



March 20-24, 2023
San Francisco, CA

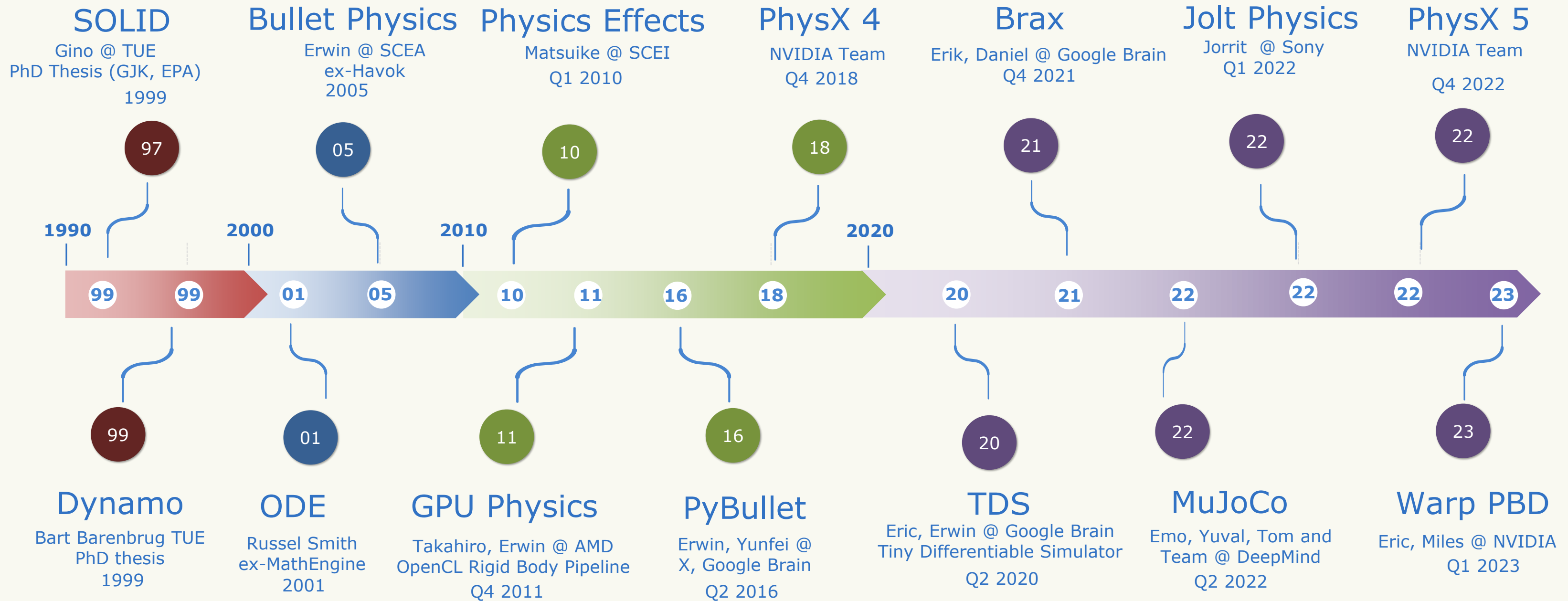
Evolution of Open Source 3D Physics Simulators

Erwin Coumans
Distinguished Engineer @ NVIDIA

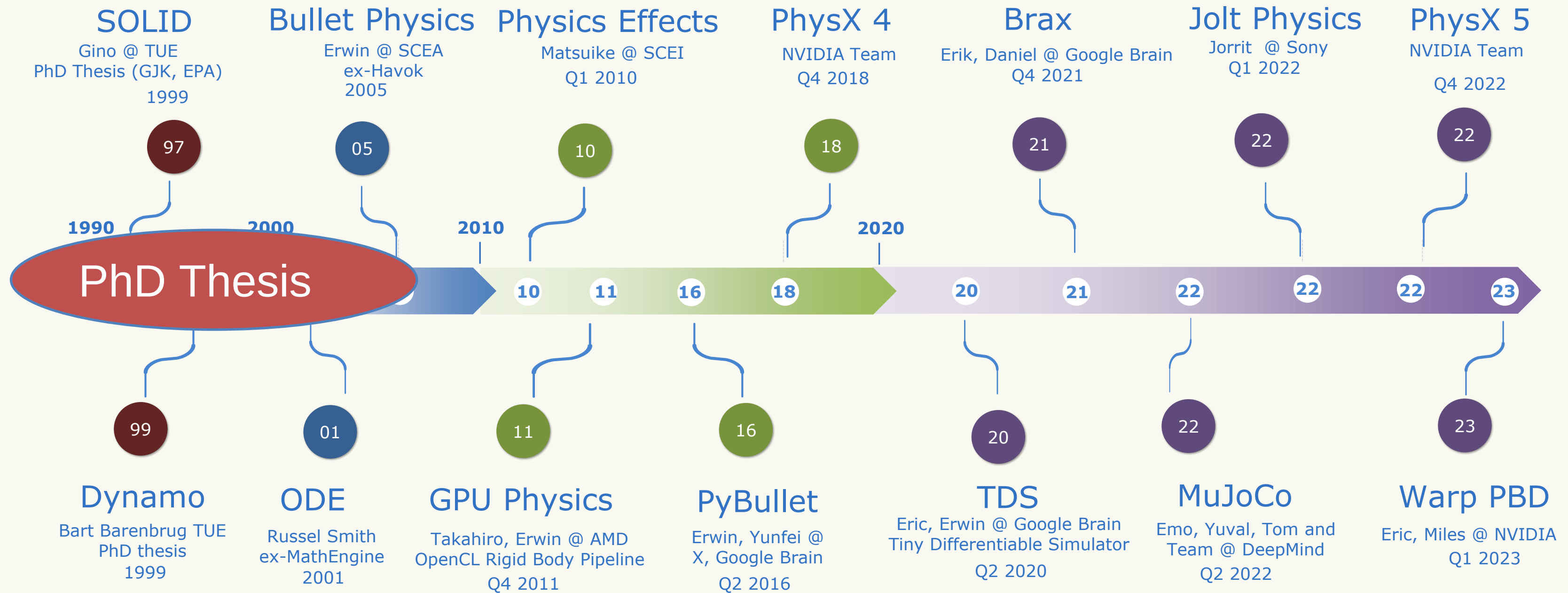
[link to pptx version with vides](#)

#GDC23

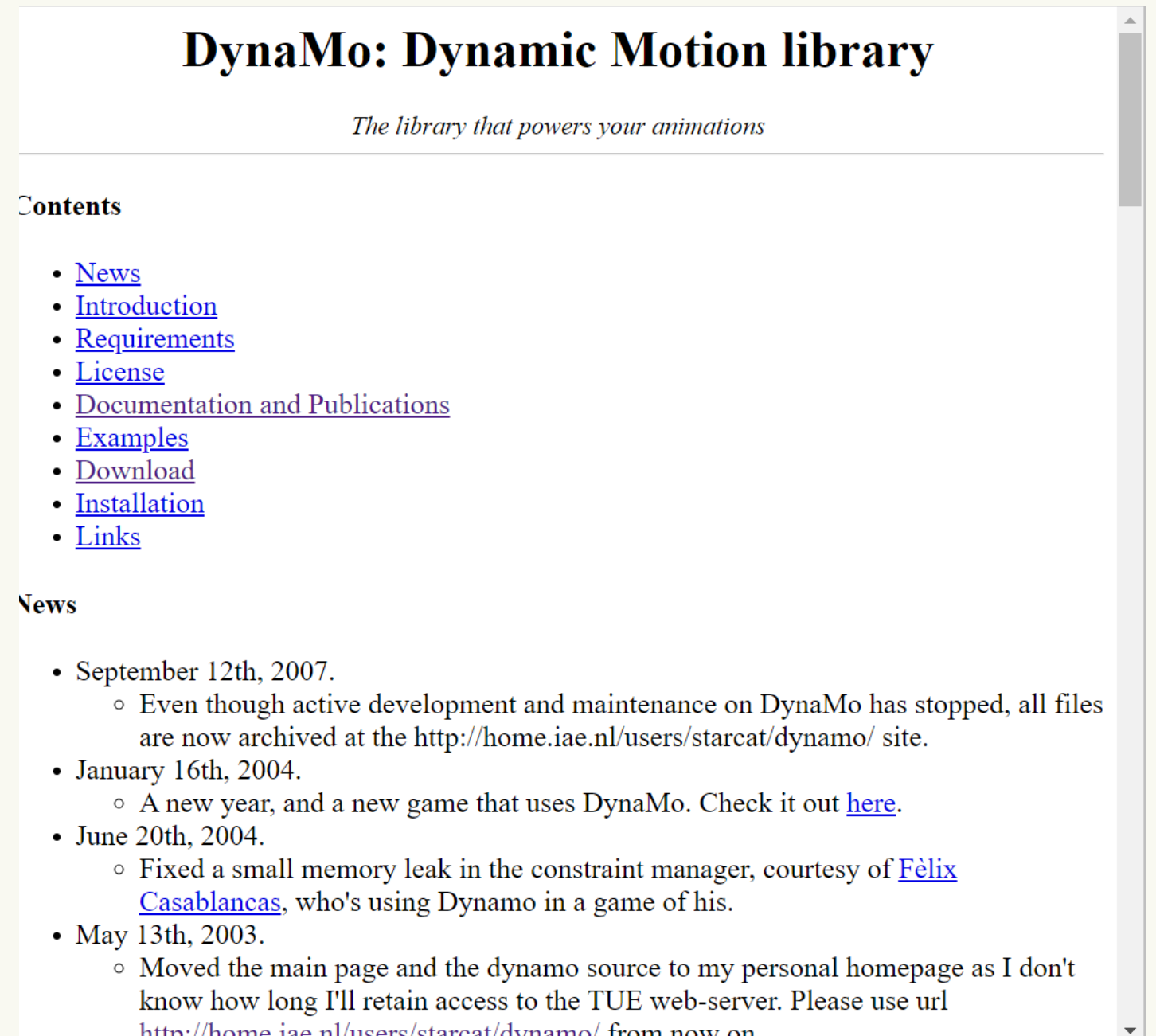
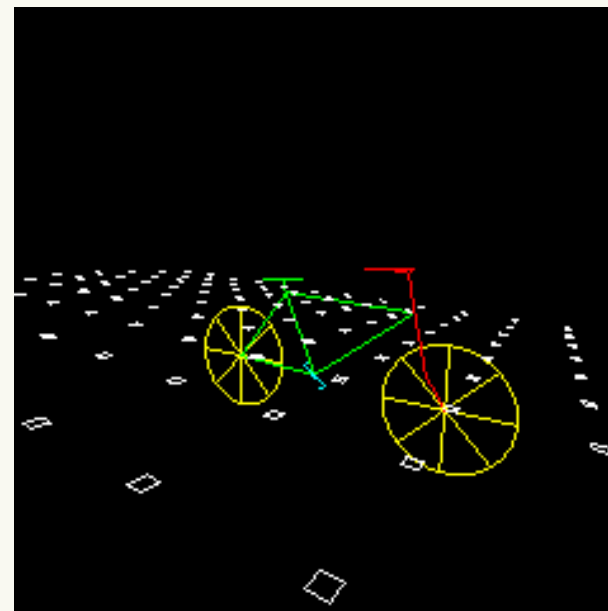
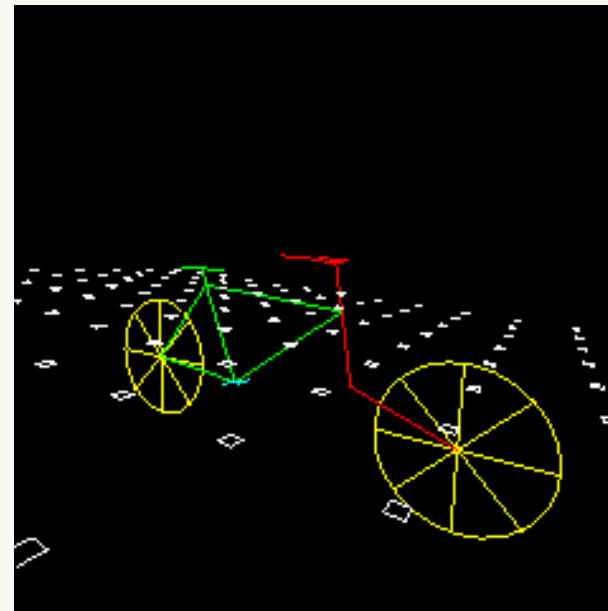
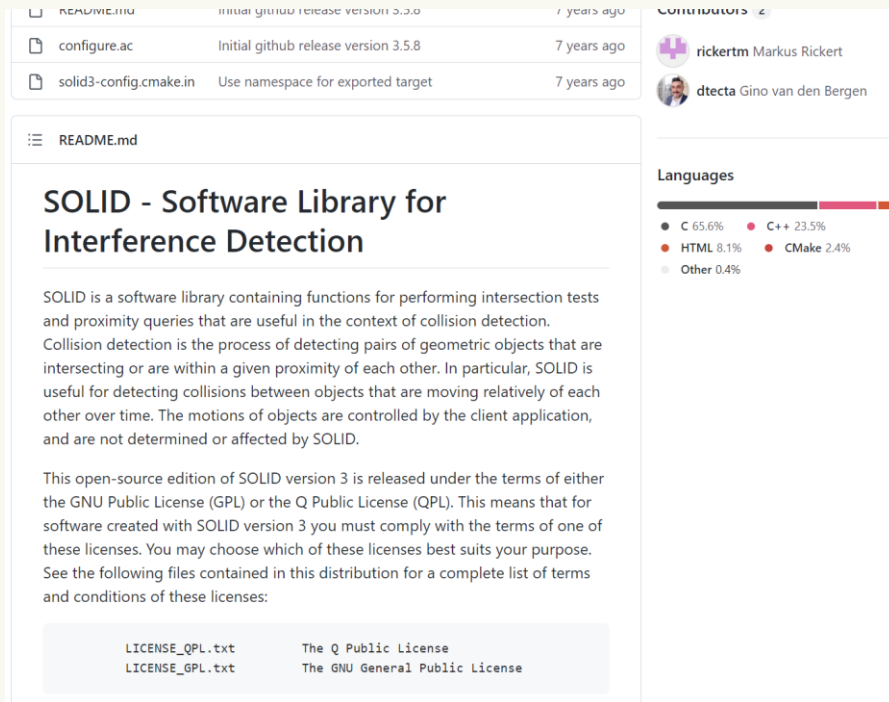
Open Source Timeline



Motivation



SOLID, Dynamo (1999)



