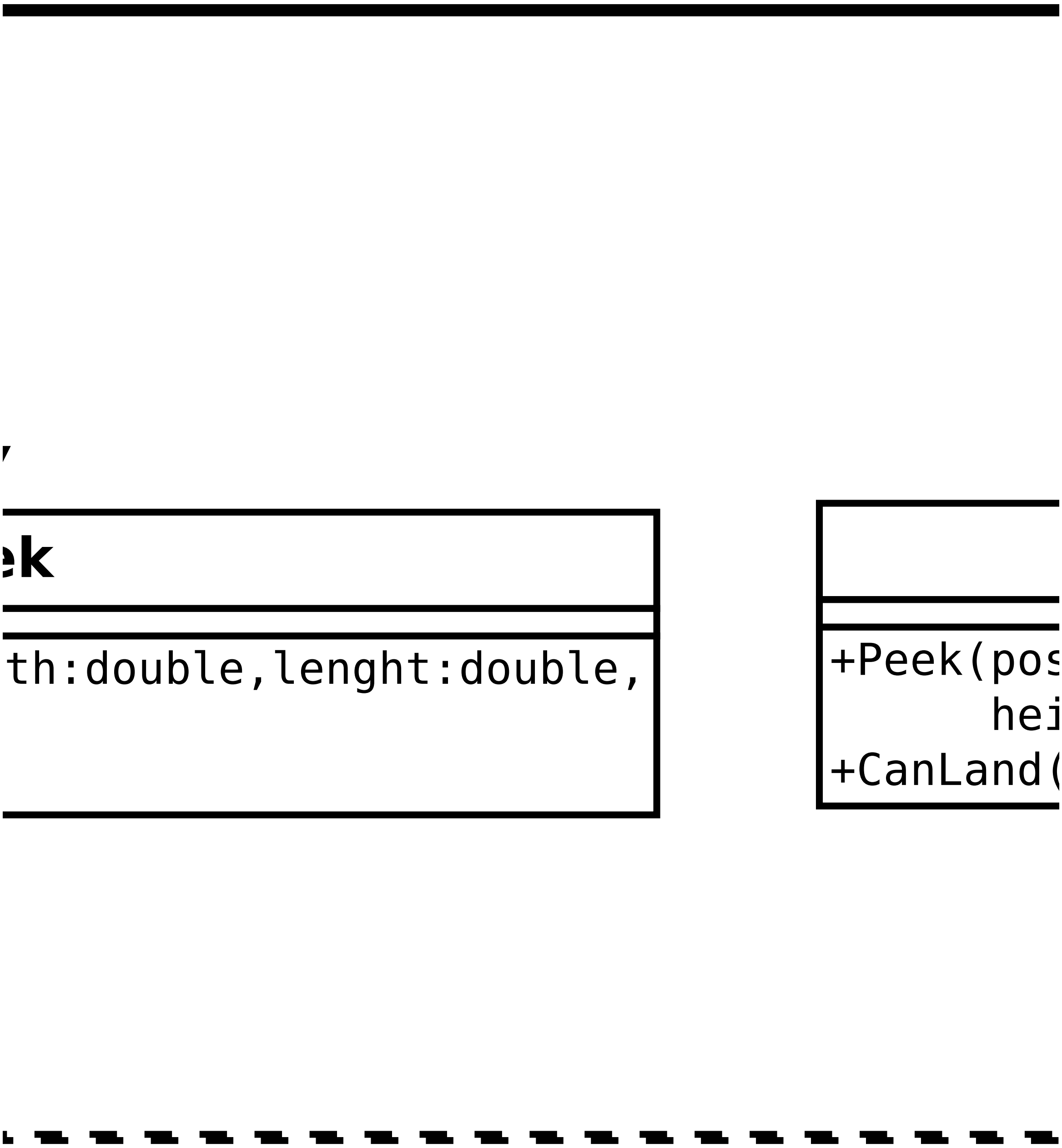


