Final Write Up/Reflection
Due: Week of May 2, 2025
Rubric: TAs will be grading this check-in based on this rubric.

- **Title**
  - DeepMimic Project
- **Who**
  - Team Name: C++ Undergrad Neural Training Team
  - Authored by: Effy Pelayo Tran (ttran59), Rainer Gardner-Olesen (rgardne8), Jason Uranta (juranta)
- **Introduction**
  - We are re-implementing DeepMimic (Peng et al. 2018) to enable a simulated humanoid to imitate motion-capture clips and adapt those skills for new tasks (e.g., steering, obstacle traversal, target strikes). DeepMimic's core innovation is combining a motion-imitation reward with task-specific rewards and two key training strategies, Reference State Initialization (RSI) and Early Termination (ET), to produce generalizable and natural character movement.
- **Methodology**
  - As Reinforcement Learning does not require a dataset in the same way as Supervised Learning does, we are only using two Motion Capture datasets for the added Example-Guided Imitation Learning portion of the Deep Reinforcement Learning paper we are implementing. We will use a total of ~20 clips (1-5 seconds each) from CMU and SFU mocap for the walk, run, backflip, and spin-kick tasks. We will need to preprocess these clips by parsing the BVH files and storing the rotation and position data for each joint into a large dictionary that maps to different tasks that map to different clips and map to different frames and then map to the different joints. We are training our model with the stable baselines-3 PPO that allows us to put in a Mujoco Env that we define and that should have a defined reset() step() and _get_obs() function. It will use these functions to define the state (joint position, velocities, center of mass) and action (applied joint torques) run PPO with clipped surrogate objective, GAE for the advantage and temporal difference TD for the value, and where the policy network and value network each have two fully connected layers (1024→ 512 units, ReLu).

- **Results**
  - We achieved our base goal of getting our humanoid to stand, walk and run in a user-given specified direction with pure RL. We also made great strides in achieving our target goal of incorporating natural movement using motion capture imitation reward. We rendered a rollout of the learned policies, walk and run. We succeeded in making the motions of the humanoid character imitate the feed in mocap data in order to make its movements look more human. We integrated the model's imitation abilities with task optimization techniques to allow the model to reach a goal. In the Target Heading Control Task, the humanoid moves in a direction that is within +-1 radian of the desired direction.

- **Challenges**
  - The hardest part of the project we have encountered so far was the parsing of the motion capture clips to fit our humanoid. Our humanoid has less joints than the motion capture skeleton and the Mujoco humanoid's center mass is its upper torso while the mocap center of mass is its hips. We had to modify our Mujoco's humanoid's joint hierarchy to match this, and use less of the mocap joints for the imitation reward. Long training times (20,000,000-40,000,000 steps) were also a challenge we had to deal with. Learning how to run the physics environment on GPU as the physics simulations are usually run on CPU.

- **Reflection**
    - How do you feel your project ultimately turned out? How did you do relative to your base/target/stretch goals?
        - Overall, we feel that our project was an impressive satisfying success. Through we did not implement the full paper, we accomplished an impressive part of it; of training our model how to walk and giving it life There is definitely room for improvement in terms of our implementation of our stretch goals, but we fleshed out our base and target goals very well and gained great clips of our model walking and running according to our specification and in a human-like way.
    - Did your model work out the way you expected it to?
        - Not entirely because while our model walks and runs relatively well we do struggle to get some parts of the body to look more human in motion such as the arms, spine, and ankles which look a bit crooker as the model walks.
    - How did your approach change over time? What kind of pivots did you make, if any?
        - We added more constraints to the model, the more we worked on our projects as we saw more issues with the kinematics. For example: its upper back was falling forward too much so we contrinastred the range of the upper back and increased rewards for upright behavior.
    - Would you have done differently if you could do your project over again?
        - We would adjust the model to match the size of the mocap so that way we could position information qs opposed to just rotation information.
    - What do you think you can further improve on if you had more time?
        - We could manipulate the physical environment more and add more obstacles for the model to interact with and we would also incorporate more actions that the humanoid could do such as striking and throwing. We would also introduce other types of characters such as the T-rex, dragon, and robot models.
    - What are your biggest takeaways from this project/what did you learn?
        - RL is time-consuming, takes a long time to train and code, involves a lot a trial and error, takes a lot of timesteps when compared to DL algorithms, and is hard to use in general. RL is also a very enriching thing to learn about, as it is learned from its own experience rather than a predefined loss function.