<https://github.com/dtecta/solid3>

Dynamo PhD Thesis

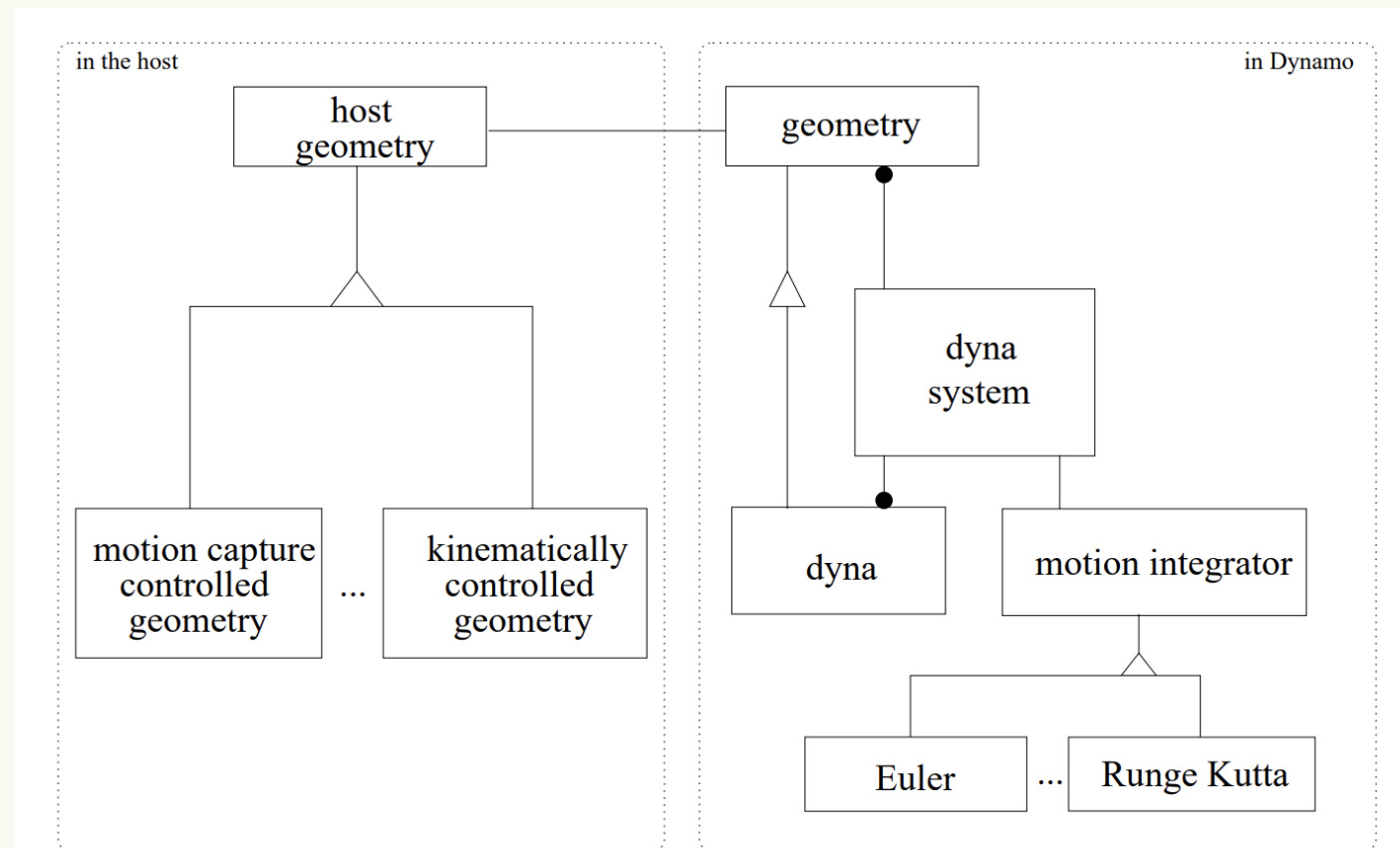
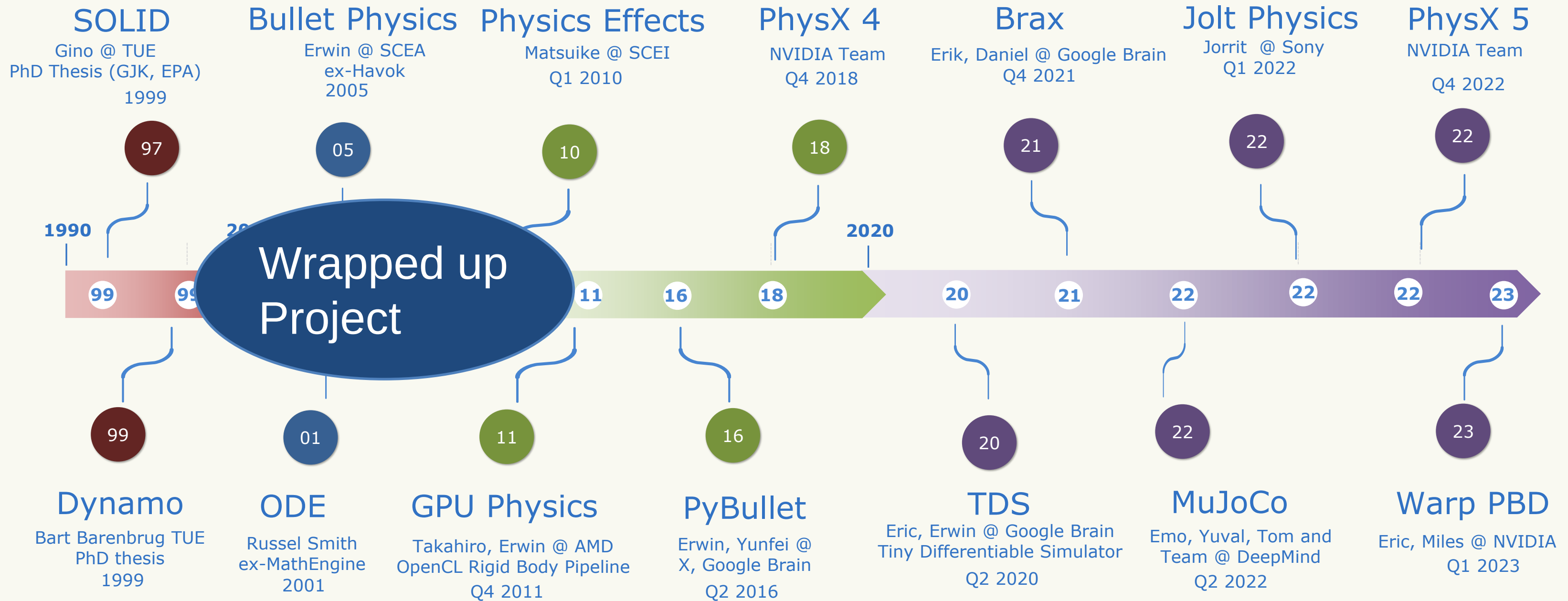


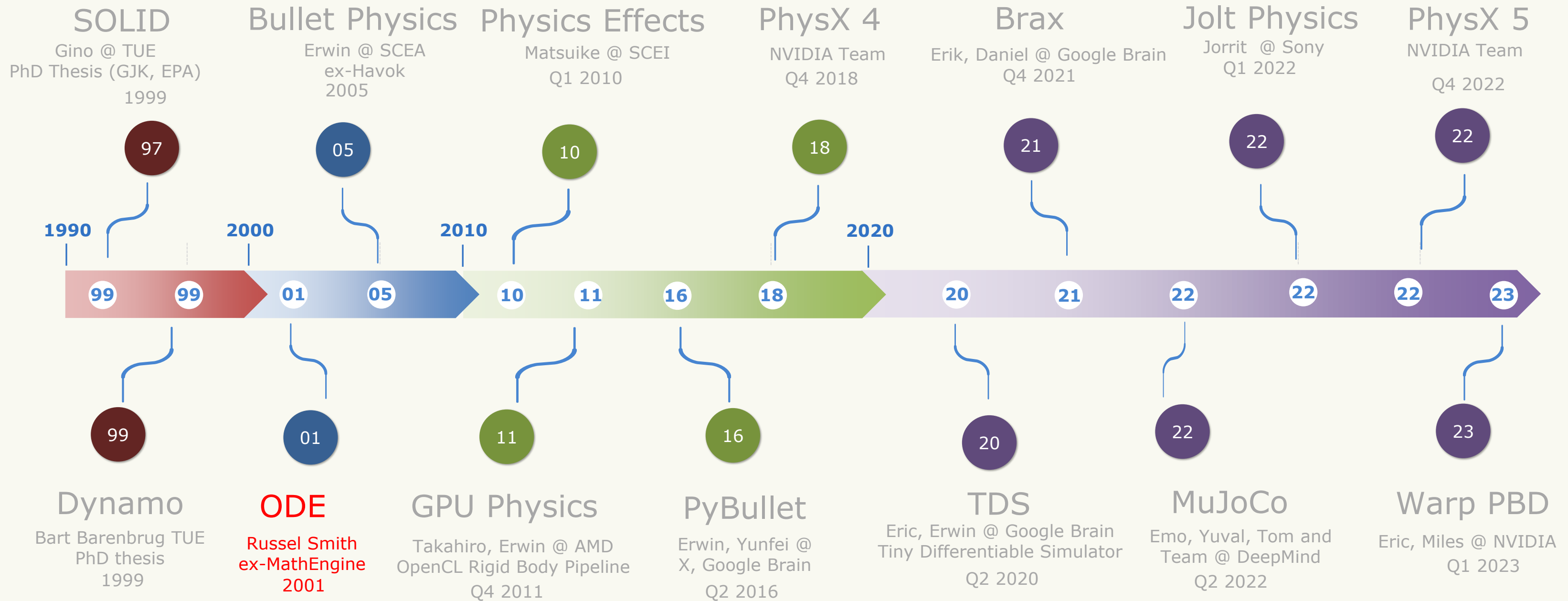
Figure 3.1: The major classes in the forward dynamics subsystem

See BartThesis.pdf in the Doc folder here:
<https://github.com/erwincoumans/dynamo>

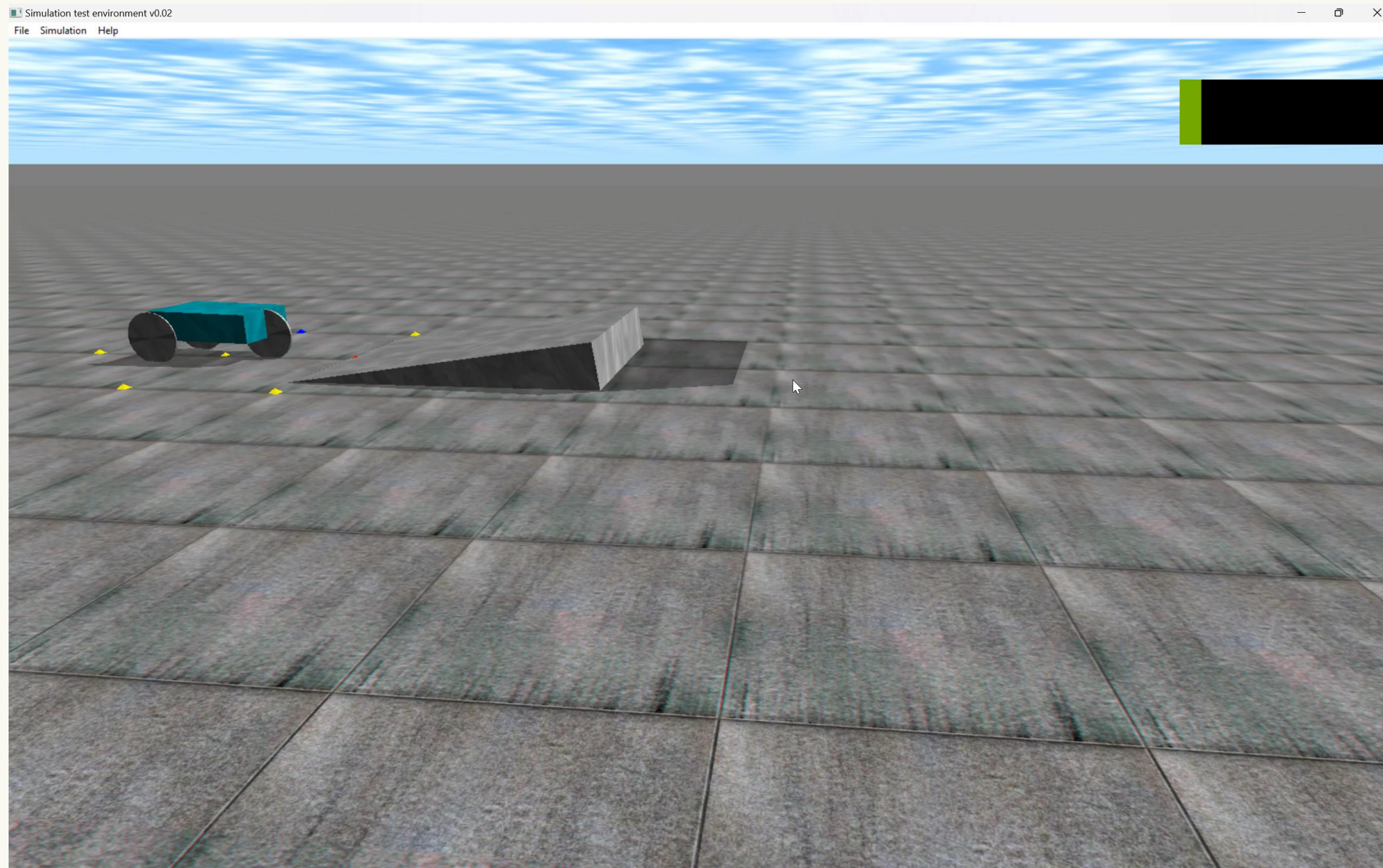
Motivation



Open Dynamics Engine



Open Dynamics Engine



Open Dynamics Engine

Russell L. Smith.

