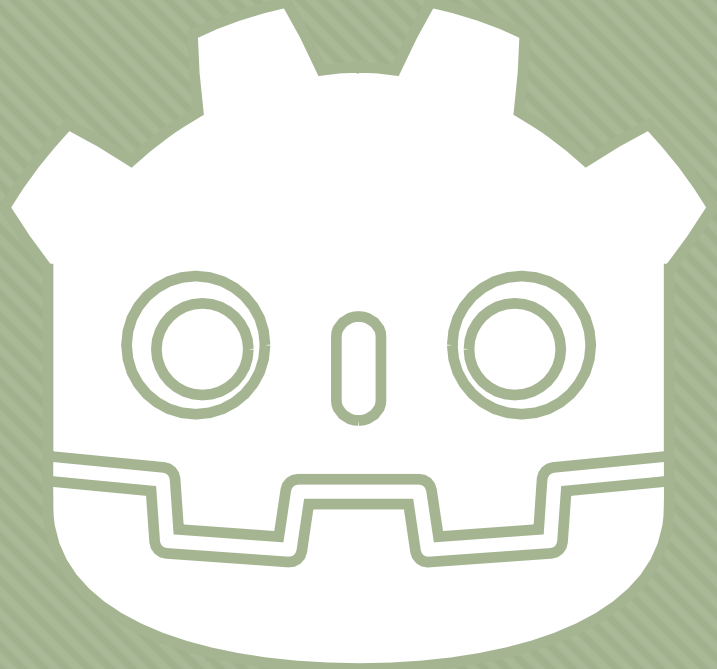
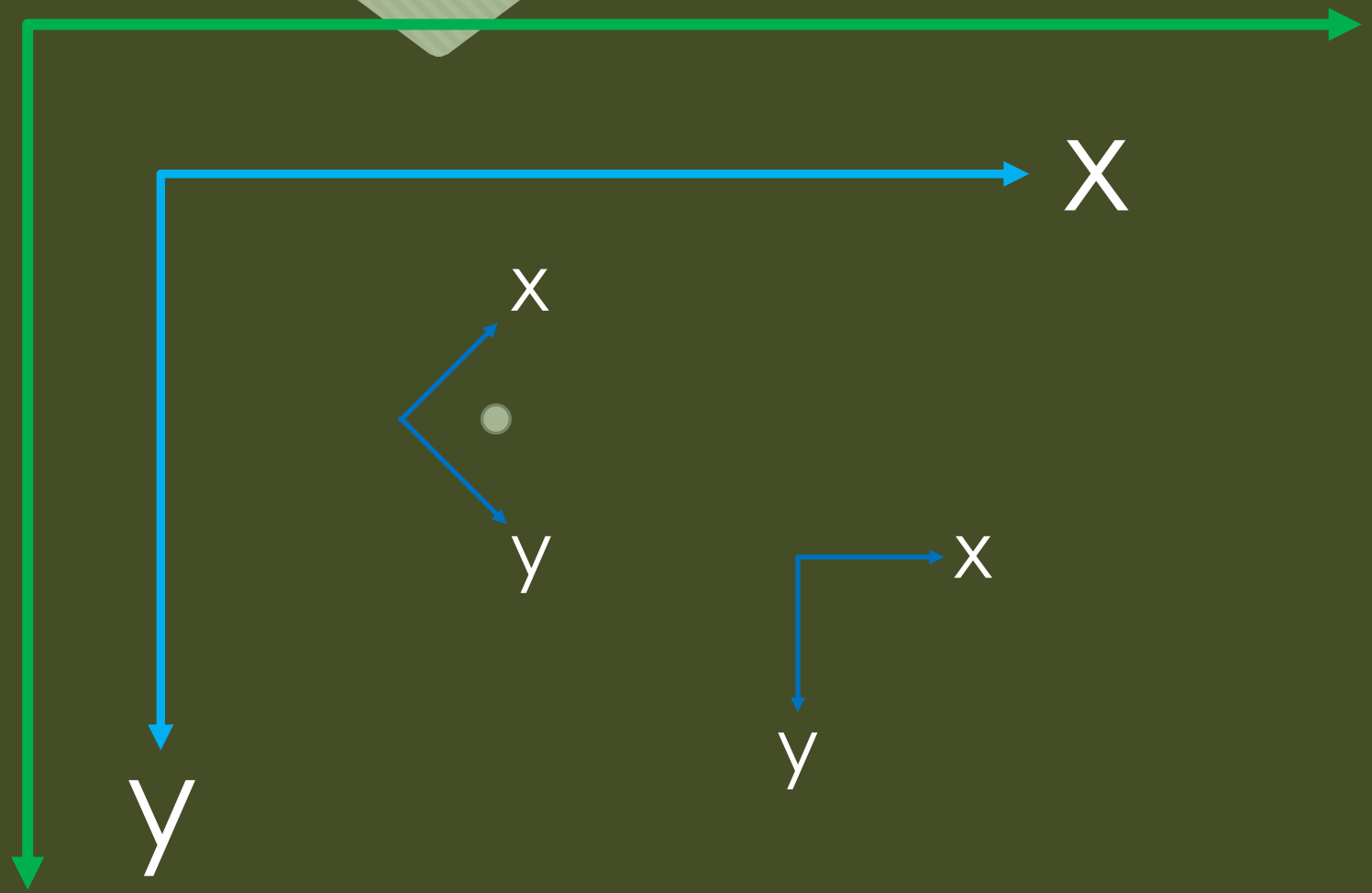