Project Check-in #2
Due: Week of April 14, 2025
Rubric: TAs will be grading this check-in based on this rubric.
For the second check in, there are two parts:

1. You will submit an outline that details your plan and the main ideas via email to your mentor TA or posted on your Github **before your meeting**. The outline requirements are described below. Additionally, submit the URL of your Github repo in this form.
    - Our Github Repo can be found here! https://github.com/effypelayotran/C_Undergrad_Neural_Trainers.git
    - We have our outline that details our plan and main ideas written below, *highlighted in yellow.*

2. You will meet with your mentor TA and review the work you have done since your project has been approved. Reach out to your mentor TA by 04/10 to schedule a meeting within the week of 04/14 - 04/18. Prior to this meeting, please try to have an idea of:
    - **Understanding**: have as thorough of an understanding as possible of the paper you're replicating/problem you're solving

○ **Data**: come with ideas on what data you'll need to use (and how you can access it) ○ **Methods**: have a rough idea of what kind of architecture you plan on implementing ○ **Metrics**: have a proposal for your base, target, and stretch goals.

Note:

1. **Base goal** = what you think you definitely can achieve by the final due date.
2. **Target goal** = what you think you should be able to achieve by the due date.
3. **Stretch goal** = what you want to do if you exceed your target goal.

Further note that these goals are flexible and can be re-evaluated at later checkpoints.)

The outline that you submit/write-up should contain the following:

- **Title**: DeepMimic. Using Motion Capture Clips to Guide Deep Reinforcement Learning that Teaches a Humanoid Model how to Move with Natural Gait.
- **Who**: Effy Pelayo Tran (ttran59), Rainer Gardner-Olesen (rgardne8), Jason Uranta, (juranta) ● **Introduction**: What problem are you trying to solve and why? **If you are implementing an existing paper, describe the paper's objectives and why you chose this paper:** We are re-implementing DeepMimic (Peng et al. 2018). The paper is creating a set-up that allows a humanoid model to move in physically realistic ways, by learning skills from motion capture clips and adapting those skills to new tasks (such as walking in new directions, over new obstacles, and striking new targets.) It does so by proposing a **character + task reward + imitation reward** framework, with 2 new key features that have not been popularized before in previous RL humanoid training set-ups; Early Termination and Reference State Initializations. ○ What kind of problem is this? Classification? Regression? Structured prediction? Reinforcement Learning? Unsupervised Learning? Etc.:
  - ■ This is a Reinforcement Learning + Imitation Learning problem.

- **Related Work**: Are you aware of any, or is there any prior work that you drew on to do your project? Please read and briefly summarize (no more than one paragraph) at least one paper/article/blog relevant to your topic beyond the paper you are re-implementing/novel idea you are researching. In this section, also include URLs to any public implementations you find of the paper you're trying to implement. Please keep this as a "living list"–if you stumble across a new implementation later down the line, add it to this list.
  - ○ We initially came across this NVIDIA paper of AI-Driven Physics Based Character animation. (demo: https://www.youtube.com/watch?v=8oIQy6fxfCA&ab_channel=NVIDIA) This drew our attention because animating rigged characters to do realistic movements in non-physics animation tools such as Blender or Maya is a super tedious task, and the combination of a physics environment + RL could be used as a tool for speeding up the animation process by simply giving your desired character a task in this physics environment. We then came across the previous fundamental RL/IL paper that this NVIDIA paper built off of call "DeepMimic: Example-Guided Deep Reinforcement Learning of Physics-Based Character Skills" (demo: https://www.youtube.com/watch?v=vppFvq2quQ0&ab_channel=JasonP, paper: https://xbpeng.github.io/projects/DeepMimic/DeepMimic_2018.pdf) We plan to reimplement DeepMimic, as it has the foundational RL/IL concepts that all the other paper in this area of interest build of off, such as PPO, Early Termination, use of only a few MoCap footage (does not require too much data!), and Retargeting. The rpo for DeepMimic is public, and we will be tracking it. However, the code base is huge and unintuitive, and written with C++ and SWIG that wraps their C++ code into Python. We want to re-implement a cleaner, simplified version of this C++ workflow in pure Python and JAX.
- **Data**: What data are you using (if any)? If you're using a standard dataset (e.g. MNIST), you can just mention that briefly. Otherwise, say something more about where your data come from (especially if there's anything interesting about how you will gather it). How big is it? Will you need

to do significant preprocessing?

