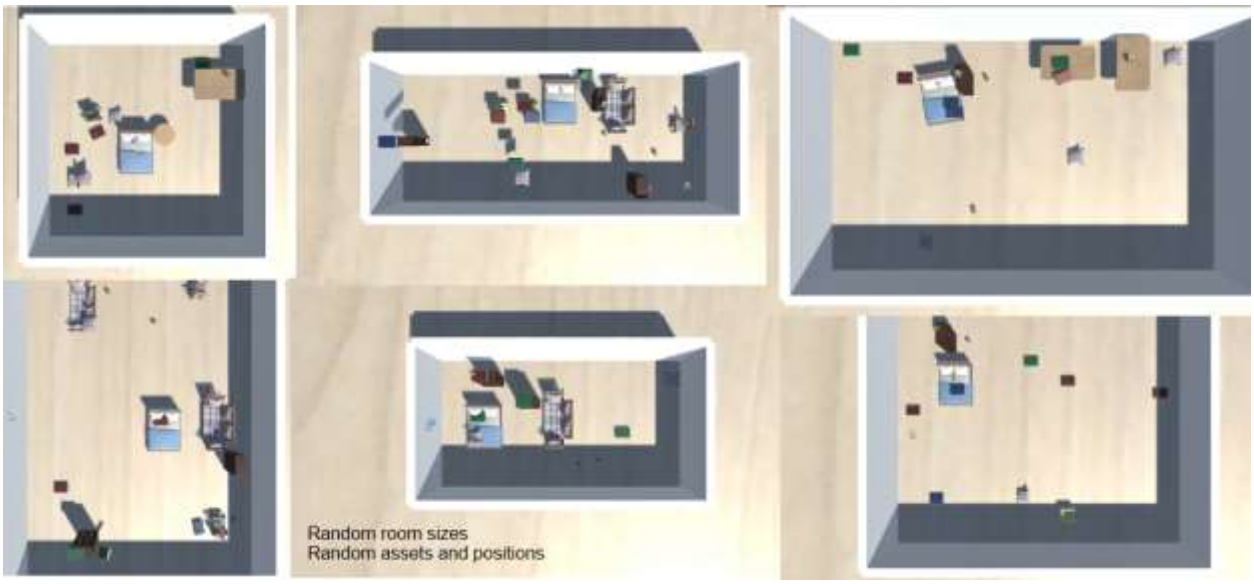
Felicity Huang
Project: Room Generation

I implemented generating randomized room sizes and spawned random furniture/props at random positions and rotations in the room in Unity (RoomGeneration scene). I also implemented a simple room plan generation algorithm in RoomPlanGeneration scene). I came up with my own algorithms for both parts.

The rooms are rectangles, and the sizes are generated under constraints of minimum and maximum lengths. I spawned objects by adding prefabs to different folders of different types (large furniture, small items) and loading them into a list by script then randomly choosing one and spawning it at a random location and 90 degrees rotation. The furniture are spawned on the ground based off the bounds of the room, and items on top of furniture have position randomized within the bounds of the size of the furniture below it. Originally, I had another type of object called "stackable" which are objects with flat tops and bottoms like books or boxes that can be stacked and wrote code for generating random number of stackable objects stacking on top of each other to simulate more realistic scenes, but commented it out to focus on the core part of room generation.

I added rigid body to the small items as I wanted them to interact with the other objects more dynamically to create a more realistic scene (like a messy room with scattered items).
I coded the collision detection between furniture by checking intersection of two squares using their x and z positions. For each new object to be spawned, I use its collider to get its width and

length and use its position to find the 4 corner points, then compare the points to the left-most, right-most, up-most, down-most values of every furniture that has already been instantiated.



The hardest part of this was getting the objects to spawn in the scene correctly. A lot of the prefabs had orientation or sizing or setup problems that I had to debug. I manually added box colliders to each prefab so that I could get its actual size properties (width, length, height) to instantiate it at the right position. I used prefabs from 2 different unity packages so one set had the origin of its prefabs at center of geometry and the other had it at the base which I also had to account for. I struggled a lot with this simple step because the objects kept floating in the air or intersecting with the walls and each other.

A separate component I implemented was room plan generation. I randomly generate the floor size to the house and then generate walls to separate the rooms in the building. For future extensions, the two components could be integrated.

Note: The video demo was recorded before I added collision detection and tweaked my floor plan algorithm to work better. Here are some better screen shots of the final implementation.

Room Generation