Siyuan Yu

Prof. Murphey

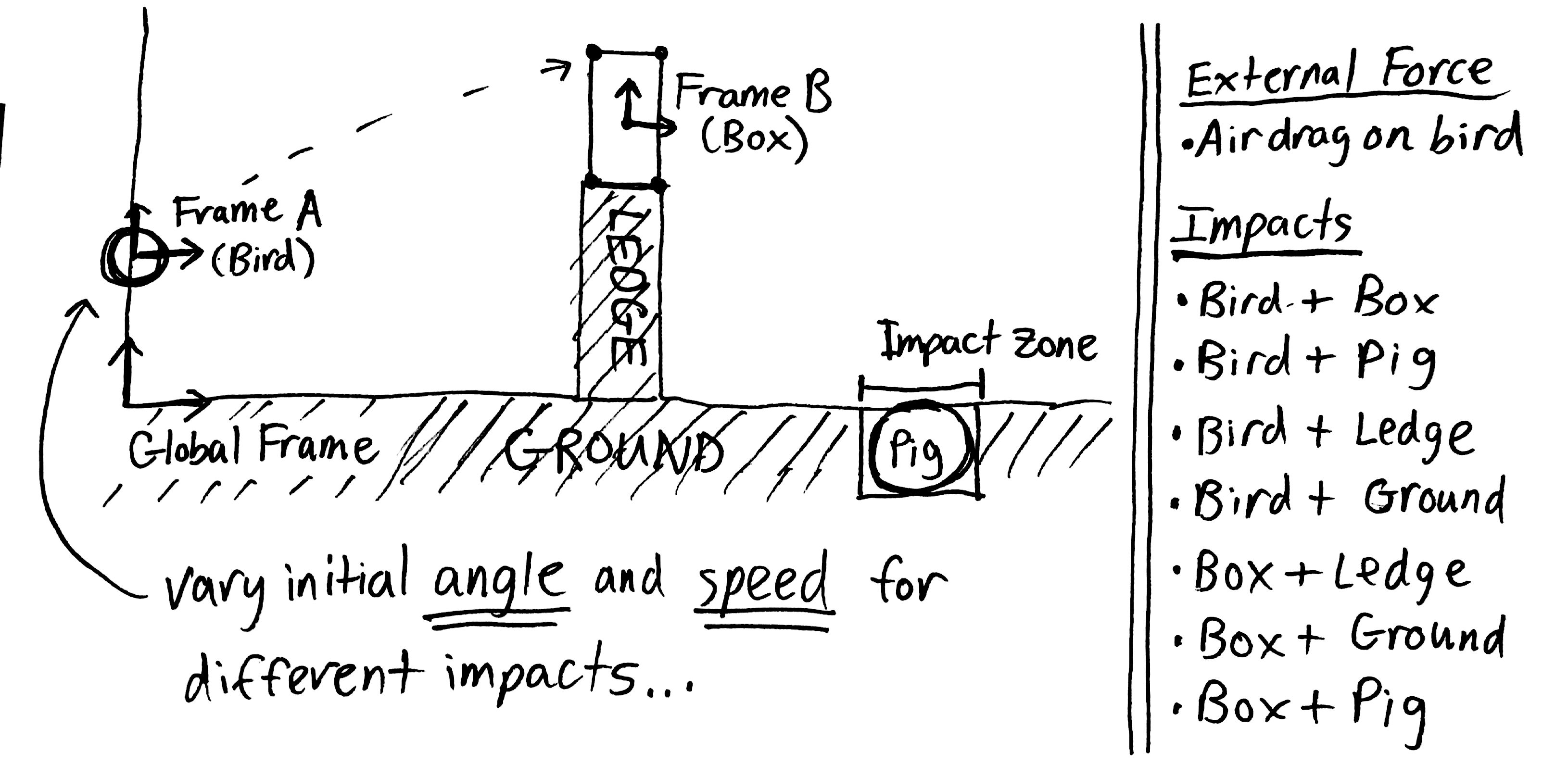
ME-314

**ME-314 Final Project: Angry Birds Simulation**

[**https://www.youtube.com/watch?v=rxypGSAkQgo**](https://www.youtube.com/watch?v=rxypGSAkQgo)

The goal of this project was to develop a playable simulation of angry birds using the dynamics concepts that we learned this quarter, with a heavy emphasis on impact and collision detection. The original concept involved launching a bird at a box sitting on flat ground, but there were too many issues with the box flipping and clipping through the ground. Therefore, the final configuration had the box on a ledge, which helped reduce the number of initial, close-to-ground impacts that caused the clipping problem.

