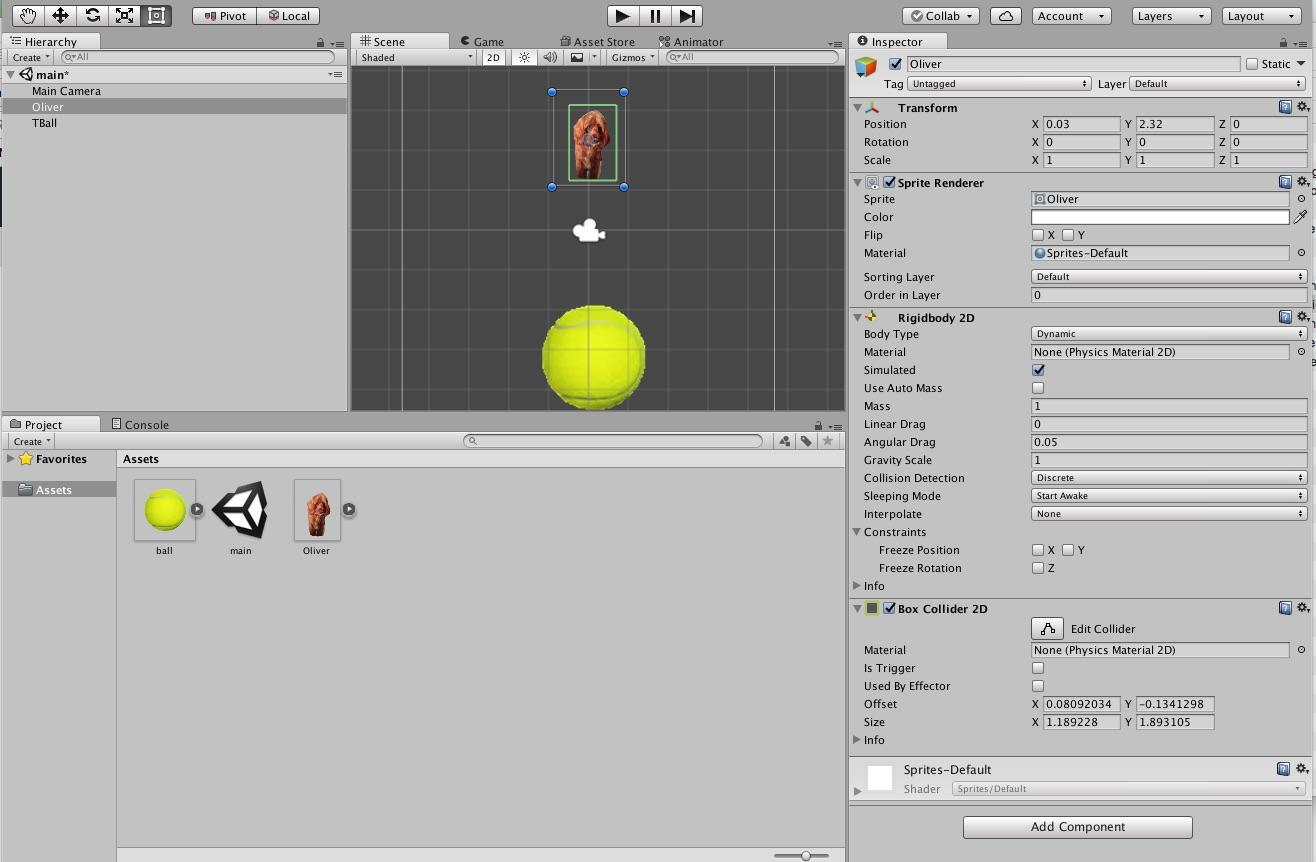
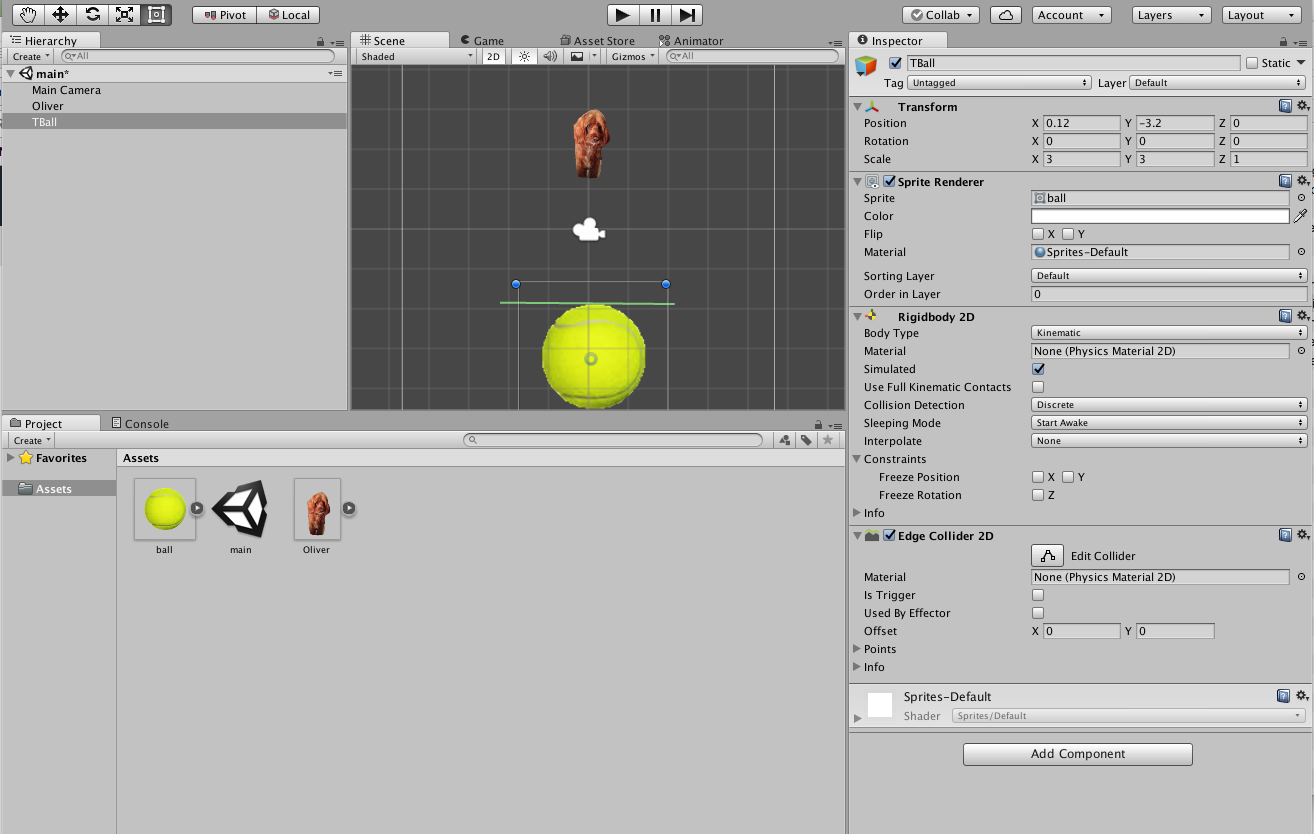
**Topic: Fun with Physics**