<https://www.q12.org/phd/thesis/thesis.pdf>

Reinforcement Learning

OpCode collision detection by Pierre Terdiman

ODE is maintained by Oleh Derevenko and contributors

BSD-3 License

<https://www.ode.org>

git clone <https://bitbucket.org/odedevs/ode.git>

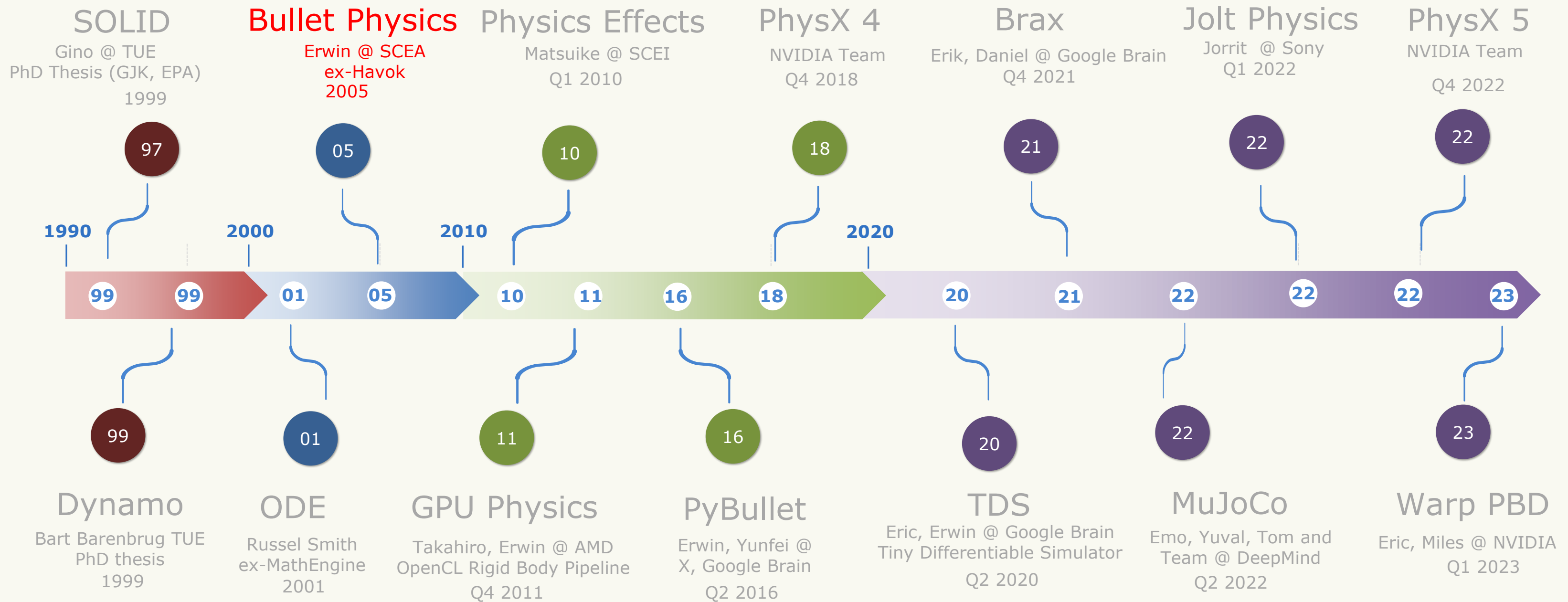
- `dSpaceCollide (space,0,&nearCallback);`
- `dWorldQuickStep (world, simstep);`
- `dBodyGetPosition(bodyId);`

Open Dynamics Engine

[ode-users - Google Groups](#)

≡	 Groups	🔍 Conversations ▾	Search conversations withi... ▾	⚙️	⋮
☆	ode-users	1124 members	1-30 of 2111	<	>
□	↻	⋮			
	Oleh Derevenko	ODE 0.16.3 was released — Hi, I've published ode-0.16.3.tar...	12/19/22	☆	
	mids...@gmail.com 2	Capsule can slide up certain walls and boxes — As an updat...	10/22/22	☆	
	ril...@na..., ... zheng... 16	Making 64-bit ode.dll on Windows — Hi Riley Do you solve th...	9/22/22	☆	
	Oleh Dere..., ... Vaillan... 3	An administrative announcement — I'm sorry it has gone thi...	2/24/22	☆	
	Vaillan..., ... Oleh Dere... 6	How can I limit a hinge2 joint to a single direction? — Hi Da...	7/3/21	☆	
	gviz...@gmail.com	Integrating ODE4j for OBB-OBB intersetion — Greetings, I ha...	6/12/21	☆	
	Oleh Derevenko	Bitbucket Pipeline Changes — Hi, They appear to have chan...	5/26/21	☆	
	taro.51...@..., Oleh De... 3	How to solve "ODE Message 3: LCP internal error, s <= 0 (s=-0	5/13/21	☆	

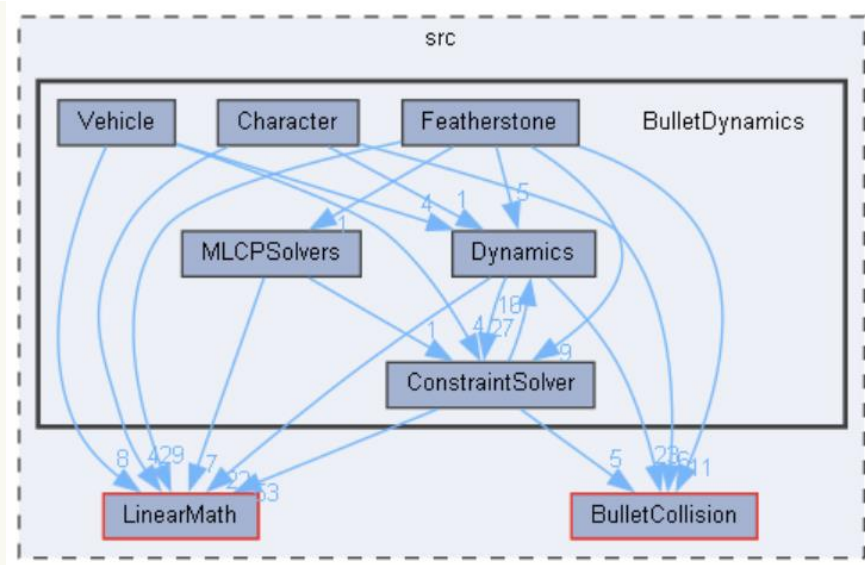
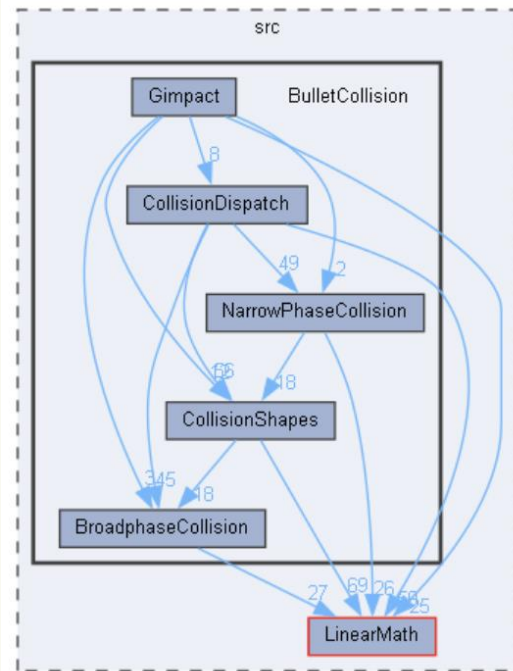
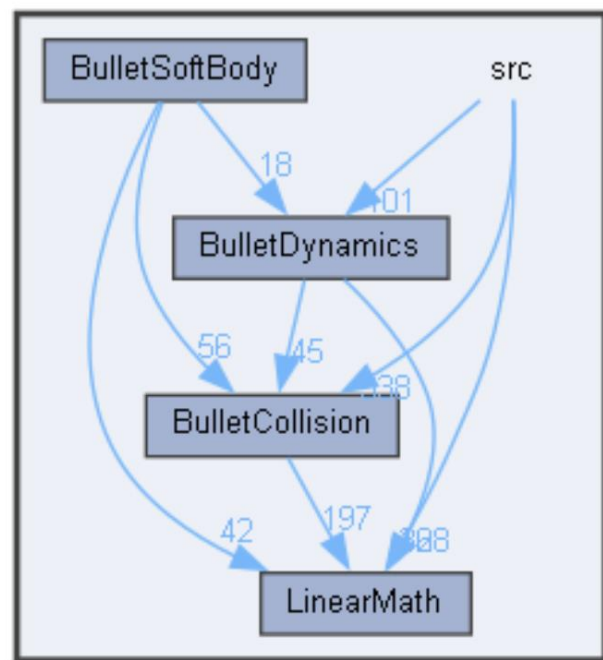
Bullet Physics



Bullet Physics



Bullet Physics

[Main Page](#)[Namespaces](#)[Classes](#)[Files](#)

Bullet Documentation

Introduction

Bullet is a Collision Detection and Rigid Body Dynamics Library. The Library is Open Source and free for commercial use, under the ZLib license (<http://opensource.org/licenses/zlib-license.php>)

The main documentation is Bullet_User_Manual.pdf, included in the source code distribution. There is the Physics Forum for feedback and general Collision Detection questions.

Installation

Step 1: Download

You can download the **Bullet** Physics Library from the github repository: <https://github.com/bulletphysics/bullet3/releases>

Step 2: Building

Bullet has multiple build systems, including premake, cmake and autotools. Premake and cmake support all platforms. Premake is included in the Bullet/build folder to create Microsoft Visual Studio projects. On Mac OSX and Linux you can open a terminal and generate Makefile, codeblocks or Xcode4 projects: `cd Bullet/build . ./premake4_osx xcode4 cd Bullet/build/gmake make`

An alternative to premake is cmake. You can download cmake from <http://www.cmake.org> cmake can autogenerate projectfiles for Microsoft Visual Studio, Apple Xcode and choose the options and generate projectfiles. You can also use cmake in the command-line. Here are some examples for various platforms: `cmake -G "Visual Studio 10" .` recommended, you can also use autotools for UNIX: `./autogen.sh ./configure` to create a Makefile and then run make.

Step 3: Testing demos

Try to run and experiment with BasicDemo executable as a starting point. **Bullet** can be used in several ways, as Full Rigid Body simulation, as Collision Detection only, or as a physics engine for a game engine. See the examples seen in this documentation under Directories

Step 4: Integrating in your application, full Rigid Body and Soft Body simulation

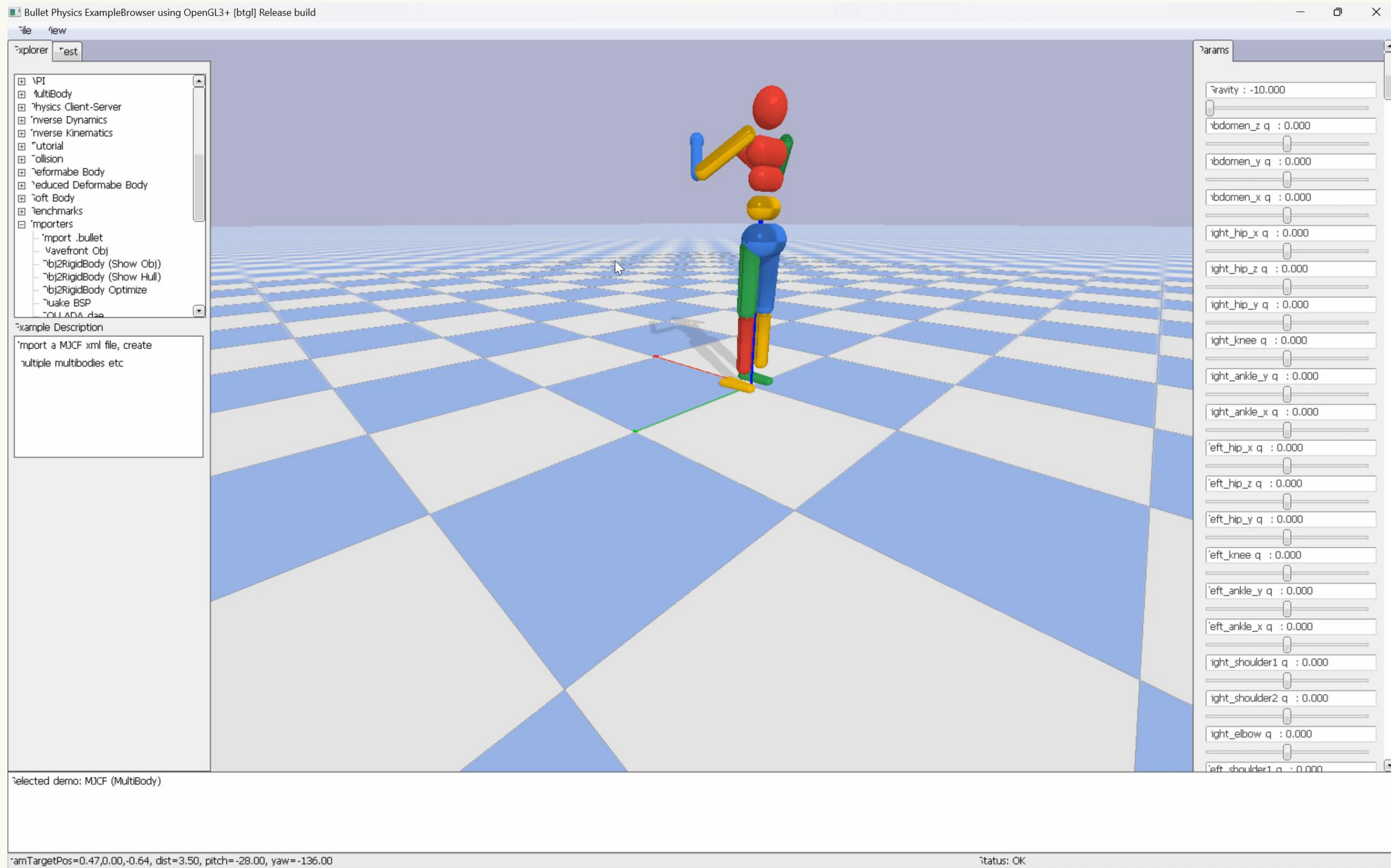
Check out BasicDemo how to create a **btDynamicsWorld**, **btRigidBody** and **btCollisionShape**, Stepping the simulation and synchronizing your graphics objects with the physics engine using **btSoftRigidDynamicsWorld**.

Step 5 : Integrate the Collision Detection Library (without Dynamics and other Extras)

Bullet Collision Detection can also be used without the Dynamics/Extras. Check out **btCollisionWorld** and **btCollisionObject**, and the CollisionInterfaceDemo.

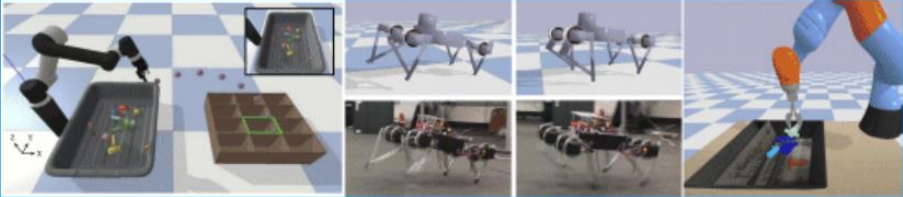
Step 6 : Use Snippets like the GJK Closest Point calculation.

Bullet has been designed in a modular way keeping dependencies to a minimum. The ConvexHullDistance demo demonstrates direct use of **btGjkPairDetector**



<https://github.com/bulletphysics/bullet3>

Physics Simulation Forum



Real-Time Physics Simulation Forum

Search...