The final simulation consisted of two main bodies (the bird and the box), a ledge, the ground, and a target zone for the pig. The bird was modeled as an object with mass and no rotational inertia and was subjected to air drag. The bird’s mass was set at 2.5 kg with an over exaggerated diameter of 0.75 meters in order to make impacts easier to see. The box was more accurately modeled and a rotational inertia term was calculated based off of its shape and density. The box had dimensions of 3x1x0.5 meters with a mass of 450 kg. The rotational inertia about the z-axis was calculated to be 112.5 kg-m^2. The box sits on a ledge that is 6.5 meters high at a distance of 30 meters away. All impacts were modeled as elastic impacts. The target zone with the helmet pig was designed to end the simulation upon detecting impact to represent the “winning” condition. See the provided video for reference.



The initial speed and angle of the bird was designed to be adjustable in the simulation to create different trajectories and cause different impacts to occur (see video). The most difficult part of this specific project was coding impact conditions and rules for all impact cases in order to allow for playability. For most angle and speed configurations, the resulting impacts worked as intended. However, when the bird or box impacted directly with a corner of the ledge, the discontinuous impact detection equations would occasionally fail to catch an impact and the object would clip through the ledge/ground. More cases need to be tested to find which specific impact detection equations fail, but this bug was most likely caused by boundary conditions between detection zones that were accidently left unaccounted for (there were too many conditions to debug completely).

The box presented a lot of modeling issues in and of itself. In order to simplify the box model, only the four corners of the box were used for impact detection (the bird and box collisions were custom impact cases). This meant that the edges of the box only existed visually, and could not impact with the rest of the environment. This caused many problems between the ledge corners and the box. As seen in case 6 in the video, an edge of the box passed straight through the upper right corner of the ledge because one corner of the box was still on the ledge while the other corner of the box had already fallen off of the ledge. In addition, this corner-centric model of the box made it necessary to create hard-coded impact conditions for the bird-box impacts that only triggered when the box was sitting at its initial location. With more time and effort, the edges of the box can be modeled to work with impacts more robustly by defining discontinuous collision detection equations for the shape of the box in the box frame and transforming the other objects into the box frame to evaluate collisions. According to online resources, vectors can be used to better model edges of objects as well.

The lack of friction in this model also caused the impacts and bounces to appear unrealistic at times in the simulation. Without a model for friction, the box would bounce indefinitely after impacts since energy is conserved. The box could never tip over on one corner the way it normally would in certain real world impact cases as well. Given more time, frictional affects can be modeled by introducing frictional forcing terms that depend on the normal force at each impact into the right hand side of the Euler Lagrangian equations. In order to model the friction force as an impulse, an extremely short solve/NDSolve loop can be triggered after each impact to momentarily include the frictional forcing term. This should theoretically make the simulations look more realistic for impacts.

Finally, some limitations occurred with Mathematica as well. For the event locator function, events with more than three combined Boolean conditions simply did not evaluate or trigger. This problem made it necessary to include additional if statements to check more complicated impact cases with multiple conditions for impact. Another limitation was the step size and adaptability of the NDSolve function. For example, in certain cases where the box began spinning very quickly while impacting the ground, the step size for NDSolve would be too big to detect collisions in quick succession. In addition, NDSolve had problems detecting multiple impacts that occurred at near the same time.

Overall, this project helped me realize the strengths and weaknesses of the techniques and tools that we used this quarter for modeling dynamic systems. Coordinate transformations, EL equations, constraint equations, and impact equations are immensely powerful when used together to model dynamics. However, it is difficult to use them efficiently and correctly and a lot of intuition and practice seems necessary to model things realistically. I really enjoyed learning and applying these new concepts this quarter and this class has really opened my mind up to a whole new way of looking at dynamic systems. Thank you for taking the time to teach this class and I hope you have a great winter break!