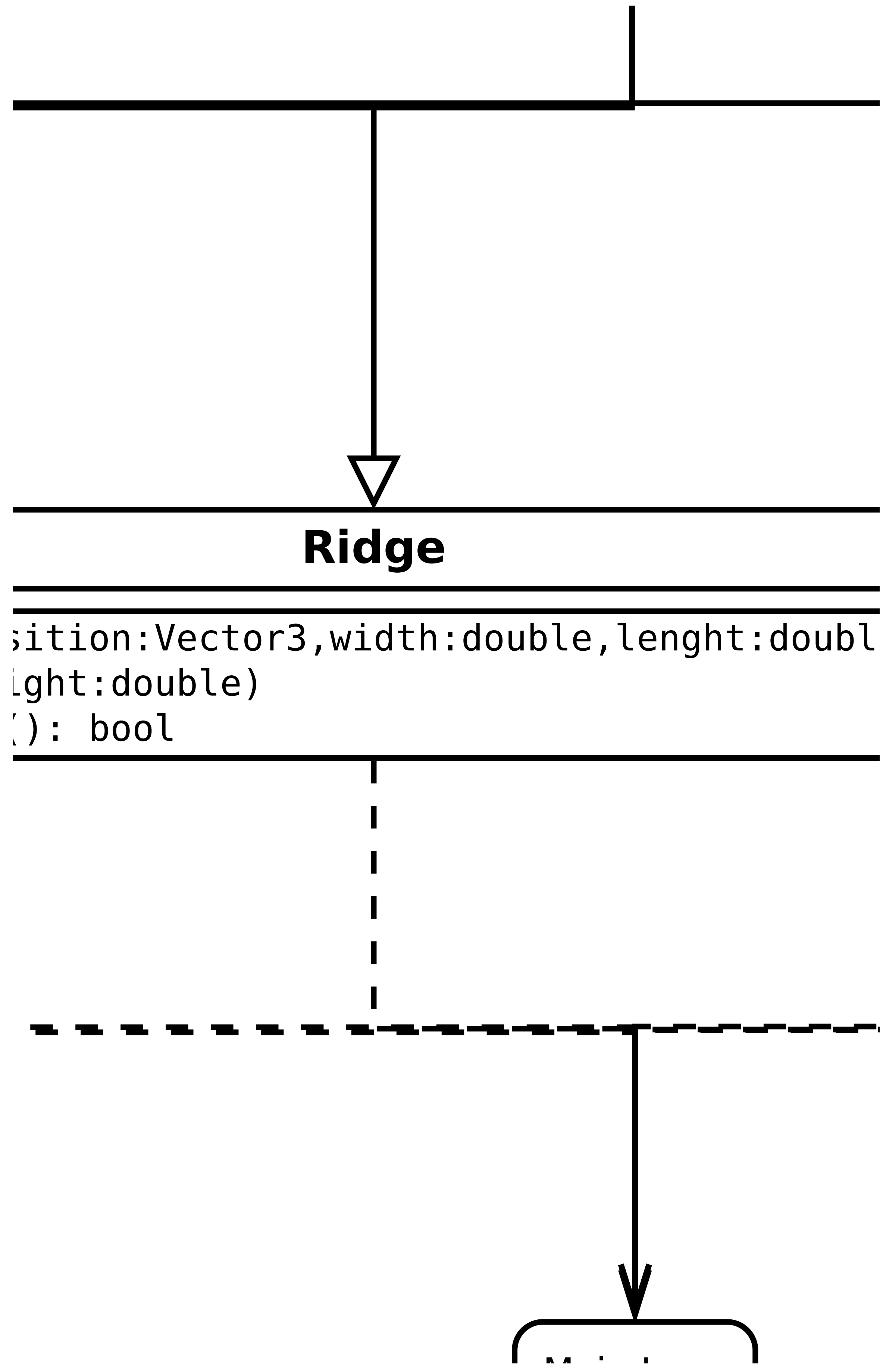
Drone

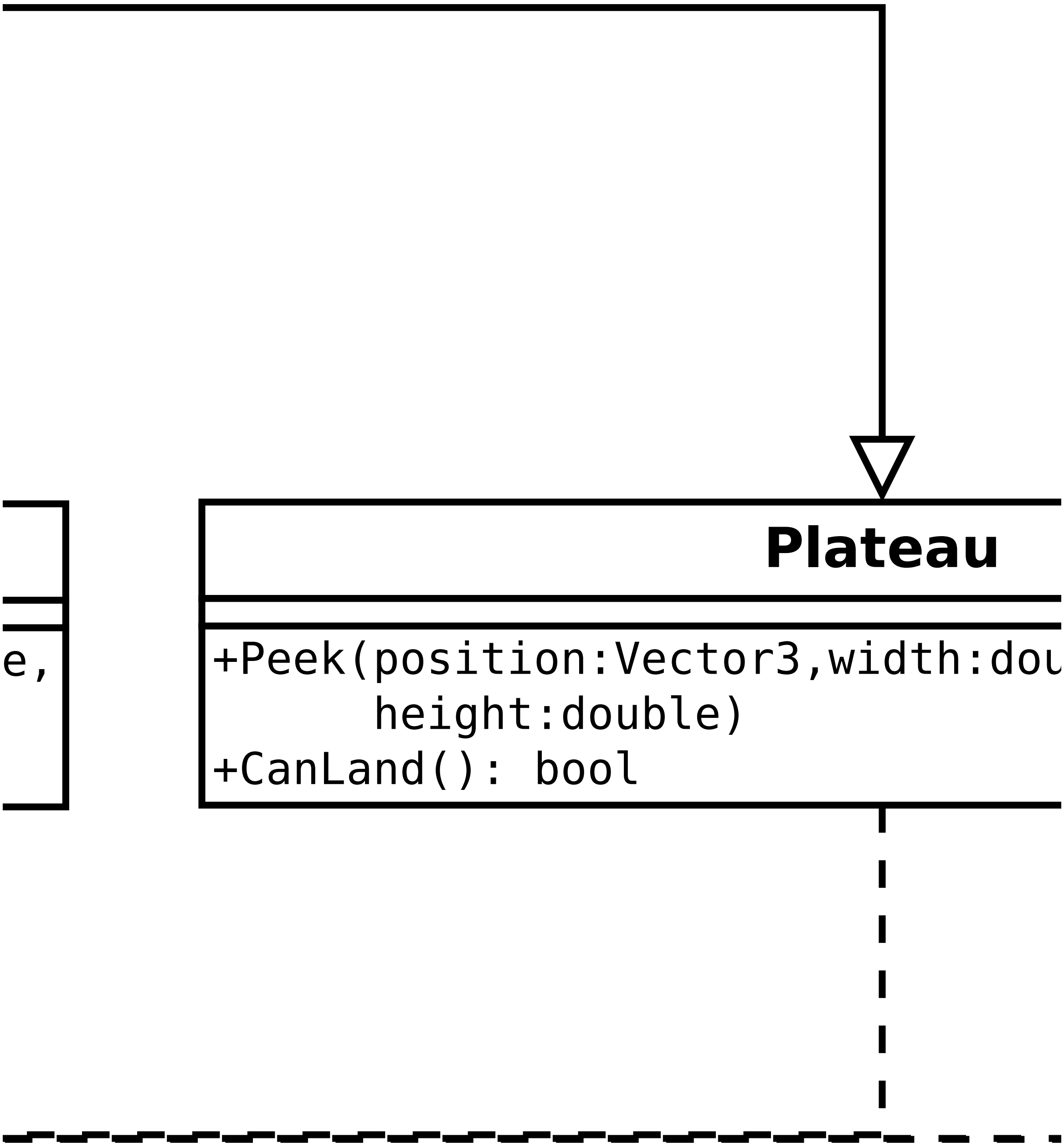
d::vector<Rotor>
s: std::vector<RotorBlades>
te

