<https://www.youtube.com/watch?v=22VYNOtrcgM&t=328s>

Legit Notes On 2D Nodes:

* Node2D: position, rotation, scale… All 2D nodes inherit from Node2D
* Camera2D: a VCAM with camera drag/smoothing options. Can rotate and change position
* Sprite2D/AnimatedSprite2D render a static image or one that is animated.
* CollisionObject2D base class for Collision, enable on/off
  + PhysicsBody2D base class for objects affected by physics
    - CollisionShape2D/CollisionPolygon2D = blue shapes to make hitboxes
    - StaticBody2D = object that cannot be moved (WALLS/Solid Objects)
    - AnimatedBody2D =Static body that can be moved with animation
    - Rigidbody2D = affected by physics
    - CharacterBody2D = made for player to control
  + Joint2D base class for 2D physics joins
    - DampenSpringJoin = connects two bodys with a spring
    - GrooveJoin2D = connects like piston
    - Pinjoin2D = like a bolt
* Area2D = detection field and zone where you apply different physics
* Audiolistener2D = listening point
* AudioStreamPlayer2D
* GPUParticle2D = make particles! Go play!
* TileMap = paint 2D tilemaps to make your levels!
* CanvasModulateNode = used to darken your scene for OTHER NODES to lighten it
  + Light2D base class for color
    - PointLight2D = lighting is done via texture
    - DirectionalLight2D = casts light like a sun, results may vary based on objects
    - LightOcculuder2D = set a shape to case a shadow
* Line2D = make lines
* Marker2D = debug node to just have something there for dev
* MeshInstance2D = 3D model in 2D
* MultiMeshInstance2D = use a lot of meshes to render
* NavigationRegion2D = pathfinding
* ParallaxBackground
  + ParallaxLayer =Set up for parallax bg, any texture will do
* Path2D
  + PathFollow2D = set up path for objects to travel on like moving platforms
* Polygon2D = make shapes
* RayCast2D = a line that detects if it something
* ShapeCast2D = similar to Raycast but with shapes like circles triangles etc
* RemoteTransform2D = ???
* Skeleton2D, PhysicalBone2D, Bone2D = a lot of work….Read doc
* VisibleOnScreenNotifier2D/VisibleOnScreenEnabler2D = detect if visual is on screen or not.
  + Notified sends signal when a thing is on screen
  + Enabler enables itself when it visible
* CanvaseGroup = group nodes
* BackBufferCopy = ???