Roles and Responsibilities:

Control Deck is a solo project, developed and implemented by Holden Denyer.

- ***Installation design and Concepts:***
  Installation design will consist of several main components, each with their own elements to be designed and produced:
  - Deep-Sea Submarine Game
    - Unity Development
      Program basic game mechanics around arduino input mechanisms in C#. Establish 2D parallax movement style. Handle player failures and successes with event handling.
    - Art and Asset Creation
      Develop game art and animations using ProCreate and Photoshop softwares.
  - Control Deck Controller
    - Control Board Chassis
      Design a box-type unit with edges of input placements included. To be lasercut with a plastic, textured material to mimic submarine interior. Prototyping to be done with 3D printing and wood.
    - Arduino Controller Wiring
      Layout input handling and connections to Arduino Mega controller. Ensure wiring can fit within the chassis.
    - Input Layout
      Determining the most visually appealing and interactively enticing arrangement of inputs. Update chassis and arduino wiring accordingly.

  I will update these elements and the design values according to presented limitations, improvements or general project changes.

- ***Programming:***
  Programming will be an essential part of both the game and controller elements of *Control Deck*:
  - Unity Programming
    Utilizing C# to define game mechanics and handle events in the game scene. Taking Arduino inputs and mapping them to in-game controls and console commands.
  - Arduino IDE Programming

Mapping and handling both the binary and analog style inputs on the controller. Defining update rates and sensitivities to match the game's mechanics and play style.

- ***Art Direction :***
  Control Deck's game art will take on a simple, doodle-esque style to keep the experience feeling imaginative and fun. Art will take on the following forms:
  - Sprite/Asset Design
    All static elements of the game will be created in ProCreate and exported as PNGs. These can then be dropped into the Unity scene, saved as prefabs, and assigned to script references for events and environment handling.
  - Animation Development
    Animated game elements will be created by drawing sequences in ProCreate, then defining frames and animation transitions in the Unity scene.

- ***Sound Design:***
  Audio queues will be essential to establishing the participants' immersion. Audio will take on the form of sound effects reacting to in-game events, as well as some background sound effects and music to fill empty space.
  - Sound Effects
    Sound effects will be recorded in foley style or located online with resources like freesound.org. These effects will then be linked to scripts in the Unity project which are fired given specific criteria.
  - Background Audio
    Ambience audio will also be included to immerse the participants in the underwater environment. Longform audio files like the rumble of the ocean's currents will be handled in the background, queued by scripts given certain criteria or events.

- ***Art Implementation: Animation \ Video\ live action,or any other source material manufacturing:***
  As defined by the above responses, art implementation is separated into various aspects under the major categories:
  - Installation Display
    - Deep-Sea Submarine Game
      - Unity Development
        Program basic game mechanics around arduino input mechanisms in C#.

        - Art and Asset Creation
          Develop game art and animations using ProCreate and Photoshop softwares.
    - Control Deck Controller
        - Control Board Chassis
          Lasercut box-type design to fit all input types.
        - Input Layout
          Determine the most visually appealing arrangement.
- Visual Assets
    - Sprite/Asset Design
      All will be created in ProCreate and exported as PNGs to be saved as prefabs and assigned to script references.
    - Animation Development
      Created by drawing sequences in ProCreate, then defining frames and animation transitions in the Unity scene.
- SFX
    - Sound Effects
      Sound effects will be recorded in foley style or located online with resources like freesound.org.
    - Background Audio
      Longform audio files queued by scripts.

- ***Direction of Prototyping and Testing***:
  Prototyping will be separated into many stages to test individual and collective components iteratively:

  Stage 1: Early Mechanics Feedback
    - Goal: Create a basic array of several button and switch inputs, labeled and sectioned on a prototype board.
    - Resources: Basic controller inputs and rudimentary layout
    - Test: Ask users to provide feedback on the types of inputs. Does the array feel dynamic and interesting? Is it too overwhelming? Provide a demo where input labels are called and must be interacted in a time limit. How does this mechanic feel?

  Stage 2: Early Game Development Feedback
    - Goal: Develop the basic structure of the game, using specific inputs in a time limit to pass obstacles.
    - Resources: Game demo, keyboard and display.

- Test: Ask users to play the game. The game will call for specific key inputs or combos in a given time limit. Ask the user if the game is too challenging or too simple. Would this be more or less fun in a cooperative context?

Stage 3: Large-Scale Controller Testing

- Goal: Provide a near-complete controller deck as described by the initial concept.
- Resources: Controller deck with complex input layout
- Test: Call labeled inputs and ask the user to interact with them given a time limit. Does the time limit feel too long or short? Is the variety of inputs immersive? Is it overwhelming?

Stage 4: Full Game Testing

- Goal: Fully developed game with set difficulty curve and ending
- Resources: Full game, keyboard, display
- Test: Ask the user to play the game a few times, modified for keyboard inputs. Note any critiques or feedback.

Stage 5: Collective Game and Controller Testing

- Goal: Provide a nearly complete experience with controller deck and game
- Resources: Controller, game
- Test: Ask the user to play. Can they figure it out? Is it cohesive and fun? Note any critiques or feedback.