**Overview:** Although Genderless in Germany, in theory, doesn’t require the Physics Engine, but it is worth playing with physics a bit to understand how the Unity Game Engine will interact with the Tutoring Engine. Physics requires that an object in the hierarchy have a Rigid Body component. In order to interact with other objects it should also have a Collider. If we find a need for collision detection, then the Rigid Body 2D class will come into play. 2D projects have rigid body and collider classes that differ from 3D; they are more efficient. The most important point for 2D physics is that all collisions take place on the plane where z = 0.

**Your Turn (Please do the following):**

1. Open the project OliverLovesTennisBalls. Select the scene OliverLands. Try running it. Click on the Oliver object in the Hierarchy window. Check out Oliver in the Inspector. Oliver has two new components: Rigid Body 2D and Box Collider 2D. In the Scene window with the image box clicked, the Box Collider is marked in green. Notice that Oliver is a *dynamic* rigid body!



1. Click on the TBall object. Notice that it is a Kinematic Rigid Body – it doesn’t actually do physics, but physics objects (dynamic) can interact with it. Notice that it has an Edge Collider 2D. 
2. Your turn: Play with physics:
   1. Change the mass of Oliver.
   2. Change the TBall to a dynamic object. Then freeze its constraints. Change its mass etc.
   3. Change the TBall Collider to a sphere.
   4. Move Oliver to one side a bit.
   5. Put a ‘ground’ object below the ball that is Kinematic so that the ball can only roll sideways.
   6. Add more balls to the picture (how can you duplicate an object?)
   7. Make gravity move ‘up’ rather than down – you do this on the object.
   8. Make gravity move ‘sideways’ along the x axis. (Edit project settings, 2D).
3. Of course you want stuff to happen other than physics when things collide. This requires code. Switch the scene to BallDestroyed. Notice that the ball is dynamic. Run the project. Make sure you stop the run, don’t just pause it. Now load the script OliverCollision into this scene, and add the script as a component of Oliver. Run it again, then study the script. (Destroying objects is easy from code, creating new objects is a bit harder and is left for later.)

**Exploring More in Unity:**

The [Unity physics tutorials](https://unity3d.com/learn/tutorials/topics/physics) are pretty good. You could spend weeks elaborating on them. Please try to stick to the 2D Physics Tutorials and Assignments.

**Object Oriented Wrap Up:**  The theme emerging in these tutorials is that objects are everything. Objects like MonoBehaviors can be extended so that your script *inherits* all the existing data and methods of the parent. Objects are also connected through has-a links that allow objects to interact through code and data sharing. These ideas will be cricital to understanding how user interfaces (UIs) interact with objects. Consequently, game play in Genderless In Germany will require UI objects to make decisions about what visuals should appear, and what other UI objects should be created or destroyed. Of particular interest later will be objects that don’t necessarily appear on the screen such as the tutoring engine.