Quick Links

FAQ

Register

Login

Board index

It is currently Tue Mar 21, 2023 4:18 am

BULLET PHYSICS LIBRARY USERS		TOPICS	POSTS	LAST POST
	General Bullet Physics Support and Feedback	8211	29721	defining a robot not defined ... by nojo  Mon Mar 20, 2023 9:21 am
	PyBullet Support and Feedback Official Python bindings with a focus on reinforcement learning and robotics.	321	717	Re: objects not visible after... by maurock  Tue Mar 14, 2023 5:27 pm
	Release Announcements Open source Bullet Physics SDK release information	76	245	Re: PyBullet and Bullet 2.89 ... by Mark97  Fri Oct 01, 2021 7:01 pm
	Applications, Games, Demos or Movies using Bullet Show what you made with Bullet Physics SDK: Games, Demos, Integrations with a graphics engine, modeler or any other application	354	1243	Re: [GAME] Coin Plunger. Medi... by danielhertz  Thu Mar 09, 2023 11:00 pm

PHYSICS AUTHORING TOOLS, SERIALIZATION AND STANDARDS		TOPICS	POSTS	LAST POST
	Physics authoring tools, serialization, standards and related topics Physics APIs, Physics file formats, Maya, Max, XSI, Cinema 4D, Lightwave, Blender, thinkingParticles™ and other simulation tools, exporters and importers	514	1601	Re: Bullet serialization from... by sky10086  Thu Oct 27, 2022 12:53 pm

RESEARCH AND DEVELOPMENT IN COLLISION DETECTION & PHYSICS. DON'T POST BULLET SUPPORT QUESTIONS HERE!		TOPICS	POSTS	LAST POST
	Research and development discussion about Collision Detection and Physics Simulation <i>Please don't post Bullet support questions here, use the above forums instead.</i>	1304	4819	Re: Impulse Solvers and Conti... by spaceDoctor  Fri Mar 03, 2023 8:54 pm
	Links, Papers, Libraries, Demos, Movies, Comparisons Subforum:  Non-technical forum and license/patent discussion	281	1210	Re: A New Continuous Collisio... by christian_scheurer  Thu Sep 22, 2022 7:41 pm
	Career Opportunities Job seekers and recruitment for game physics developers.	94	123	Re: Physics Programming oppor... by davidsmithhello  Thu Mar 09, 2023 10:50 am

March 20-24, 2023 | San Francisco, CA

#GDC23

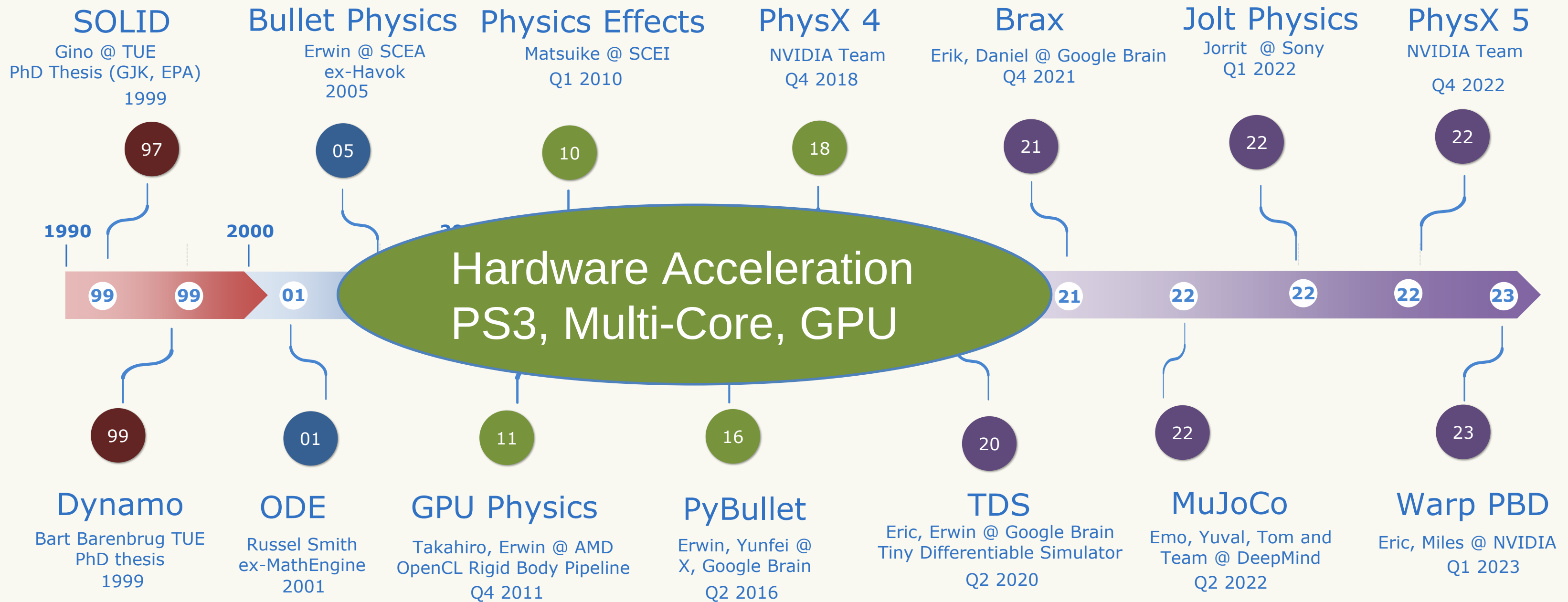


Bullet in Games and VFX

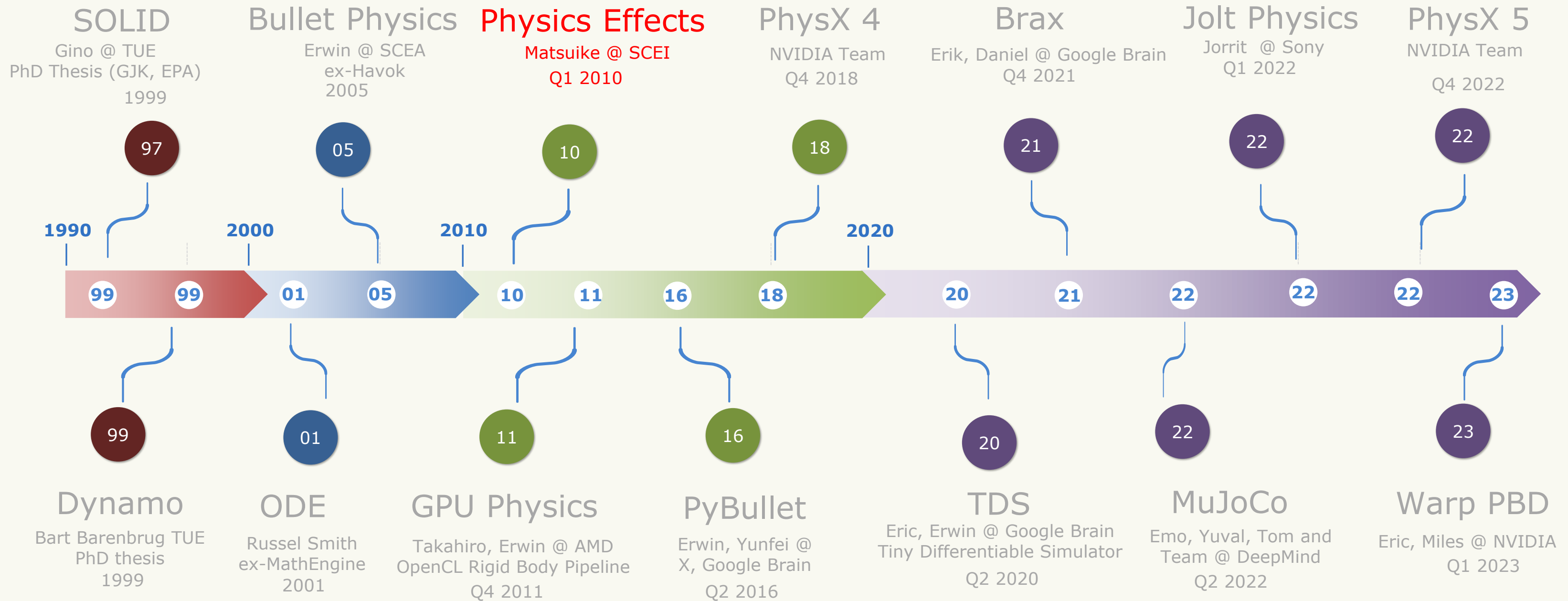


User Feedback defines the Product

Motivation



Physics Effects

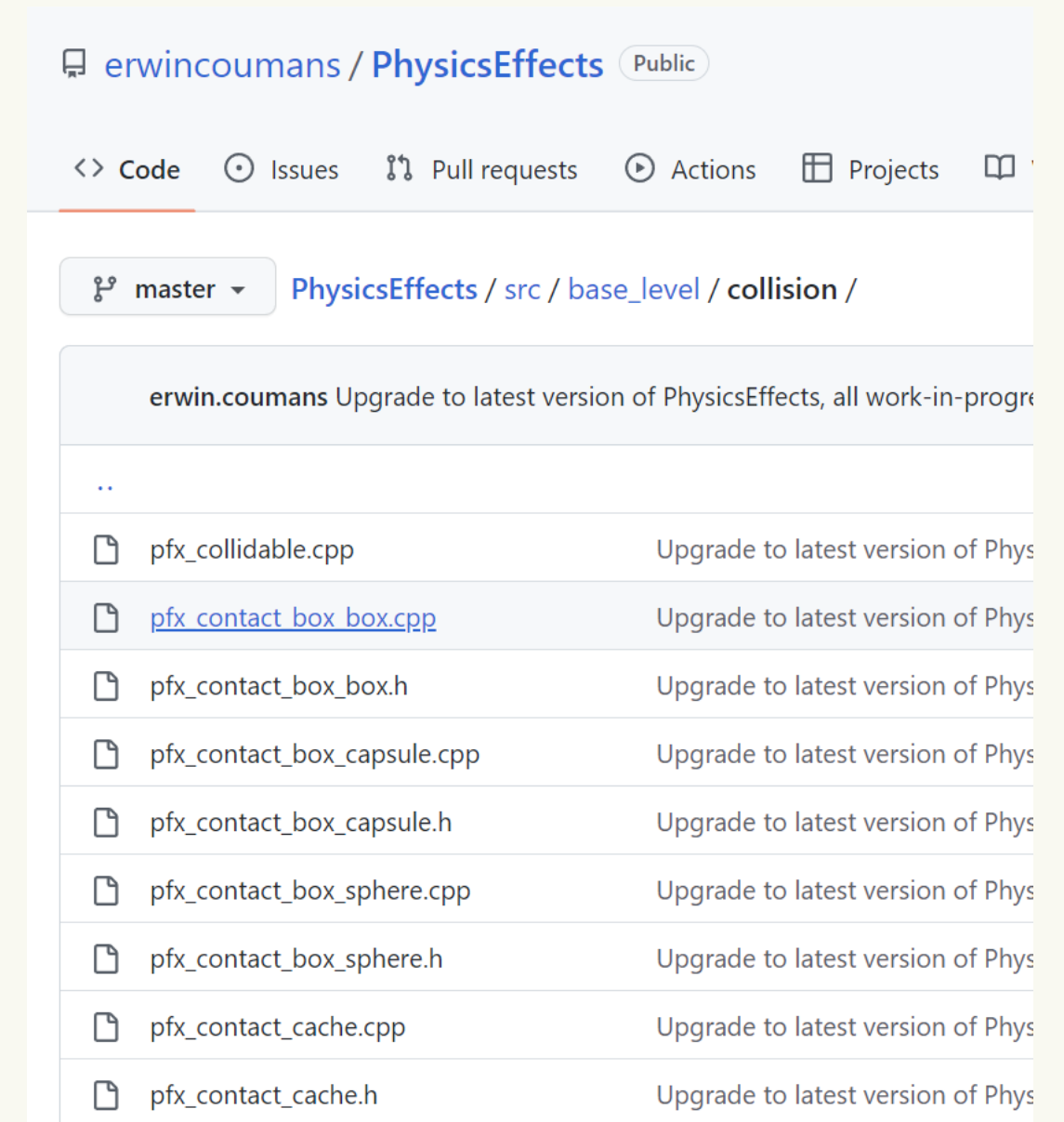


Physics Effects ‘Acceleration’

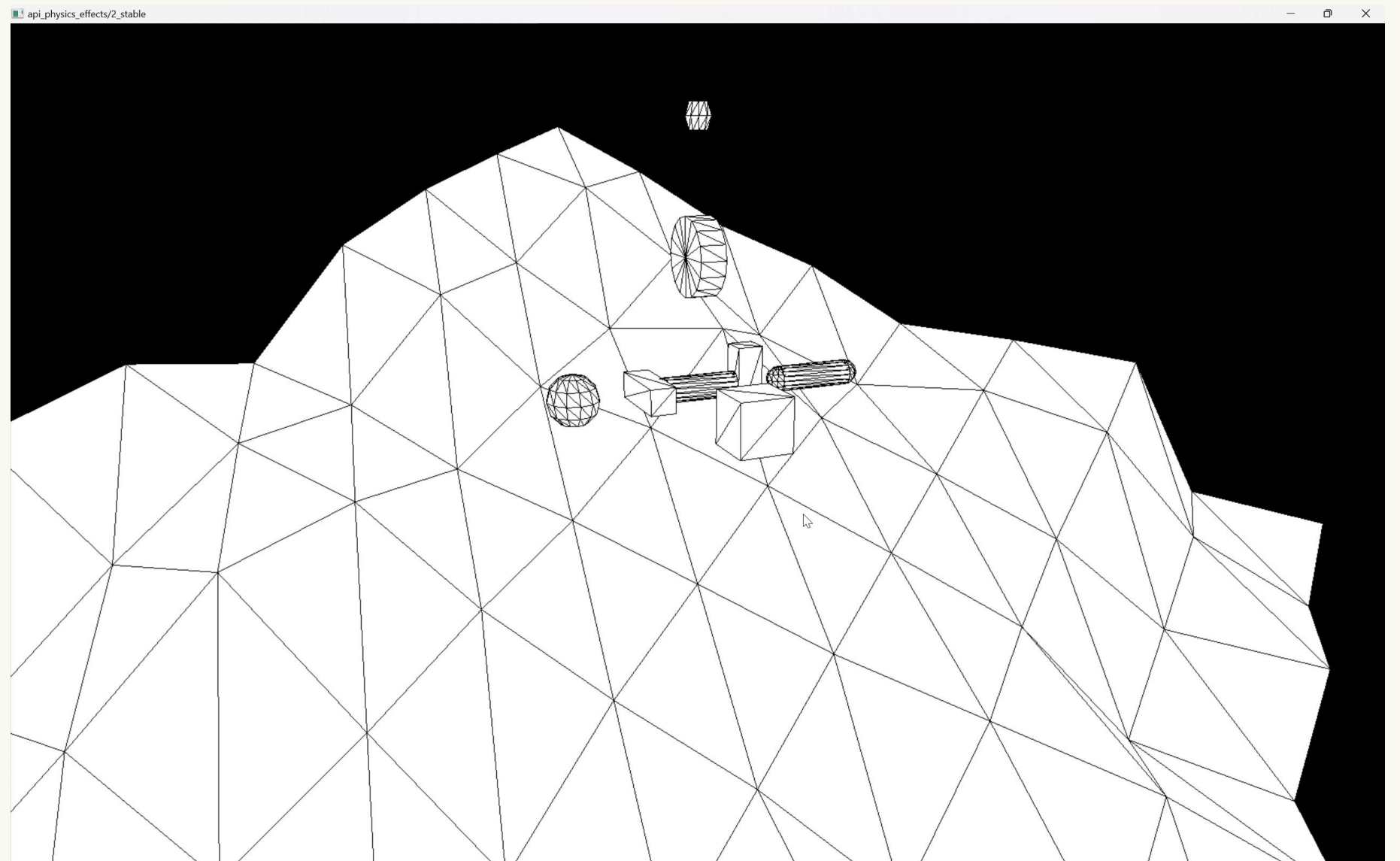
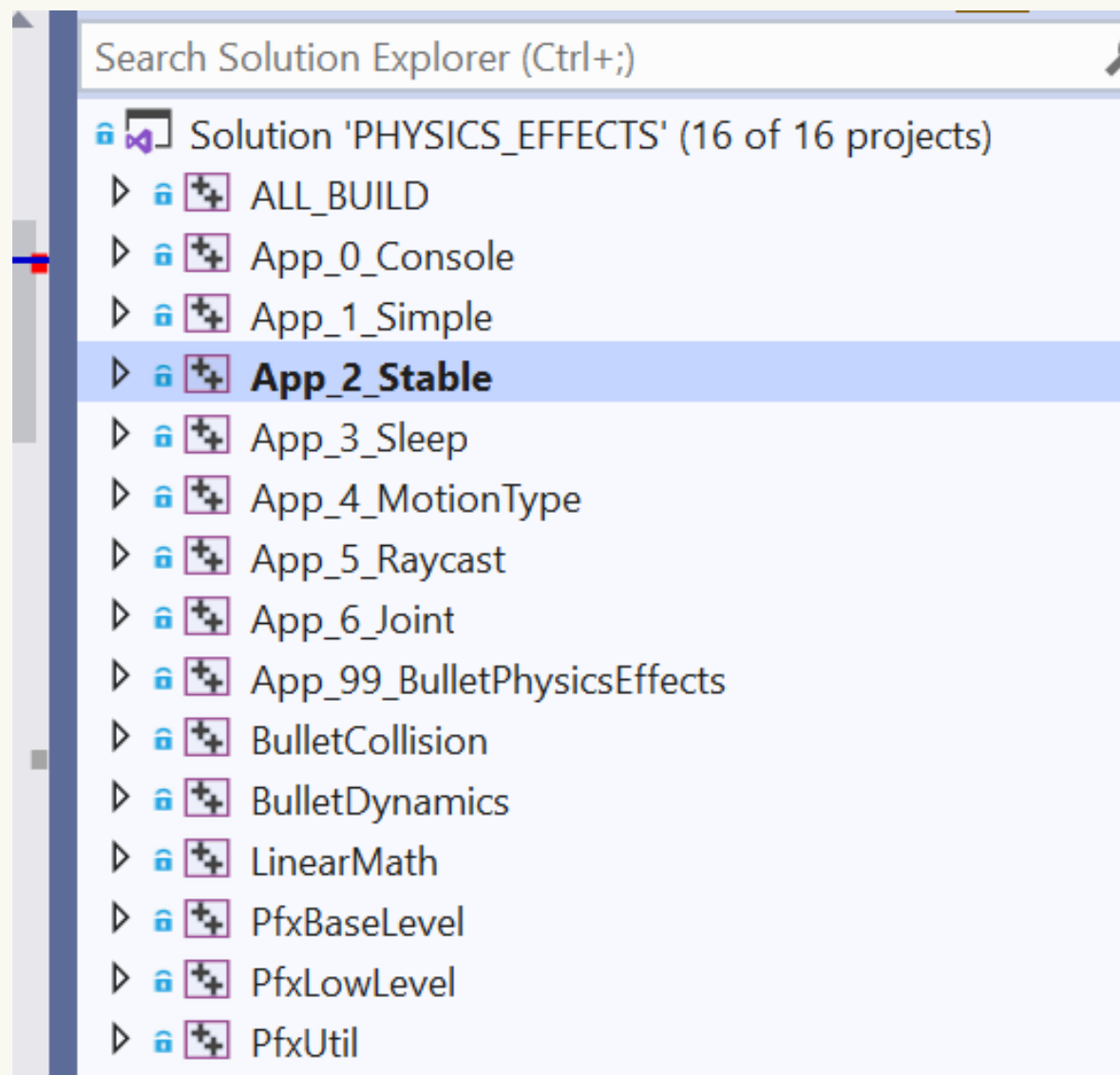
Immediate Mode Physics
base level + low level
PS3 SPU Implementation