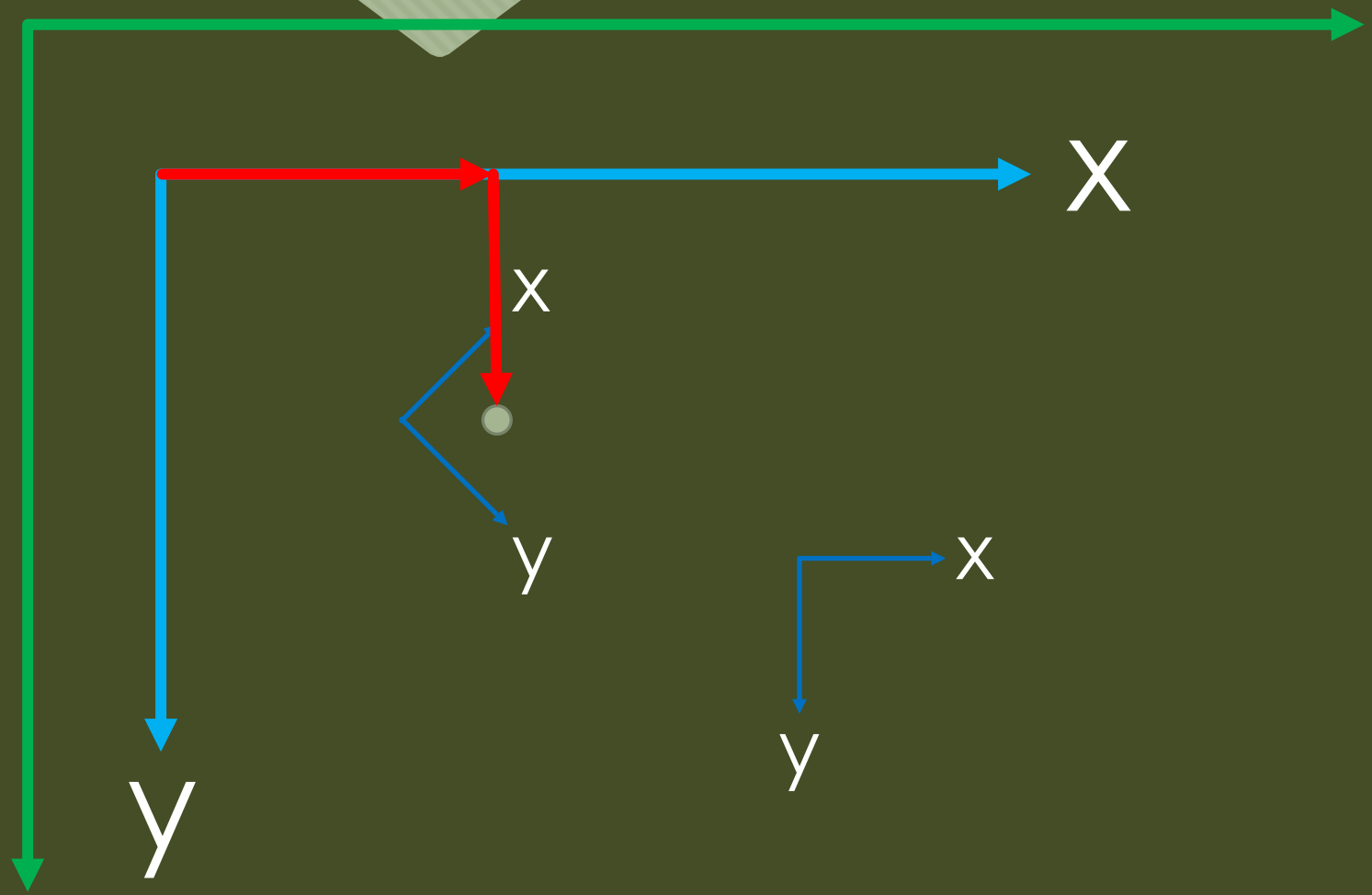
C# Godot Tutorial 7

Transformations

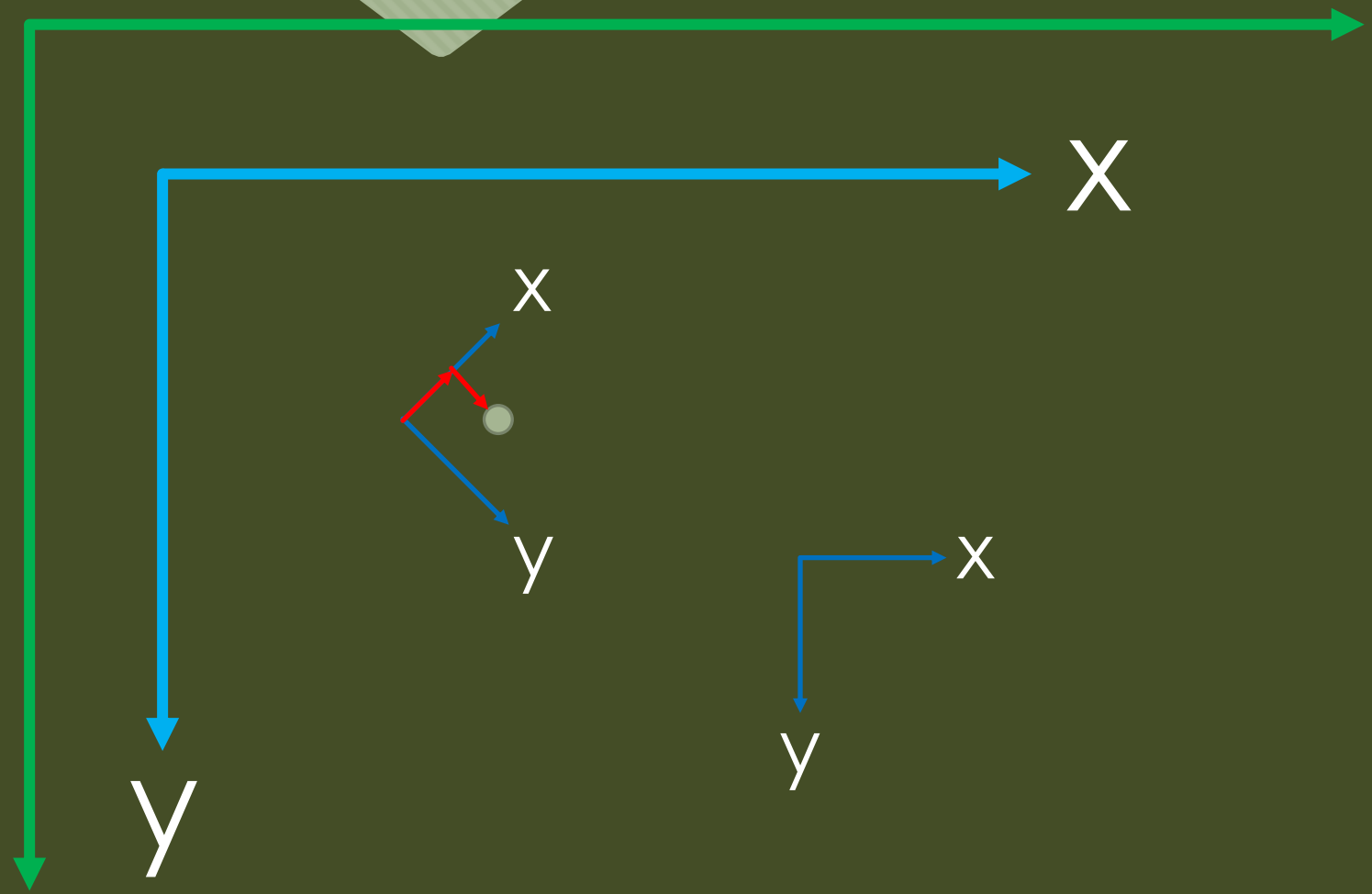
Space Hierarchy



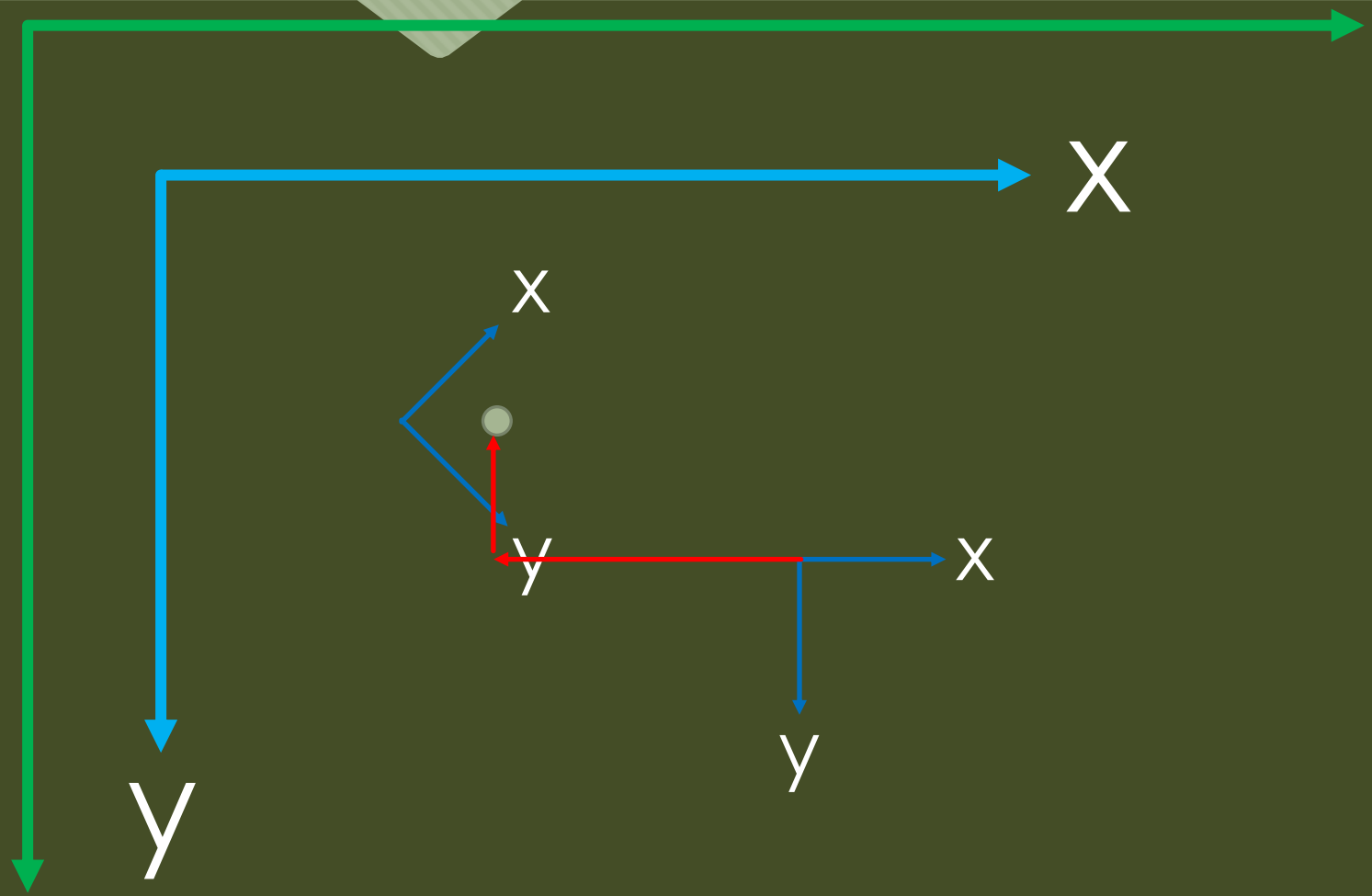
Space Hierarchy



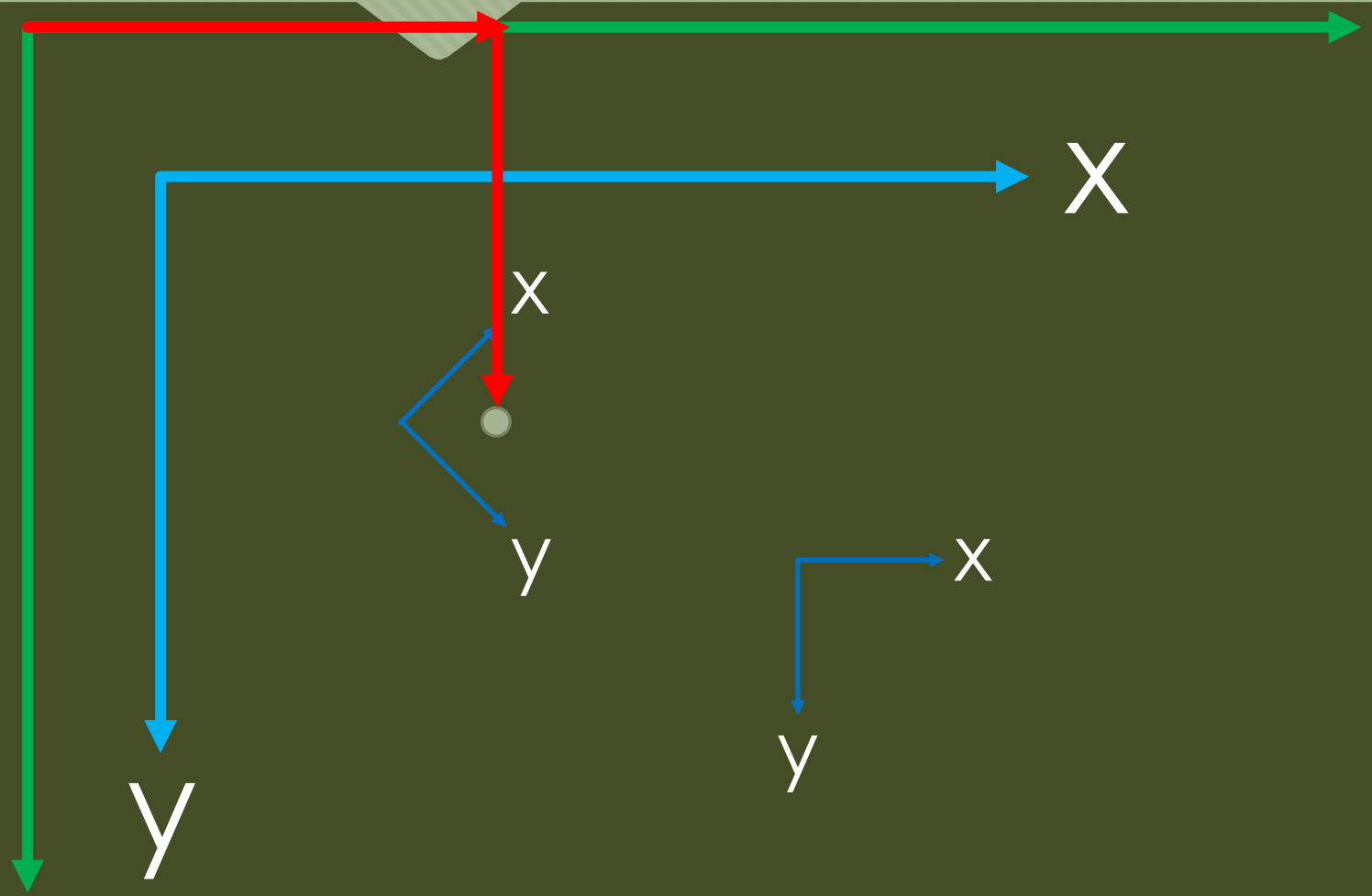
Space Hierarchy



Space Hierarchy



Space Hierarchy



‘Transform’ is local coord space

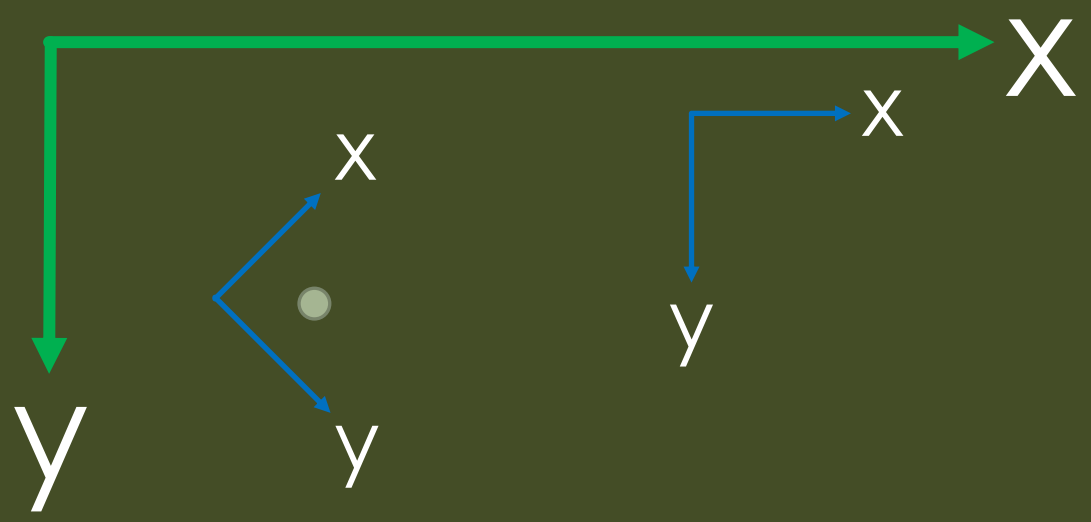
- node.Transform.origin
- node.Transform.x (“forward vector”)
