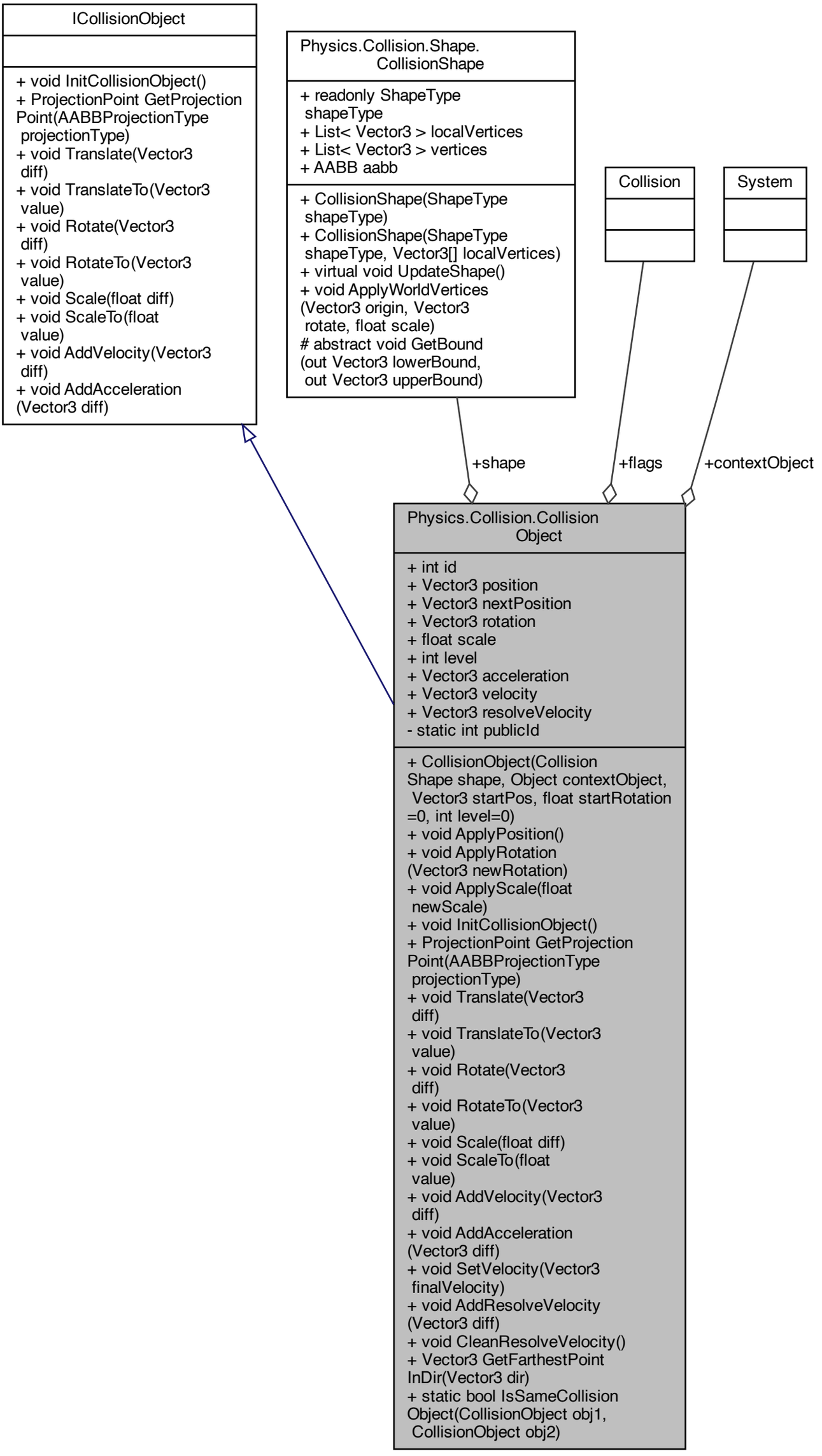
ICollisionObject
+ void InitCollisionObject()
+ ProjectionPoint GetProjection Point(AABBProjectionType projectionType)
+ void Translate(Vector3 diff)
+ void TranslateTo(Vector3 value)
+ void Rotate(Vector3 diff)
+ void RotateTo(Vector3 value)
+ void Scale(float diff)
+ void ScaleTo(float value)
+ void AddVelocity(Vector3 diff)
+ void AddAcceleration (Vector3 diff)
Physics.Collision.Shape. CollisionShape
+ readonly ShapeType shapeType
+ List< Vector3 > localVertices
+ List< Vector3 > vertices
+ AABB aabb
+ CollisionShape(ShapeType shapeType)
+ CollisionShape(ShapeType shapeType, Vector3[] localVertices)
+ virtual void UpdateShape()
+ void ApplyWorldVertices (Vector3 origin, Vector3 rotate, float scale)
# abstract void GetBound (out Vector3 lowerBound, out Vector3 upperBound)
Collision
System
+shape
+flags
+contextObject
Physics.Collision.Collision Object
+ int id
+ Vector3 position
+ Vector3 nextPosition
+ Vector3 rotation
+ float scale
+ int level
+ Vector3 acceleration
+ Vector3 velocity
+ Vector3 resolveVelocity
- static int publicId
+ CollisionObject(Collision Shape shape, Object contextObject, Vector3 startPos, float startRotation =0, int level=0)
+ void ApplyPosition()
+ void ApplyRotation (Vector3 newRotation)
+ void ApplyScale(float newScale)
+ void InitCollisionObject()
+ ProjectionPoint GetProjection Point(AABBProjectionType projectionType)
+ void Translate(Vector3 diff)
+ void TranslateTo(Vector3 value)
+ void Rotate(Vector3 diff)
+ void RotateTo(Vector3 value)
+ void Scale(float diff)
+ void ScaleTo(float value)
+ void AddVelocity(Vector3 diff)
+ void AddAcceleration (Vector3 diff)
+ void SetVelocity(Vector3 finalVelocity)
+ void AddResolveVelocity (Vector3 diff)
+ void CleanResolveVelocity()
+ Vector3 GetFarthestPoint InDir(Vector3 dir)
+ static bool IsSameCollision Object(CollisionObject obj1, CollisionObject obj2)