ThreeDWorld (TDW) – A Multi-Modal Platform for Interactive Physical Simulation
Jeremy Schwartz, Brain and Cognitive Sciences at MIT
DiCarlo, McDermott and Tenenbaum Labs

- Addresses the difficulty and cost of acquiring large amounts of labeled data for training machine perceptual systems
- Generating scenes in a virtual world enables full access to all generative parameters
- Uses state-of-the-art videogame technology to collect experimental data and generate large-scale datasets for training AI systems
- Publicly released: https://github.com/threedworld-mit/tdw Website: www.threedworld.org
- General, flexible design enables a broad range of use cases, including a) visual recognition transfer; b) multi-modal physical scene understanding; c) learnable physics models and d) visual learning in curious agents

For details on use-cases, see paper "ThreeDWorld: A Platform for Interactive Multi-Modal Physical Simulation"

---

**Overview of TDW Features and Capabilities**

**Visual Modality:**
Near-photoreal image rendering using real-time GI, HDR lighting model and physically based rendering materials

**Auditory Modality:**
High-fidelity audio, with real-time, physics-based impact sound synthesis via PyImpact python library

Supports both interior and exterior environments:
- Highly-detailed interiors
- Exterior environments use 3D assets scanned from real-world
- Populate environments with high-quality 3D models, optimized for research purposes – procedurally or fully scripted

Advanced physical interactions:
- Fast but accurate rigid body collisions
- Uniform, particle-based approach supports rigid body, soft body, cloth and fluids
- Physics benchmark dataset for training and evaluation of physically-realistic forward prediction algorithms

Multiple paradigms for object interaction:
- Direct – use API commands, e.g. apply force so ball collides with stack of cups
- Indirect – Avatar as embodiment of agent. Range from simple camera to “sticky-mitten” avatar with articulated arms to lift objects
- Direct Human – user as Agent in VR; interact with objects using “hands”

High-level architecture:
- The Build is a compiled executable running on the Unity3D Engine, responsible for image rendering, audio synthesis and physics simulations
- The Controller is an external Python interface to communicate with the build.
- Controller sends commands to Build; Build returns wide range of output data types representing the “state of the world”

Rich command and control Python API:
- Over 200 commands
- Extensive documentation, including multiple example and use-case controllers
- Controller can send multiple commands per time-step, for complex behaviors

---

**Ongoing Development Goals**

- Expand capabilities of PyImpact sound synthesis – more audio materials, support for scraping and rolling sounds
- Interface to Robotic Operating System (ROS), enable import of existing robotics assets via URDF
- Add ability to receive haptic feedback from physical interactions
- Integrate NVIDIA GPU raytracing for enhanced photorealism
- Develop a humanoid avatar with fully-rigged skeleton and soft-fine-grained hand gripper