COLLADA Physics format
Unreal Engine Integration

Long term goal:
Open physics cross-platform standardization



Physics Effects



<https://github.com/erwincoumans/PhysicsEffects>

COLLADA Physics

Physics Engine	PhysX	Havok	ODE	Bullet
Box, capsule, sphere, cylinder	Yes	Yes	Yes	Yes
Convex mesh	Yes	Yes	Partial	Yes
Concave Triangle mesh	Yes	Yes	Yes	Yes
Compound shapes	Yes	Yes	Yes	Yes
Recursive Shapes	No	Yes	No	Yes
Rigid Body mass, inertia	Yes	Yes	Yes	Yes
Friction, Restitution	Yes	Yes	Yes	Yes
Center of mass offset	Yes	Yes	Yes	Yes
Hinge, Ball Socket, Slider	Yes	Yes	Yes	Yes
Motors, Limits	Yes	Yes	Yes	Partial
Generic 6DOF Constraint	Yes	TBD	TBD	Partial

Table 2. Physics Engines common features

3D Tool	Collision shapes	Rigidbody	Constraints
Maya 8 + COLLADA Maya [Feeling]	All primitives, compound, convex & concave mesh	Yes	Yes
3ds Max 8 + PhysX 2.6.2 plugin [Ageia]	Yes	Yes	Yes
XSI	Planned	Planned	Planned
Blender 2.43	Box, sphere, cylinder, convex and concave mesh	Yes	Partial

Table 3. COLLADA Physics tools support

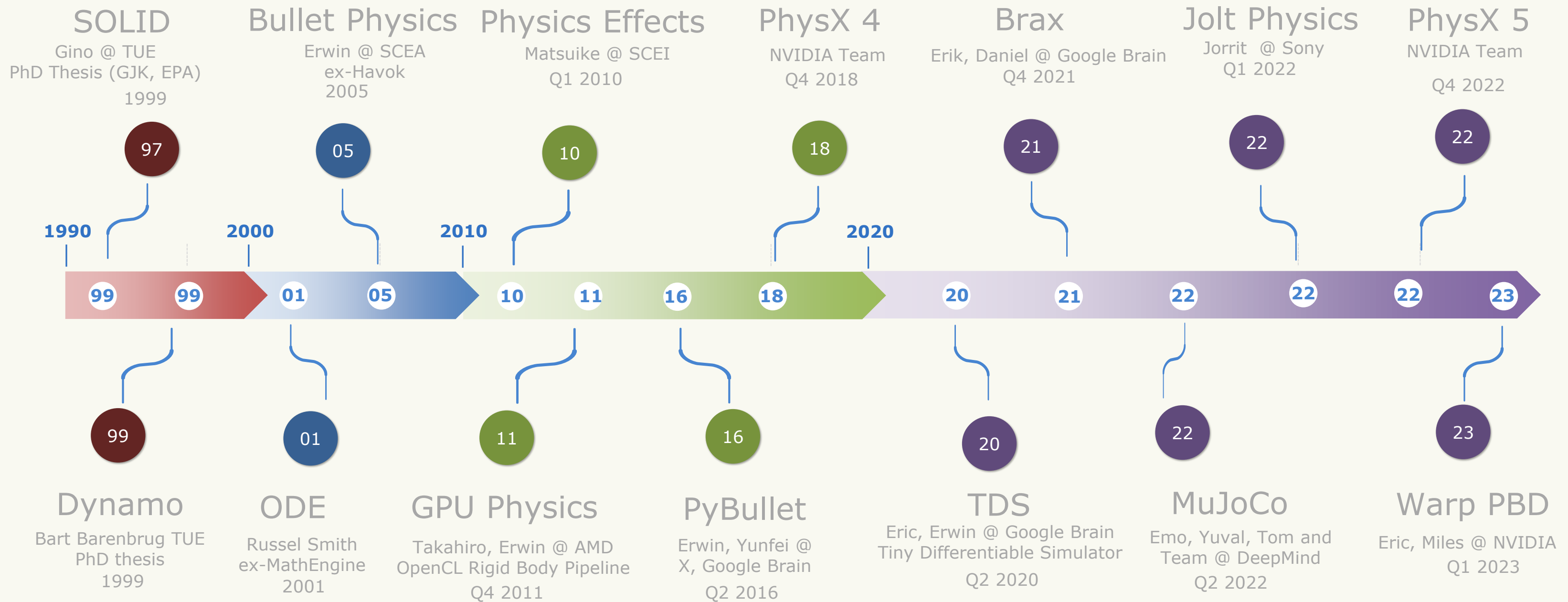
```
<rigid_body name="RigidBody1" sid="RigidBody1">
  <technique_common>
    <dynamic sid="RigidBody1">1</dynamic>
    <instance_physics_material url="#RigidBody1-material"/>
    <shape>
      <hollow>false</hollow>
      <sphere>
        <radius>10.000000</radius>
      </sphere>
    </shape>
  </technique_common>
</rigid_body>
```

```
<?xml version="1.0" encoding="utf-8"?>
<COLLADA
  xmlns="http://www.collada.org/2005/11/COLLADASchema"
  version="1.4.1">
  <library_physics_scenes>
    <physics_scene id="NativePhysicsScene">
      <instance_physics_model url="#PhysicsModel1">
        <instance_rigid_body target="#RigidBody1"
          body="graphicsNode1">
          <technique_common/>
        </instance_rigid_body>
      </instance_physics_model>
      <technique_common>
        <gravity>0 -9.81 0</gravity>
      </technique_common>
    </physics_scene>
  </library_physics_scenes>
  <scene>
    <instance_physics_scene url="#NativePhysicsScene"/>
    <instance_visual_scene url="#VisualSceneNode"/>
  </scene>
</COLLADA>
```

```
<physics_material id="RigidBody1-material"
  name="RigidBody1-material">
  <technique_common>
    <dynamic_friction>0.200000</dynamic_friction>
    <restitution>0.600000</restitution>
    <static_friction>0.200000</static_friction>
  </technique_common>
</physics_material>
```