- As Reinforcement Learning does not require on a dataset in the same way as Supervised Learning does, we are only using the follow two Motion Capture datasets for the added Example-Guided Imitation Learning portion of the Deep Reinforcement Learning paper we are implementing:
  - Carnegie Mellon University Mocap Database: https://mocap.cs.cmu.edu/
  - Simon Fraser University Mocap Database: https://mocap.cs.sfu.ca/
- We will use a total of ~20 clips (1-5 seconds each) from CMU and SFU mocap for the walk, run, backflip, and spin-kick tasks.
- We will need to preprocess these clips by parsing the BVH files and storing the rotation and position data for each joint into a large dictionary that maps to different tasks that map to different clips and map to different frames and then map to the different joints.

- **Methodology**: What is the architecture of your model?
  - How are you training the model?
    - We are training our model with the stable baselines-3 PPO that allows us to put in a Mujoco Env that we define and that should have a defines a reset() step() and _get_obs() function. It will use these functions to define the state (joint position, velocities, center of mass) and action (applied joint torques)run PPO with clipped surrogate objective, GAE for the advantage and temporal difference TD for the value, and where the policy network and value network each have two fully connected layers (1024→ 512 units, ReLu)
  - If you are implementing an existing paper, detail what you think will be the hardest part about implementing the model here.
    - I think the hardest part will be parsing the motion capture data to match our humanoid model's skeleton joint positions and training pipeline. Specifically, parsing the mocap correctly so that the first state of a training rollout is a frame from the mocap.
- **Metrics**: What constitutes "success?"
  - What experiments do you plan to run?
    - We plan to render a rollout of the learned policies (walk, run, strike, backflip or sidekick) and if the motion of the humanoid in the rollout (a) follows the task at hand and (b) looks realistic then those two criteria of the experiment constitute success.
  - For most of our assignments, we have looked at the accuracy of the model. Does the notion of "accuracy" apply for your project, or is some other metric more appropriate?
    - A metric of Task Adaptation would be more appropriate than accuracy. Explicitly, checking that with the Target Heading Control Task, the humanoid moves in a direction that is within +-1 radian of the desired direction, and for Strike tasks that humanoid's end effectors (hands and feet) strike the desired volume within 0.2 meters. Also, episode reward could be used as a general notion of accuracy.
  - If you are implementing an existing project, detail what the authors of that paper were hoping to find and how they quantified the results of their model.
    - The authors of the paper were looking to find an integration of multiple skills under one policy that could be adapted to new tasks such as Striking in a new target and Walking in a new direction. If the humanoid completed the tasks successfully, the model was a success.
  - What are your base, target, and stretch goals?
    - Base Goal: Get the Humanoid to Stand, Walk, and Run in a user-given specified direction with pure RL
    - Target Goal: Get the Humanoid to Stand, Walk, and Run in a user-given specified direction with Natural Movement using Motion Capture Imitation Reward.
    - Stretch Goal: Get the Humanoid to do a Backflip, Sidekick, Strike, and Throw.

Retarget Running Humanoid to run or jump through obstacle (gaps, rough terrain, cliffs.)

