Bugs:

* For some of the curves, the correct tangent/normal/binormal data pulled from the global data model seemed to be slightly off (i.e. tangent and normal not perpendicular)
* As a temporary solution, the entire curve menu wall is currently moved with the user when they choose to go to/from the exercise room to avoid creating a new state machine. (When trying to just move the menu, it would render behind the other wall). Ideally the menu is duplicated in both rooms and does not move, assuming the exercises remain in the same scene as the main display room. However, if the user transports, the wall will shift with the user.
* If the user transports in the exercise room and uses the exit button to return to the main room, the table curve display is shifted when they return.
* The DrawTangentNormal.cs script only works if there is 1 instance of the TangentNormal prefab in the scene. If there are more, the answer evaluation is wrong because it is comparing all the tangent/normal spheres in the scene with the correct answers. This should be fixed to compare only the spheres tied to that particular instance.

Miscellaneous Notes:

* View is not yet set up for tangent normal exercises, so selecting that type of exercise from the wall activates gameobjects it should not
* The retry button is intended to start the current exercise over from the beginning, so it is only visible at the results panel. The reset button is intended to undo user choices in the current sub exercise (deselect a pillar for SelectThree or reset height adjustments and sphere positions for TangentNormal).
* Image on central display screen in cockpit is generated from img folder path and curve name, so it does not come up with an image for curves that are not in that folder and named with their curve name
* The bottom portion of the circle pad on the right controller toggles on and off the time vs distance/velocity plots while in the cockpit scene (set in GraphDisplayControl.cs)
* The grip button on the right controller currently shows the solution for the tangent normal exercise. It also resets any height adjustment the user made. The results are currently printed to the console. (Set in DrawTangentNormal.cs)
* Both directions of the tangent and normal vectors are accepted as correct in the tangent normal exercise, but the solution lines show only the correct direction
* Linecoords.txt
  + Used to carry curve data from SingleScene to CockpitTravel scene
  + Written to on CockpitButton press, cleared on application quit
  + Format:
    - Name of current curve
    - 2 or 3 (for dimension)
    - Total points in curve
    - For each point (scaled)
      * x y z
      * tangentX tangentY tangentZ
      * normalX normalY normalZ
      * (if 3d: binormalX binormalY binormalZ)
      * timeDistanceX timeDistanceY
      * timeVelocityX timeVelocityY
* Exercisedata.txt
  + Used to store exercise run results and chosen answers for every exercise attempt
  + Only written to if user reaches results display panel for current exercise (will not record incomplete run data), cleared on new application load
  + Format:
    - Exercise name
    - Attempt number
    - previousAnswer chosenAnswer correctAnswer
    - Overall score (X/X)
* In general: red for tangent, green for normal, blue for binormal
* Exercise structures:
  + Select Three Exercise
    - Left Curve, Middle Curve, Right Curve
    - PillarIndex (1, 2, 3) = user choice
  + Tangent Normal Exercise
    - Curve
      * X number highlight points along indices of curve
        + For each highlight point: [tangent x, tangent y] and [normal x, normal y]

In DrawTangentNormal.cs:

* compareVectors() is currently hard coded for the first highlight point in the 3rd selection exercise since that is all there is in the initFile. Once the tangent normal exercise is integrated into the exercise framework, this should be generalized to access the current exercise and answers based on indices from the GlobalDataModel.
* When compareVectors() is called, the current tangent and normal positions should be assigned to the ChosenAnswer for each highlight point in the current sub exercise. If they are not drawn (check the tangentDrawn and normalDrawn booleans), write -1. Alternatively, move those booleans to the TangentNormalExerciseAnswer class and use those in return value for IsValid()
* The answer evaluation in compareVectors() should likely be moved to a function in the TangentNormalExerciseAnswer class. Compare user answers to what is stored in CorrectAnswers for all highlight points in the current sub exercise. This function can be called for each chosen answer when evaluating overall results.

In TangentNormalExercise.cs:

* GetValues() and SetValues() are both set up for only xy values for tangent and normal. Can be extended to allow for 3D positions.
* Initializing values in constructor, GetValues(), and SetValues() have not been thoroughly tested. There may be errors in setting all dimensions of the answers correctly.

In CurveSelectionControl.cs:

* To get state change to exercises working properly, I had to move the state change outside the if statement (Line 178). Appears that the if statements are there to only change state on initial load, but it seems they can be updated to change on any call to that function (like from ExerciseButtonBehavior or ExitButtonBehavior)

In ReturnToRoomButtonBehavior and ExitButtonBehavior.cs:

* If exercises are moved to another scene, ReturnToRoomButtonBehavior script can be used in place of ExitButtonBehavior

In CockpitButtonBehavior.cs:

* A boolean for ShowFlyButton or something similar should be added and used to set the button as active in Start() if cockpit mode should be enabled (similar to the quiz button)
* Coordinates written in WriteCoordsData are currently scaled using specific values and offsets. This could be changed to scale depending on curve size, or the curve can be mapped to the size of the boundaries in the cockpit scene
* Coroutine can likely be improved

In CockpitTravel.cs:

* The mapping from regulator position to speed can likely be improved or simplified
* The compass pin is currently facing forward statically. Ideally it moves to show direction of travel, maybe facing direction of tangent?