See Chapter 6 “COLLADA Physics Reference” in <https://www.khronos.org/collada>

PhysX 4

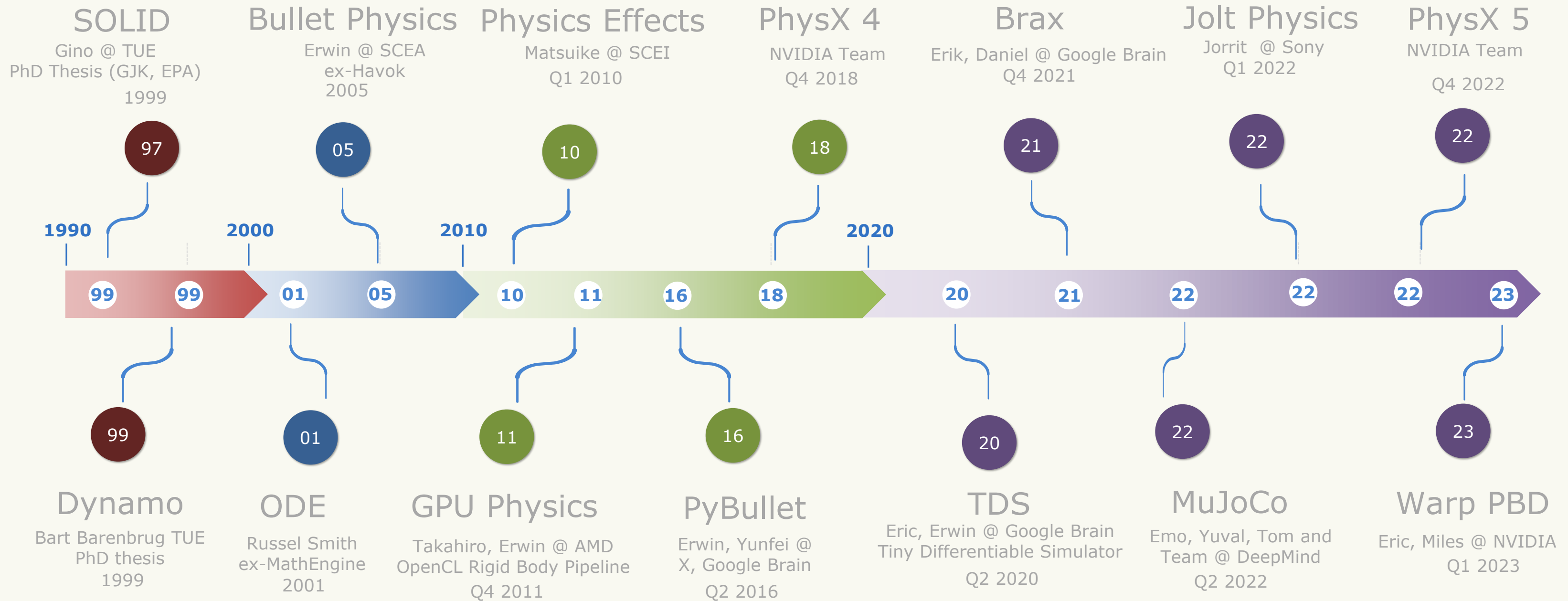


PhysX 4

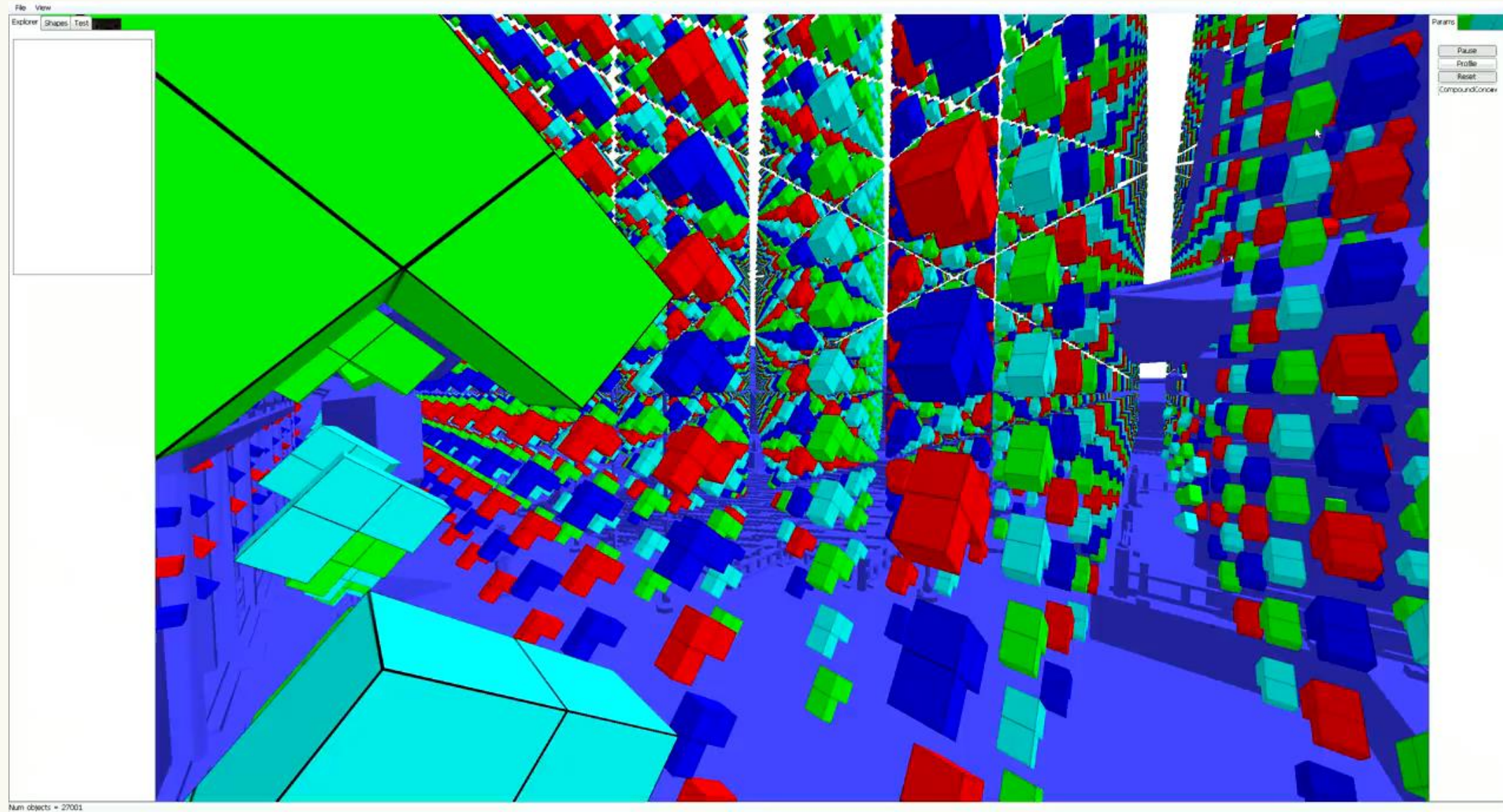


<https://github.com/NVIDIAGameWorks/PhysX>

GPU Physics

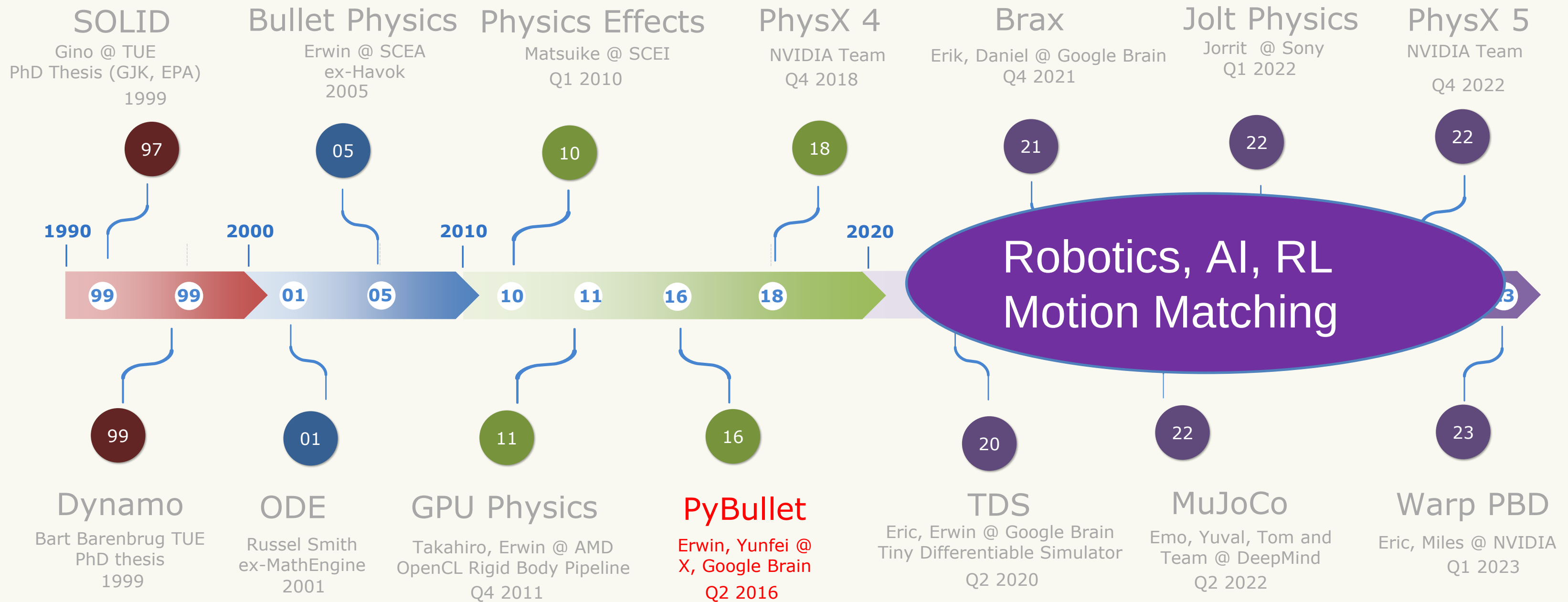


GPU Physics



<https://github.com/bulletphysics/bullet3/tree/master/src/Bullet3OpenCL>

PyBullet

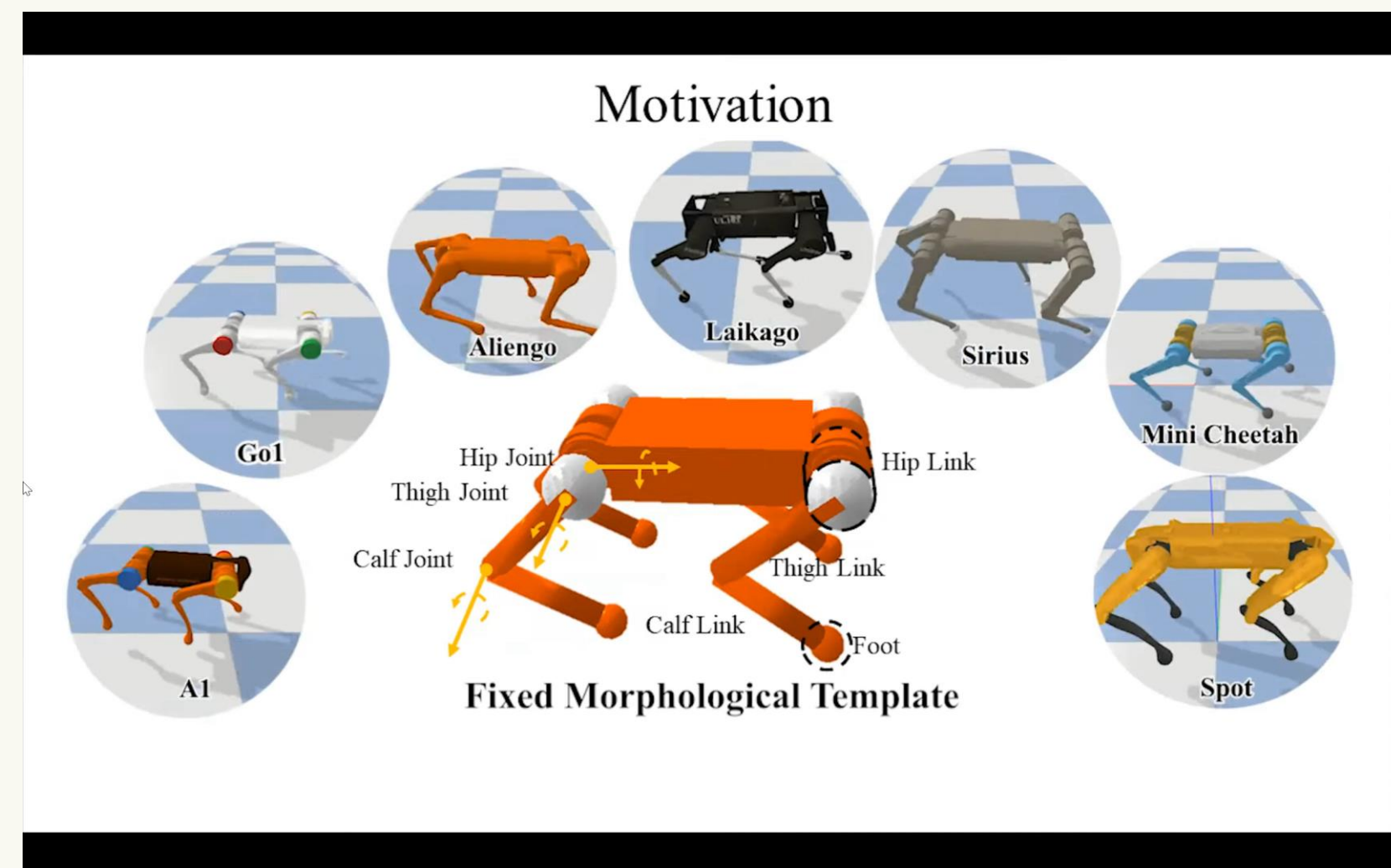
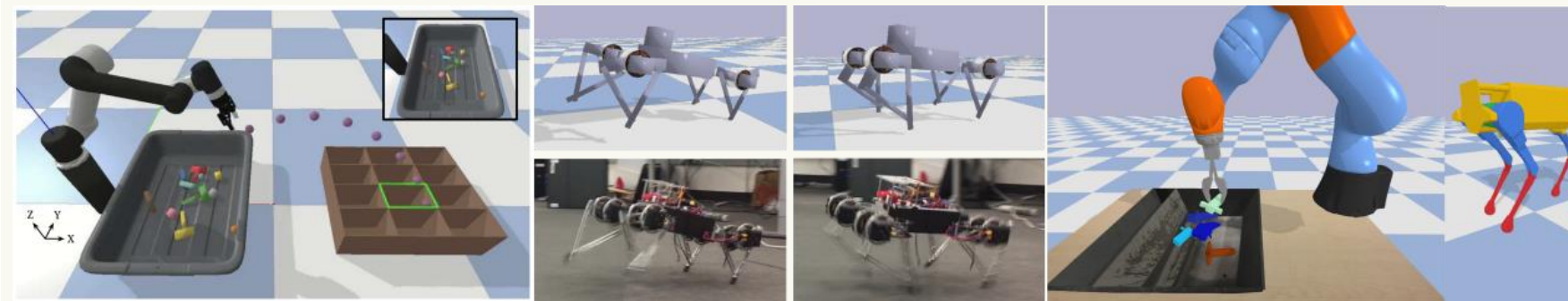


PyBullet Quickstart Guide

[Erwin Coumans](#), [Yunfei Bai](#), 2016-2022

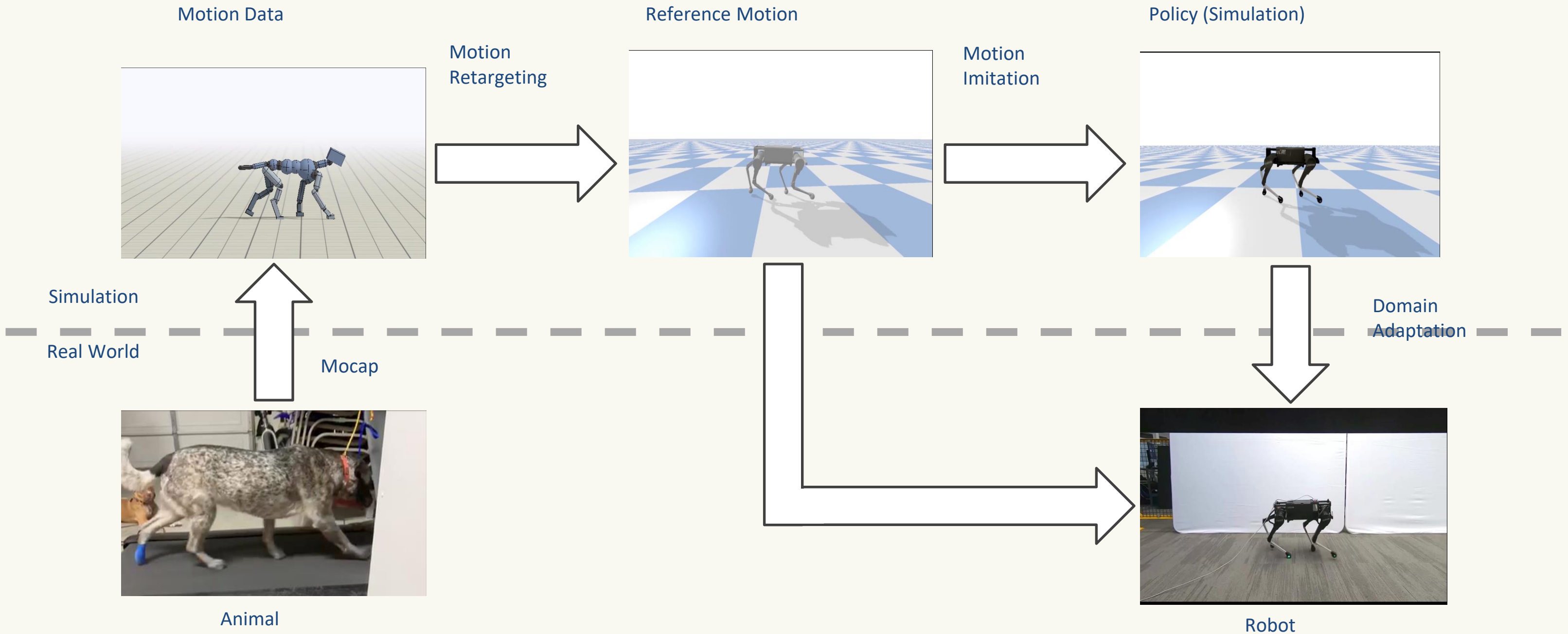
Visit [desktop doc](#), [forums](#), github [discussions](#) and star [Bullet!](#)

Introduction	2	Synthetic Camera Rendering	48
Hello PyBullet World	3	computeView/ProjectionMatrix	48
connect, disconnect, bullet_client	3	getCameraImage	49
setGravity	7	getVisualShapeData	52
loadURDF, loadSDF, loadMJCF	8	changeVisualShape, loadTexture	53
saveState, saveBullet, restoreState	11	Collision Detection Queries	53
createCollisionShape/VisualShape	12	getOverlappingObjects, getAABB	53
createMultiBody	15	getContactPoints, getClosestPoints	54
stepSimulation, performCollisionDetection	17	rayTest, rayTestBatch	56
getBasePositionAndOrientation	18	getCollisionShapeData	57
resetBasePositionAndOrientation	19	Enable/Disable Collisions	58
Transforms: Position and Orientation	19	Inverse Dynamics, Kinematics	61
Controlling a robot	22	calculateInverseDynamics(2)	61
Base, Joints, Links	22	calculateJacobian, MassMatrix	61
getNumJoints, getJointInfo	22	calculateInverseKinematics(2)	63
setJointMotorControl2/Array	24	Reinforcement Learning Gym Envs	66
getJointState(s), resetJointState	28	Environments and Data	66
enableJointForceTorqueSensor	30	Stable Baselines & ARS, ES,...	70
getLinkState(s)	31	Virtual Reality	73
getBaseVelocity, resetBaseVelocity	33	getVREvents, setVRCameraState	73
applyExternalForce/Torque	33	Debug GUI, Lines, Text, Parameters	75
getNumBodies, getBodyInfo, getBodyUniqueId, removeBody	34	addUserDebugLine, Points, Text, Parameter	75
createConstraint, removeConstraint, changeConstraint	34	addUserData (get, sync, remove)	79
getNumConstraints, getConstraintUniqueId	36	configureDebugVisualizer	80
getConstraintInfo/State	36	get/resetDebugVisualizerCamera	80
getDynamicsInfo/changeDynamics	37	getKeyboardEvents, getMouseEvents	82
setTimeStep	40	Plugins	83
setPhysicsEngineParameter	40	loadPlugin, executePluginCommand	83
resetSimulation	43	Build and install PyBullet	84
startStateLogging/stopStateLogging	43	Support, Tips, Citation	87
Deformables and Cloth (FEM, PBD)	45		
loadSoftBody/loadURDF	46		
createSoftBodyAnchor	47		