- **Ethics**: Choose 2 of the following bullet points to discuss; not all questions will be relevant to all projects so try to pick questions where there's interesting engagement with your project. (Remember that there's not necessarily an ethical/unethical binary; rather, we want to encourage you to think critically about your problem setup.)
  - **What broader societal issues are relevant to your chosen problem space?** ■ Simulating natural humanoid movement in a Physics Environment could also be useful for training Assistive Robots in the Physics Environment to assist with taking care of sick or disabled humans.
  - **Why is Deep Learning a good approach to this problem?**
    - Deep Learning is a good approach because Physics and RL can allow for highly scalable high-dimensional (many, continuous joints as features!) humanoid control without the manual slower process of keyframing used in animation.
  - What is your dataset? Are there any concerns about how it was collected, or labeled? Is it representative? What kind of underlying historical or societal biases might it contain? ○ Who are the major "stakeholders" in this problem, and what are the consequences of mistakes made by your algorithm?
  - How are you planning to quantify or measure error or success? What implications does your quantification have?
  - Add your own: if there is an issue about your algorithm you would like to discuss or explain further, feel free to do so.
- **Division of labor**: Briefly outline who will be responsible for which part(s) of the project ○ Jason: Creating Obstacle Arena for Humanoid to Run through and creating a height map using CNNs that we can pass through PPO Network, PPO hyperparameter tuning (timesteps, episodes, learning rate, network size)
  - Effy: Mujoco Environment Set-up, Defining State, Observation Space, and RL Rewards for PPO, PPO Baseline implementation, Task Rewards
  - Rainer: Parsing Motion Capture Data, Imitation Rewards, Implementing Reference State Init/Early Termination in the Humanoid Mujoco Env

Project Check-in #3
Due: Week of April 21, 2025
Rubric: TAs will be grading this check-in based on this rubric.
For the third check in, you will 1) write a one-page reflection on your progress so far and 2) meet with your mentor TA. We expect you are wrapping up the implementation and performing final experiments. If you have questions before the third check-in, please contact your mentor TA, or post questions on Ed.

Submit the reflection (as described below) by posting in on your repo or emailing your mentor TA before your meeting.
For this checkin, we also require you to write up a reflection including the following:

- **Introduction**: This can be copied from the proposal.
  - We are re-implementing DeepMimic (Peng et al. 2018) to enable a simulated humanoid to imitate motion-capture clips and adapt those skills for new tasks (e.g., steering, obstacle traversal, target strikes). DeepMimic's core innovation is combining a motion-imitation reward with task-specific rewards and two key training strategies, Reference State Initialization (RSI) and Early Termination (ET), to produce generalizable and natural character movement.
- **Challenges**: What has been the hardest part of the project you've encountered so far? ○ The hardest part of the project we have encountered so far was the parsing of the motion capture clips to fit our humanoid. Our humanoid has less joints than the motion capture skeleton and the Mujoco

humanoid's center mass is its upper torso while the mocap center of mass is its hips. We had to modify our Mujoco's humanoid's joint hierarchy to match this, and use less of the mocap joints for the imitation reward.

- **Insights**: Are there any concrete results you can show at this point? How is your model performing compared with expectations?
    - Yes! Please take a lot at some of our output videos in the link here:https://drive.google.com/drive/folders/1OVTDVivokqNsM-RJYNiH8cwv_m6ti0FI?usp=drive_link .We have succeeded in making the model stay upright and Stand, Walk, and Run with target speed and target direction. So far, the model is doing the Target Heading Reward as expected, except it cannot seem to walk in a theta direction of 180 degrees, which is the opposite direction of where it is facing. The model struggles to
- **Plan**: Are you on track with your project? What do you need to dedicate more time to? What are you thinking of changing, if anything?
    - We are on Track with our Base and Target goals. We need to dedicate more time to debugging the imitation reward and learning to create different arenas (gaps in the floor, terrain) in Mujoco. We are thinking of changing our workflow from our local computers to Colab and MJX which uses Mujoco on the GPU instead of the CPU, for a more efficient workflow using GPU acceleration.

This check in meeting with your mentor TA can either be in-person or over Zoom, Google Meet, etc. Reach out to your mentor TA before 04/24 to schedule this meeting.

Regarding what we generally expect you to have **done** by this time:

- You should have collected any data and preprocessed it. (See Parse MoCap cell of .ipynb in Github!)
- You should have shared the GitHub repo link with your mentor TA (See link:https://github.com/effypelayotran/C_Undergrad_Neural_Trainers.git)
- You should have almost finished implementing your model, and are working on training your models and ablation experiments. (See video output folder: https://drive.google.com/drive/folders/1OVTDVivokqNsM-RJYNiH8cwv_m6ti0FI?usp=drive_link)
- Please make sure you are keeping your list of public implementations you've found up-to-date.

(Yes, this is the paper's public implementation, straight from the original DeepMimic Paper: https://github.com/xbpeng/DeepMimic)