PROJECT PROPOSAL

- **Who is on the team? If you are working in a team of two, is there a clear division of labor? If the team is smaller/larger than recommended, provide a rationale.**

➔ Deep Kotadiya - NUID 002931454
➔ Abinav Anantharaman - NUID 002774223
➔ Nitin Vinodbhai Thakkar - NUID: 002766843
Deep Kotadiya - Environment and Learning Algorithm:
  - ◆ Scope: Deep is responsible for creating the interactive and dynamic escape room environment that poses challenges to the agent. Additionally, he will develop and fine-tune the learning algorithm that enables the agent to learn and adapt within this environment.
  - ◆ Work: This involves designing the layout, obstacles, and teleportation mechanics of the environment, as well as selecting and implementing the appropriate reinforcement learning algorithm for the agent.

➔ **Abinav Anantharaman** - Learning Algorithm and Evaluation Policy:
  - ◆ Scope: Abinav focuses on the development and optimization of the learning algorithm alongside Deep, ensuring that it is efficient and effective. He is also in charge of devising the evaluation policy to rigorously assess the agent's performance and learning progression.
  - ◆ Work: His tasks include collaborating on the algorithm's design and adjustments, plus developing a comprehensive set of metrics and procedures to evaluate how well the agent learns and adapts to the environment.

➔ **Nitin Vinodbhai Thakkar** - Environment and Evaluation Policy:
  - ◆ Scope: Nitin works on constructing and refining the escape room environment, ensuring it is conducive to learning and testing the agent. He also collaborates with Abinav to establish and implement the evaluation policy, assessing the agent's interactions and progress.
  - ◆ Work: His responsibilities include detailing the environment's physical and interactive elements and working on the framework and criteria for evaluating the agent's decision-making, learning efficiency, and adaptability.

- **Describe the problem you are trying to address, and provide a formulation of it. If it involves an algorithm (i.e., any implementation-based project), describe the input to the algorithm and the desired output.**

➔ The project involves creating a simulated escape room environment and developing a reinforcement learning algorithm to enable an agent to navigate and solve the room's challenges efficiently.

➔ **Environment Setup:**

- ◆ The environment is a digitally simulated escape room with interactive elements like keys, clues, traps, and teleportation points. The agent needs to interact with these elements to find the exit or solve a puzzle, with the environment designed to test various navigation and problem-solving strategies.

➔ **Algorithm Formulation:**

- ◆ The reinforcement learning algorithm processes the environment's state, which includes the agent's location and nearby elements. It learns from the agent's experiences, guided by a reward system that reflects the consequences of its actions. The algorithm iteratively refines the agent's policy based on its successes and failures, aiming to maximize the cumulative reward.

➔ **Output:**

- ◆ The output is a series of actions the agent takes to achieve its goal, demonstrating its ability to learn and adapt to the environment. The resulting policy is an optimized strategy for navigating the escape room, showcasing the agent's improved decision-making and problem-solving capabilities over time.

➔ **Takeaway:**

- ◆ The project underscores the potential of reinforcement learning in robotics, demonstrating how autonomous agents can navigate and solve complex, real-world problems. By applying these techniques, robots can adapt to dynamic environments, make informed decisions, and overcome unforeseen challenges, paving the way for innovative solutions in fields like search and rescue, healthcare, and autonomous navigation.
- ◆ In the context of urban disaster response, the escape room RL framework trains robots to adeptly navigate compromised buildings, identify key resources or survivors, and avoid hazards, mimicking the unpredictable nature of real disaster sites. The training algorithm enhances the robot's ability to learn and adapt on-the-fly, significantly boosting its efficiency and decision-making prowess in critical, life-saving search and rescue missions.

- **What is the ideal outcome of the project? What do you expect to show?**
  We plan to implement and test the following algorithms to benchmark their performance in the Escape Room environment:

  ➔ The ideal outcome of the project is to demonstrate the effective application and performance of various reinforcement learning algorithms within a custom-designed escape room environment. Specifically, you aim to show how each algorithm navigates the challenges and puzzles presented, highlighting their learning efficiency, adaptability, and problem-solving capabilities in a dynamic and complex setting. By implementing and testing multiple algorithms, you expect to:

- ➔ **Demonstrate Learning Efficiency:** Show how quickly and effectively each algorithm can learn to solve the escape room, evaluating aspects like the number of episodes required to reach a certain level of performance.
- ➔ **Showcase Adaptability:** Illustrate the algorithms' ability to adapt to changes or new challenges within the environment, demonstrating their flexibility and robustness.
- ➔ **Compare Problem-Solving Strategies:** Highlight the differences in how each algorithm approaches problem-solving within the escape room, providing insights into their decision-making processes.
- ➔ **Benchmark Performance:** Compare the performance of these algorithms against each other and possibly against a baseline or standard environment, showcasing their relative strengths and weaknesses.
- ➔ The algorithms you plan to test could include, but are not limited to, Deep Q-Network (DQN), Proximal Policy Optimization (PPO), and Asynchronous Advantage Actor-Critic (A3C). The expected outcome is a comprehensive evaluation of these algorithms, providing valuable insights into their applicability and effectiveness in complex, dynamic environments akin to the escape room scenario.