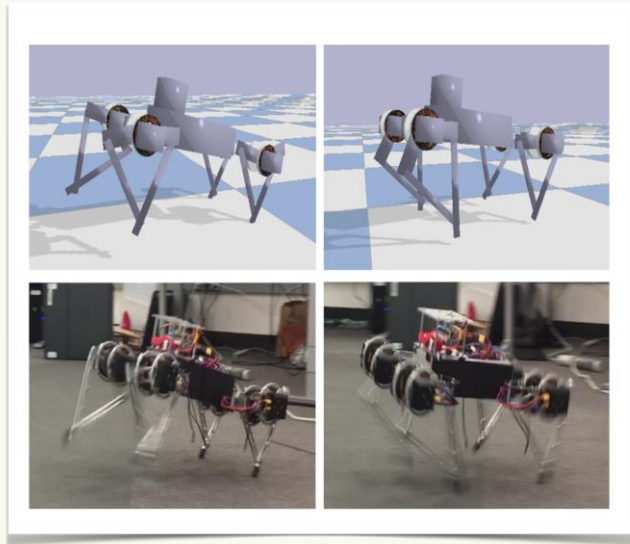
<http://pybullet.org> and <http://wiki.ros.org/urdf/XML>

Learning Agile Robotic Locomotion Skills

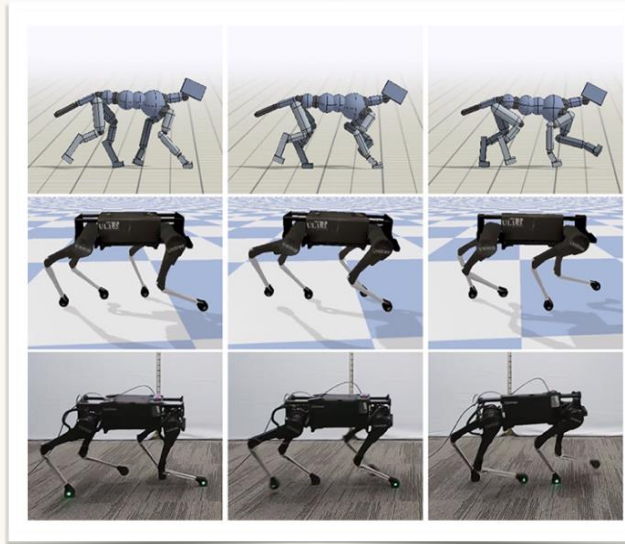


https://xbpeng.github.io/projects/Robotic_Imitation/index.html

Motivation for fast Sim for Reinforcement Learning



Sim-to-Real: Learning Agile Locomotion For Quadruped Robots (2018, Jie Tan et al)



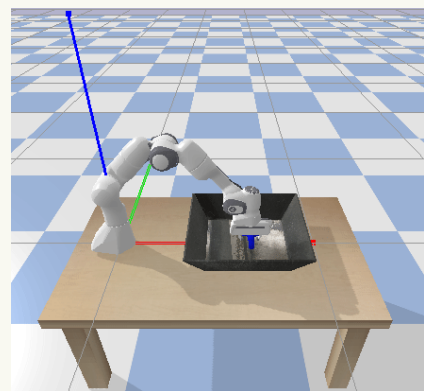
Learning Agile Robotic Locomotion Skills by Imitating Animals (2020, Jason Peng et al)



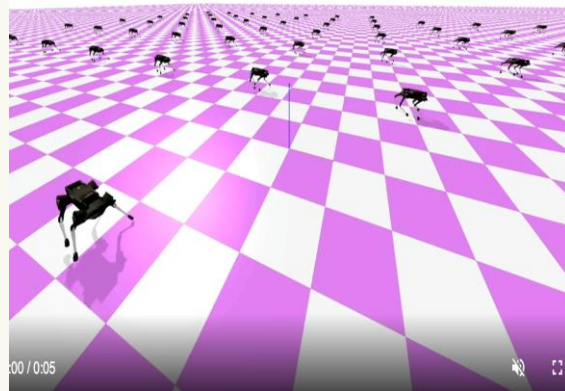
Inferring Articulated Rigid Body Dynamics from RGBD Video (2022, Eric Heiden et al)



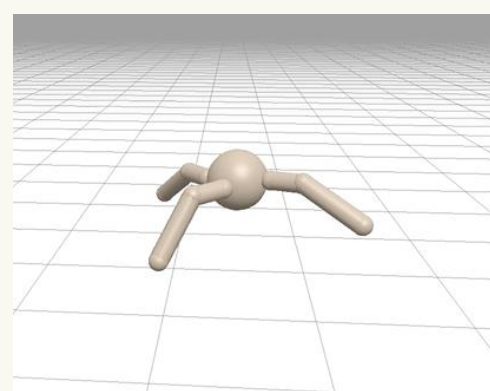
Advanced Skills through Multiple Adversarial Motion Priors in Reinforcement Learning (2022, ETH/NVIDIA)



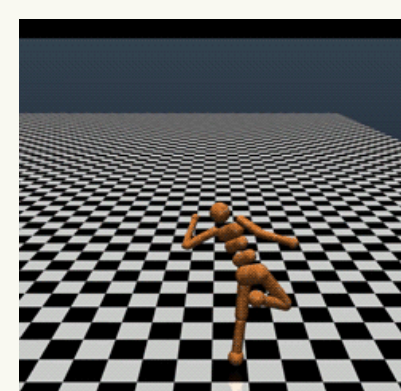
PyBullet



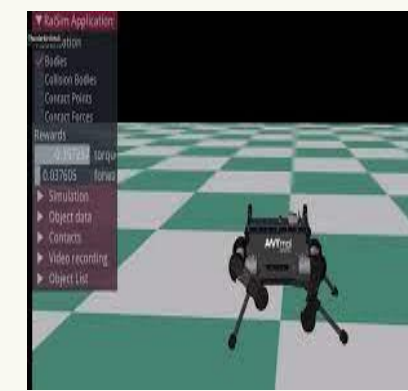
Tiny Differentiable Simulator



Brax



MuJoCo

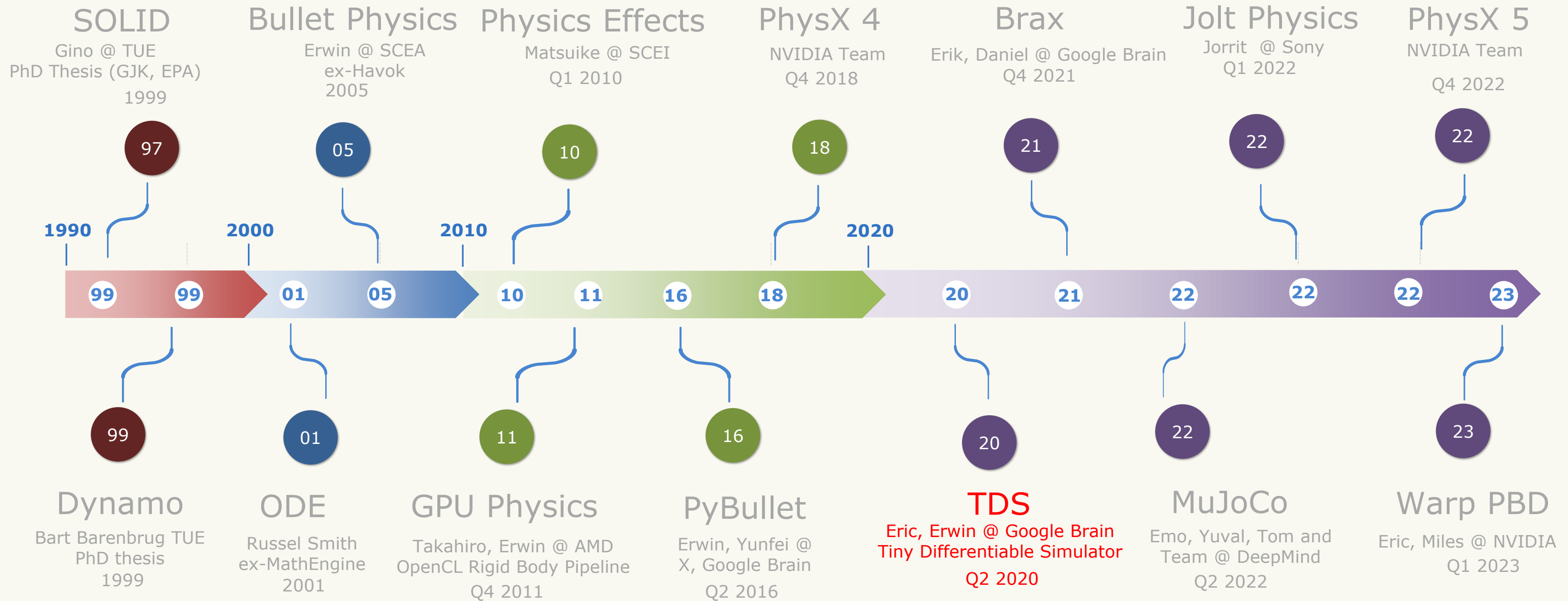


RaiSim

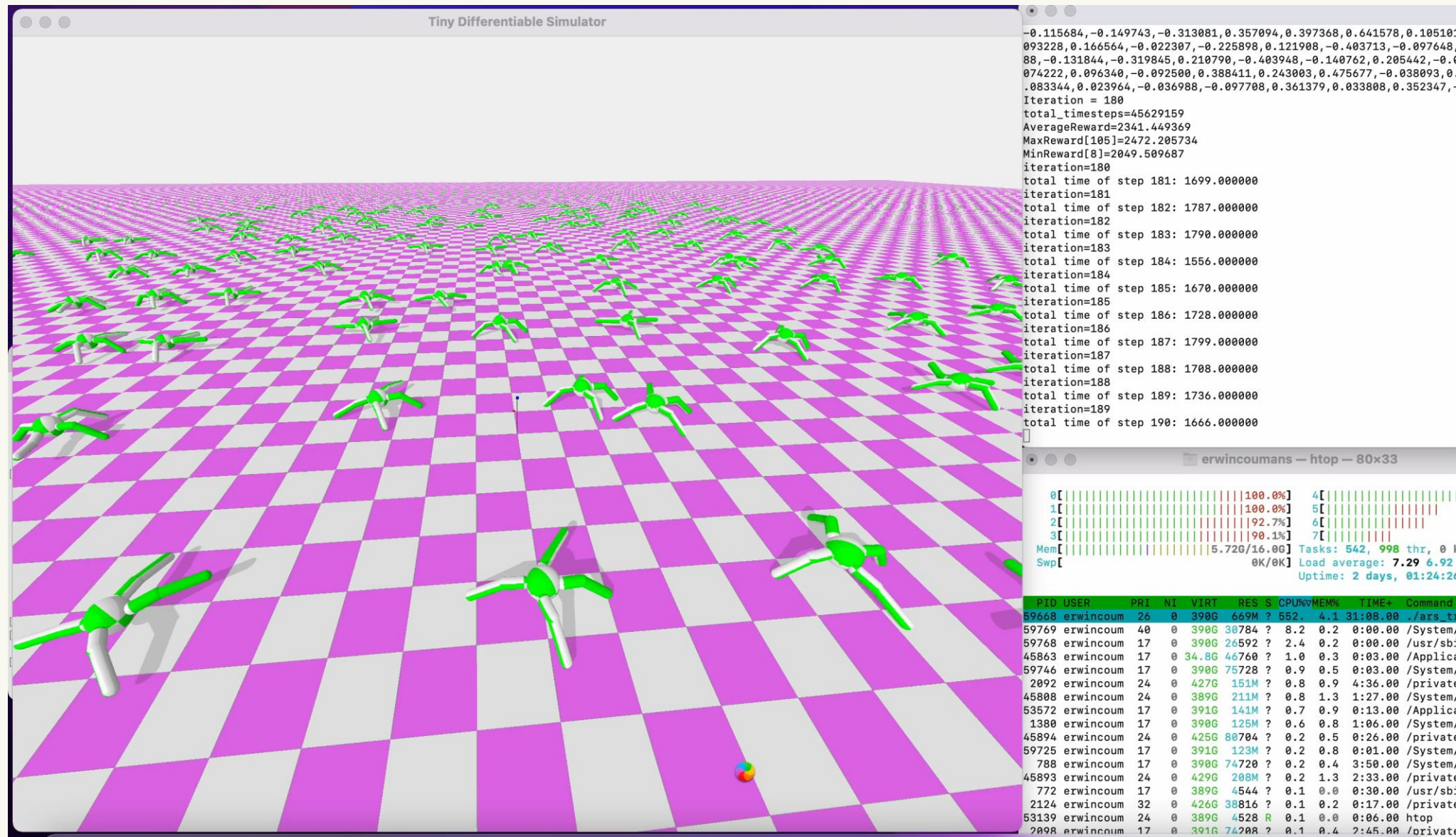


Omniverse Isaac Gym
PhysX/WARP

Tiny Differentiable Simulator



Tiny Differentiable Simulator



<https://github.com/erwincoumans/tiny-differentiable-simulator>

NeuralSim

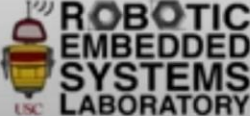
NeuralSim (ICRA 2021)