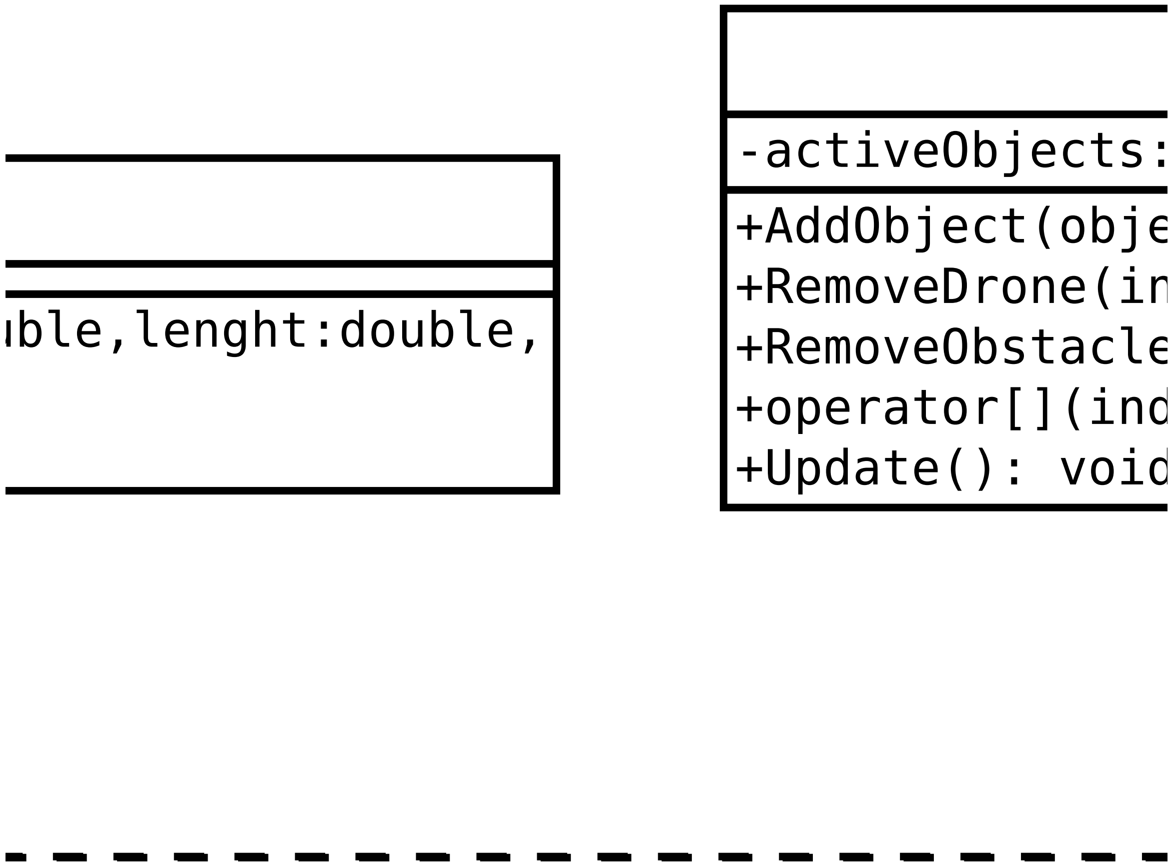
tion:Vector3={0, 0, 0},scale:Vector3={1,
le:double ,,axis:Vector3): void
nght:double): void
ght:double): void
t(): void
void
 bool