- node.Transform.y

Mapping points between spaces

- local to parent
  - child.Transform.Xform(point)
- parent to local
  - child.Transform.Inverse().Xform(point) or
  - child.Transform.XformInv(point)
- local to global
  - child.GlobalTransform.Xform(point)
- global to local
  - child.GlobalTransform.Inverse().Xform(point) or
  - child.GlobalTransform.XformInv(point)

Mapping points between spaces

- sibling to sibling
  - var asGlobal = sibling1.GlobalTransform.Xform(point)
  - sibling2.GlobalTransform.XformInv(asGlobal)



Manipulating Node's Transform info

- cannot modify Transform variable directly
- set position
  - node.Position = new Vector(x,y);
- set rotation
  - node.Rotation = 3 // in radians
- move "forward" (along +x)
  - node.Transform.x.Normalized() * numPixels;
- move "backward" (along –x)
  - node.Transform.x.Normalized() * -1 * numPixels;
- move along any vector
  - node.Position += theVector;

Summary/Reference

- node.Transform
  - node.Transform.origin
  - node.Transform.x
  - node.Transform.y
- local to parent
  - child.Transform.Xform(point)
- parent to local
  - child.Transform.XformInv(point)
- local to global
  - node.GlobalTransform.Xform(point)
- global to local
  - node.GlobalTransform.XformInv(point)
- sibling to sibling
  - var asGlobal = sibling1.GlobalTransform.Xform(point)
  - sibling2.GlobalTransform.XformInv(asGlobal)

- cannot modify Transform variable directly
- set position
  - node.Position = new Vector(x,y);
- set rotation
  - node.Rotation = 3 // in radians
- move “forward” (along +x)
  - node.Transform.x.Normalized() * numPixels;
- move “backward” (along –x)
  - node.Transform.x.Normalized() * -1 * numPixels;
- move along any vector
  - node.Position += theVector;