- **What algorithms do you expect to use?**

(Planning to implement/ test any 2 of these algorithms that we might find best fit for our use case)

- ➔ Deep Q-Network (DQN): Ideal for environments with high-dimensional observation spaces, DQN can handle complex input data and has been successful in various applications, making it suitable for the intricacies of an escape room.
- ➔ Proximal Policy Optimization (PPO): Given its balance between sample efficiency and ease of implementation, PPO is a strong candidate for environments requiring continuous or complex action spaces, potentially beneficial for nuanced interactions within the escape room.
- ➔ Asynchronous Advantage Actor-Critic (A3C): A3C's ability to learn different aspects of a problem in parallel using multiple actors could accelerate learning and offer robust policy development for the escape room's challenges.
- ➔ Soft Actor-Critic (SAC): For tasks requiring sophisticated exploration strategies and robust performance, SAC, an off-policy algorithm that maximizes entropy for more diverse action selection, could be effective.
- ➔ Twin Delayed Deep Deterministic Policy Gradient (TD3): If the escape room includes continuous action decisions, TD3's approach to reducing function approximation error could be crucial for precise, effective navigation and interaction.
- ➔ Monte Carlo Tree Search (MCTS): Although not a typical choice for RL, testing MCTS could provide insights into its utility in strategic decision-making within complex, uncertain environments.

- **What topics / libraries / platforms, if any, will you have to learn in order to undertake your project? Provide references where applicable.**

➔ Gym Api and rest as per requirement.

- **What domain(s) will you be working on? Is there a simulator available? If not, will you make a new one? If the latter, do you have the resources / data to do so?**

➔ We will be designing a custom class for our environment tailored specifically to the intricacies of an escape room scenario, incorporating unique challenges, puzzles, and traps. The extended goal of our project is to rigorously benchmark this custom environment against a well-established one, such as those found in OpenAI Gym. By conducting this comparative analysis, we aim to evaluate the effectiveness of our tailored reward system and modeling parameters, ensuring they provide meaningful and accurate learning contexts. This benchmarking will enable us to validate the performance and responsiveness of our agent within our environment, highlighting its strengths and areas for improvement in contrast with standard benchmarks.

- **Define milestones for your project.** These will be due on 03/15 and 04/15 – note that the second one is only a few days before your presentation. Milestones can include learning about certain topics / algorithms, acquiring and processing datasets, implementing an algorithm, analyzing results, etc. Again, we will not penalize you if you do not achieve your specific milestone. At each of the due dates, we will request a short progress report that addresses progress toward the milestones, and if they were not achieved, which turned out to be more challenging than expected. You may also find that your initial milestones were inappropriate, and you can choose different milestones, work toward those, and report on them. That is completely fine too – the point is to decompose your project into smaller pieces and ensure that you are making consistent progress over the next few weeks.

➔ **Milestone 1: Due 03/30**

➔ Literature Review and Algorithm Selection:
  - ◆ Complete a comprehensive review of existing literature on reinforcement learning, specifically focusing on algorithms applicable to dynamic and complex environments like escape rooms.
  - ◆ Select a set of reinforcement learning algorithms (e.g., DQN, PPO, A3C) for implementation based on the literature review findings, considering their potential effectiveness in our specific context.

➔ Environment Design and Development:
  - ◆ Define the specifications for the escape room environment, detailing its layout, interactive elements, and the dynamics of puzzles and obstacles.
  - ◆ Begin developing the custom class for the escape room environment, ensuring it can interface with the chosen RL algorithms.

➔ Learning and Preliminary Implementation:
  - ◆ Acquire any necessary knowledge or skills related to the chosen algorithms, simulation development, or any specific tools/libraries required for our project.

- ◆ Initiate the implementation of the basic framework or infrastructure needed to support our environment and algorithm testing.

➔ **Milestone 2: Due 04/15**

➔ Algorithm Implementation and Environment Integration:
- ◆ Complete the implementation of the selected reinforcement learning algorithms.
- ◆ Integrate these algorithms with our escape room environment, ensuring they can interact effectively and that the environment can provide appropriate feedback to guide learning.

➔ Preliminary Testing and Analysis:
- ◆ Conduct initial tests to assess the functionality and performance of the algorithms within our environment.
- ◆ Begin analyzing the results to identify any immediate issues or adjustments needed, focusing on the agent's learning progress, the effectiveness of the reward system, and overall system performance.

➔ Preparation for Presentation:
- ◆ Compile our findings, results, and any significant learnings or challenges encountered during the project.
- ◆ Prepare a concise yet informative progress report and presentation that summarizes our work, findings, and any adjustments or new directions taken based on our experiences and analysis.

- **Provide a week-by-week plan for your project.**

➔ **Week 1:**
- ◆ Research and select appropriate reinforcement learning algorithms.
- ◆ Initiate the design of the custom escape room environment, focusing on its core components like layout and interactive elements.

➔ **Week 2:**
- ◆ Complete the escape room environment development, including puzzles, traps, and layout.
- ◆ Start learning and integrating necessary libraries and frameworks for development and simulation, such as OpenAI Gym.

➔ **Week 3:**
- ◆ Begin implementing the selected reinforcement learning algorithms.
- ◆ Start preliminary training and testing of the agent within the environment.
- ◆ Milestone (by end of Week 3): Have a functional environment and initial algorithm implementations with basic training completed.

➔ **Week 4:**
- ◆ Analyze the early results from the training, identifying any areas for refinement or optimization in both the algorithms and the environment.
- ◆ Enhance the learning algorithms based on performance feedback and continue training the agent.
- ◆ Begin drafting the initial sections of the progress report.

➔ **Week 5:**

- Complete further training and final optimizations of the agent and algorithms, ensuring robust learning and problem-solving capabilities.
- Finalize and review the analysis of results, drawing conclusions and preparing insights for presentation.
- Complete the progress report and prepare the presentation materials.
- Milestone (by end of Week 5): Achieve thorough analysis and documentation of the project, ready for presentation and reporting.