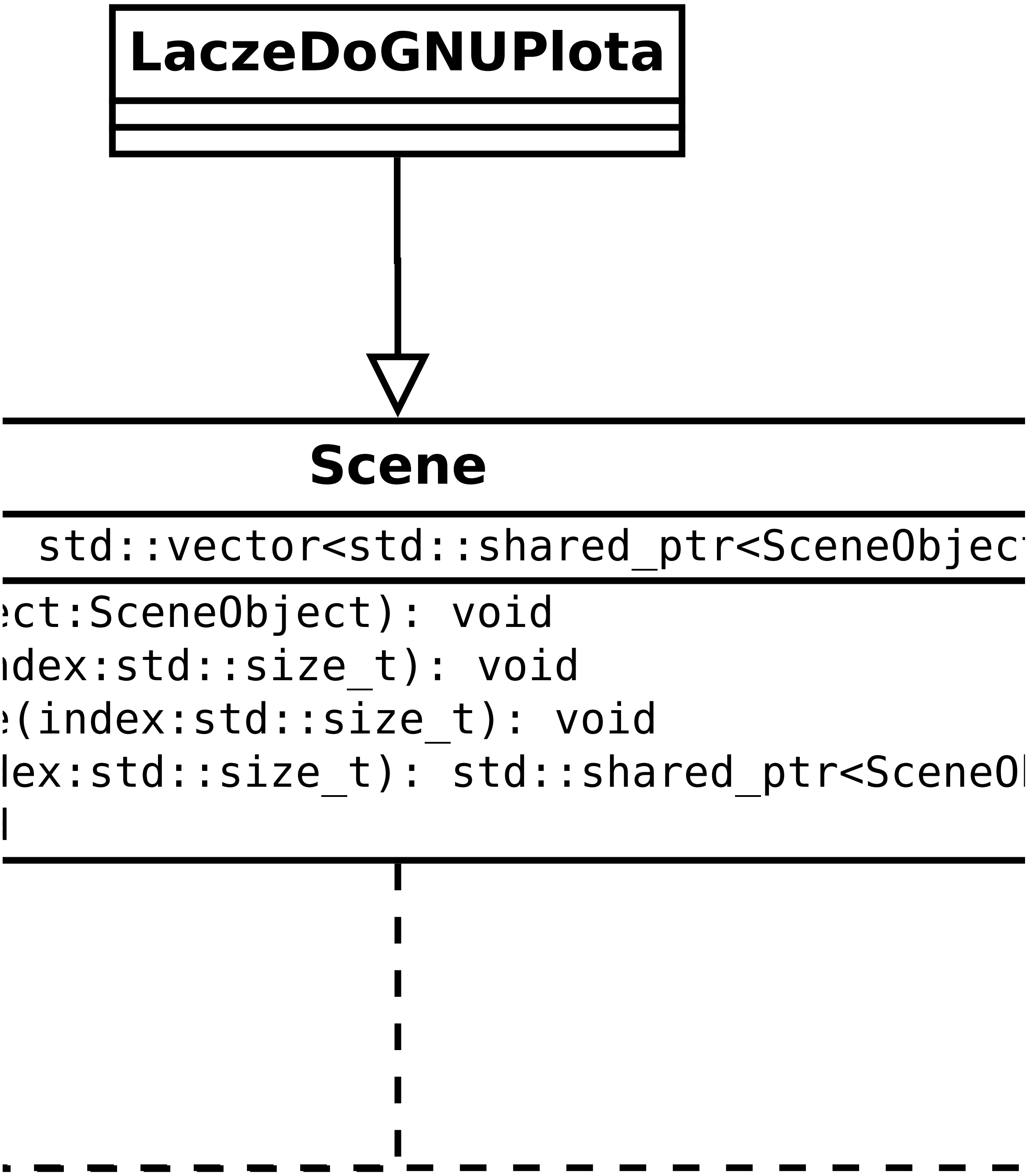

1, 1})
Pee
+Peek(position:Vector3,wid
height:double)
+CanLand(): bool


ek
th:double,lenght:double,
+Peek(pos
hei
+CanLand(


Ridge
sition:Vector3,width:double,lenght:doubl
ight:double)
(): bool


e,
Plateau
+Peek(position:Vector3,width:dou
height:double)
+CanLand(): bool


uble,lenght:double,
-activeObjects:
+AddObject(obje
+RemoveDrone(in
+RemoveObstacle
+operator[](ind
+Update(): void


LaczeDoGNUPlota
Scene
std::vector<std::shared_ptr<SceneObjec
ect:SceneObject): void
ndex:std::size_t): void
e(index:std::size_t): void
lex:std::size_t): std::shared_ptr<SceneOb