NeuralSim: Augmenting Differentiable Simulators with Neural Networks

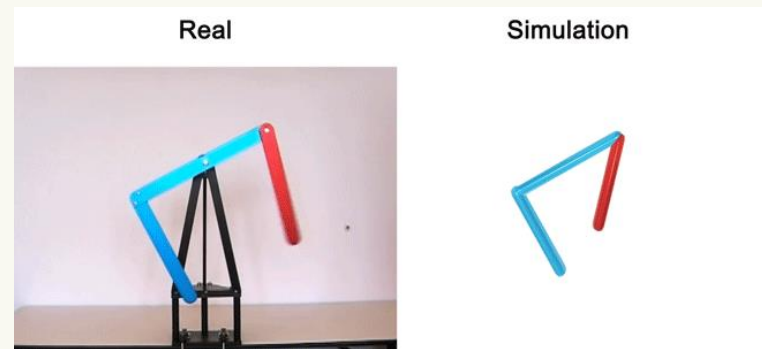
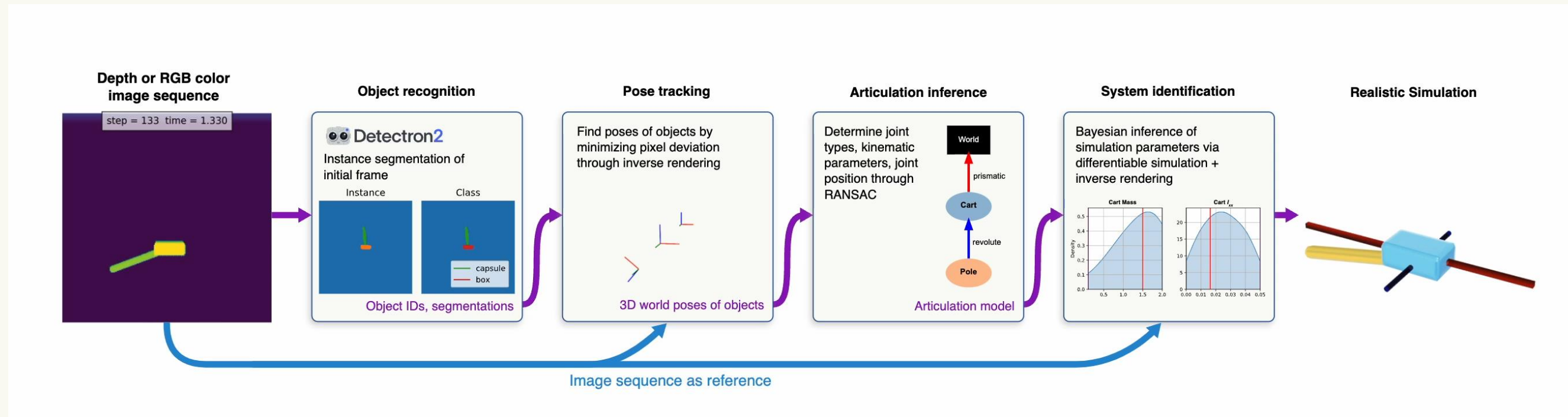
Eric Heiden*, David Millard*, Erwin Coumans,
Yizhou Sheng, Gaurav S. Sukhatme

 github.com/google-research/tiny-differentiable-simulator

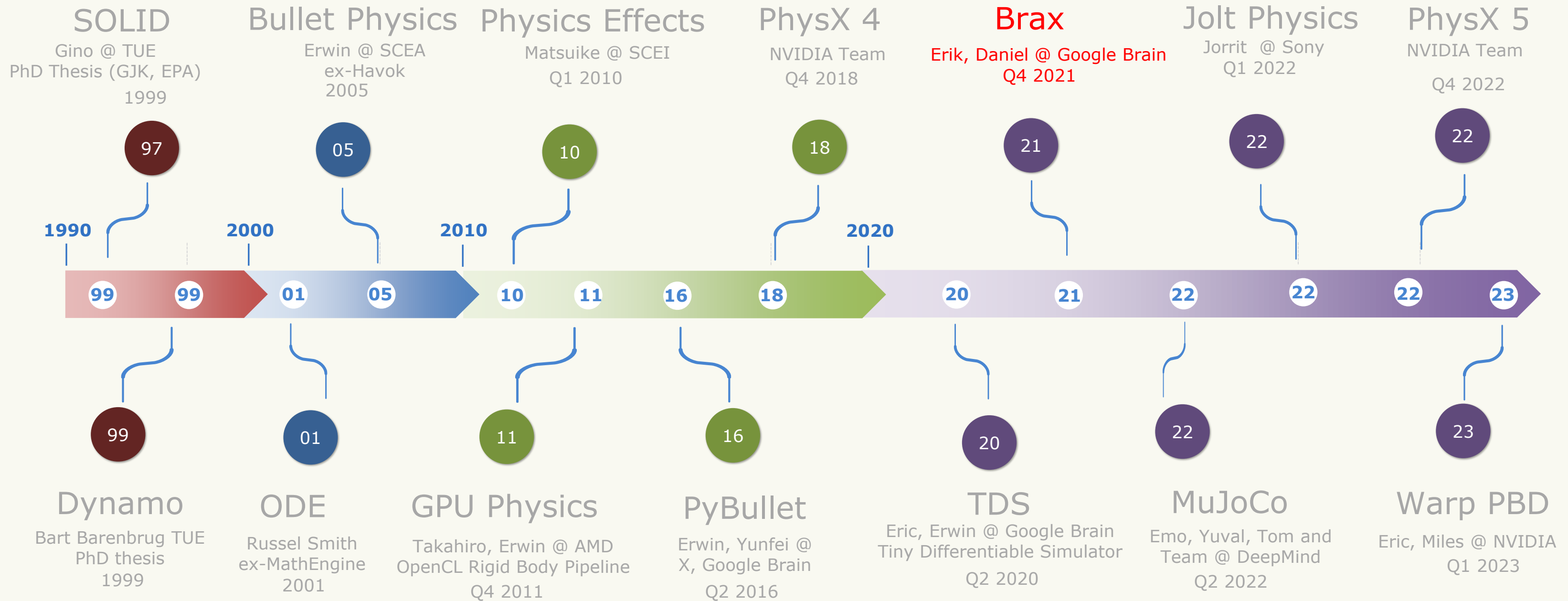
<https://arxiv.org/pdf/2011.04217.pdf>

Pixels to URDF / USD

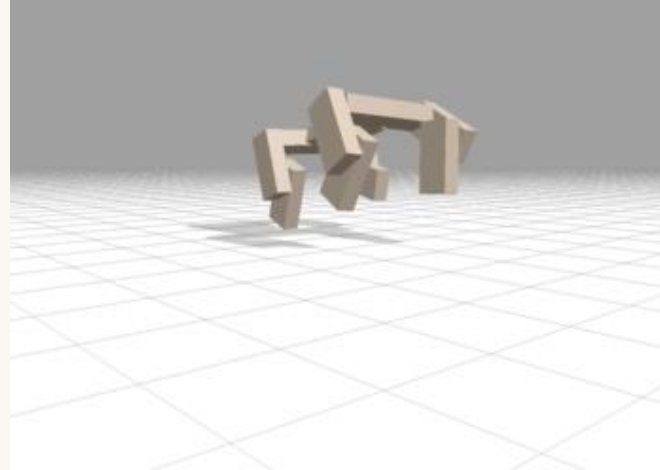
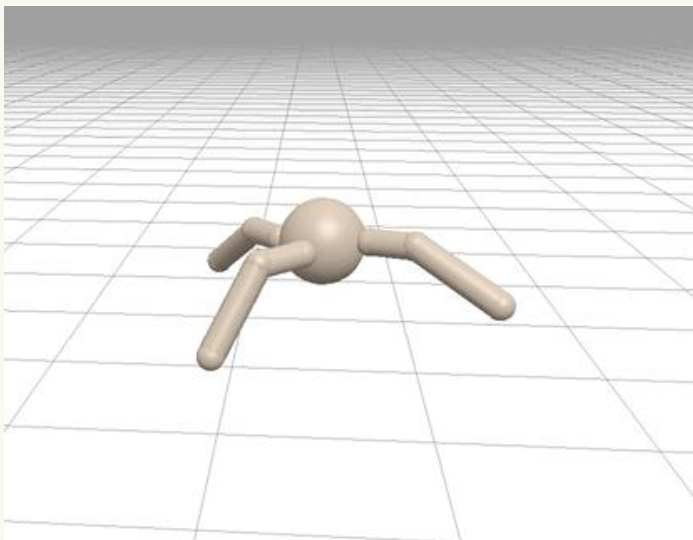
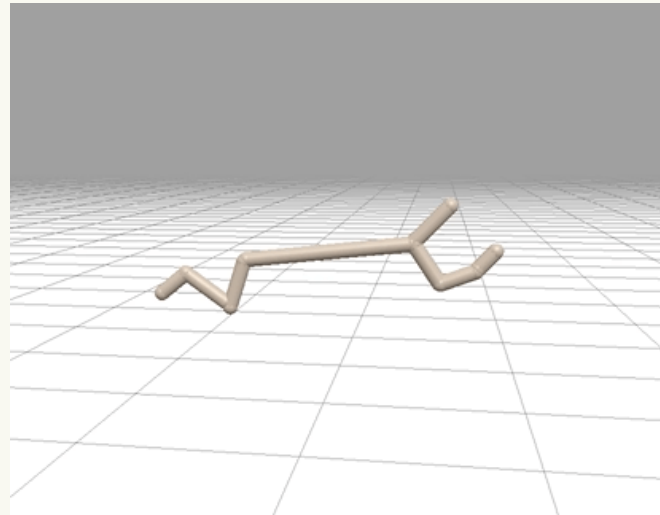
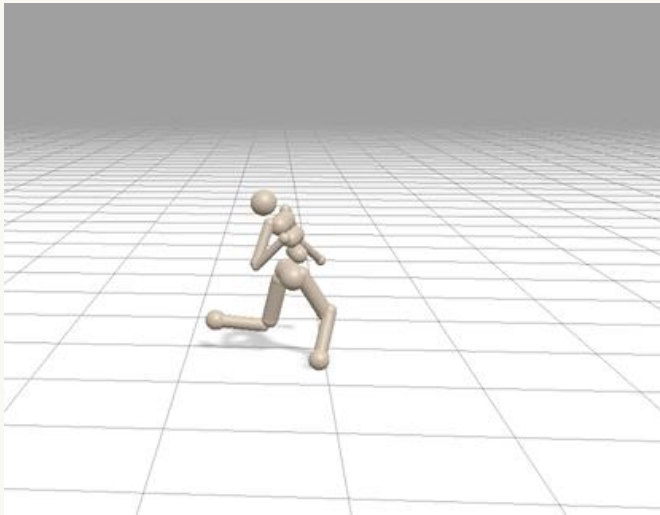


<https://eric-heiden.com/publication/2022-video2sim-iros>

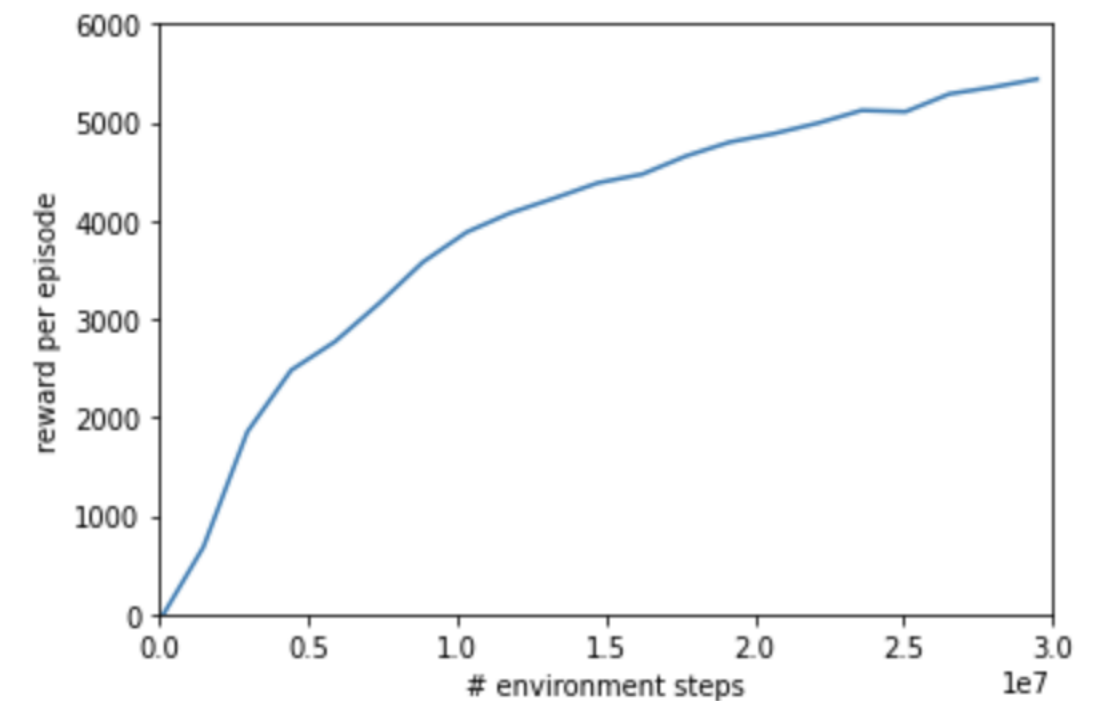
Brax



Brax



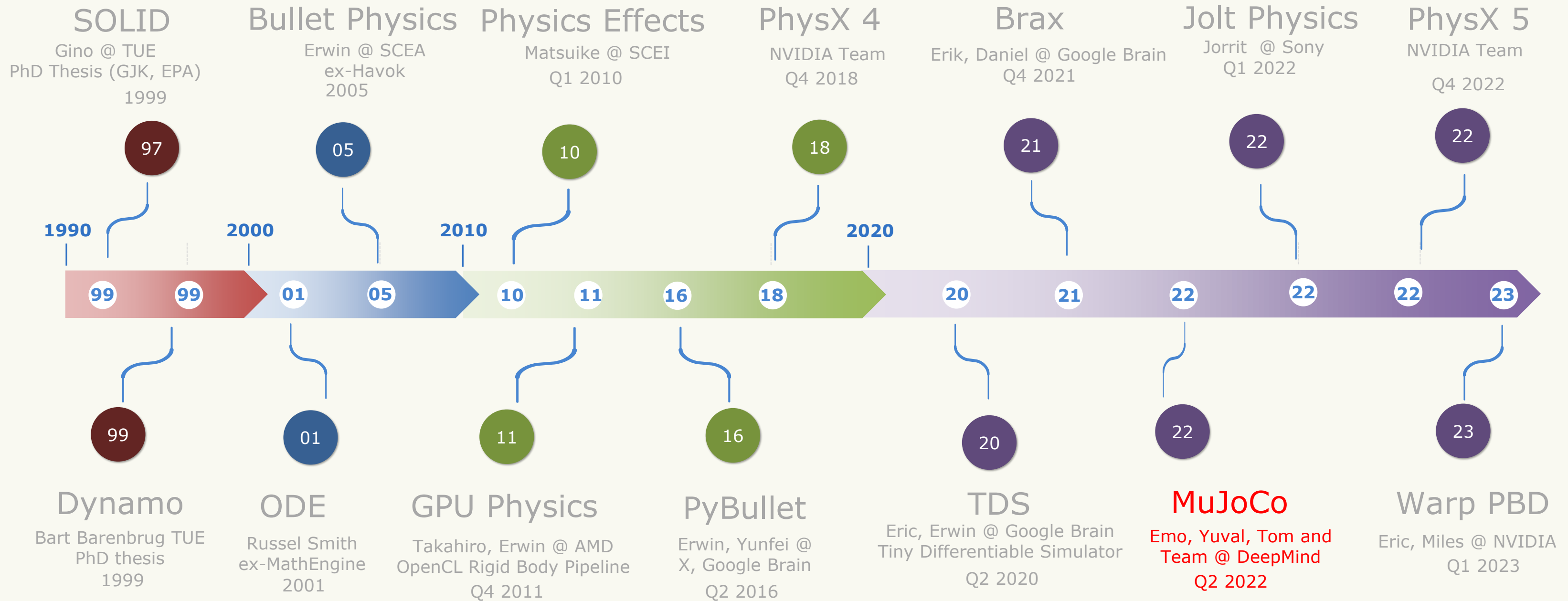
```
91  
92 print(f'time to jit: {times[1] - times[0]}')  
93 print(f'time to train: {times[-1] - times[1]}')
```



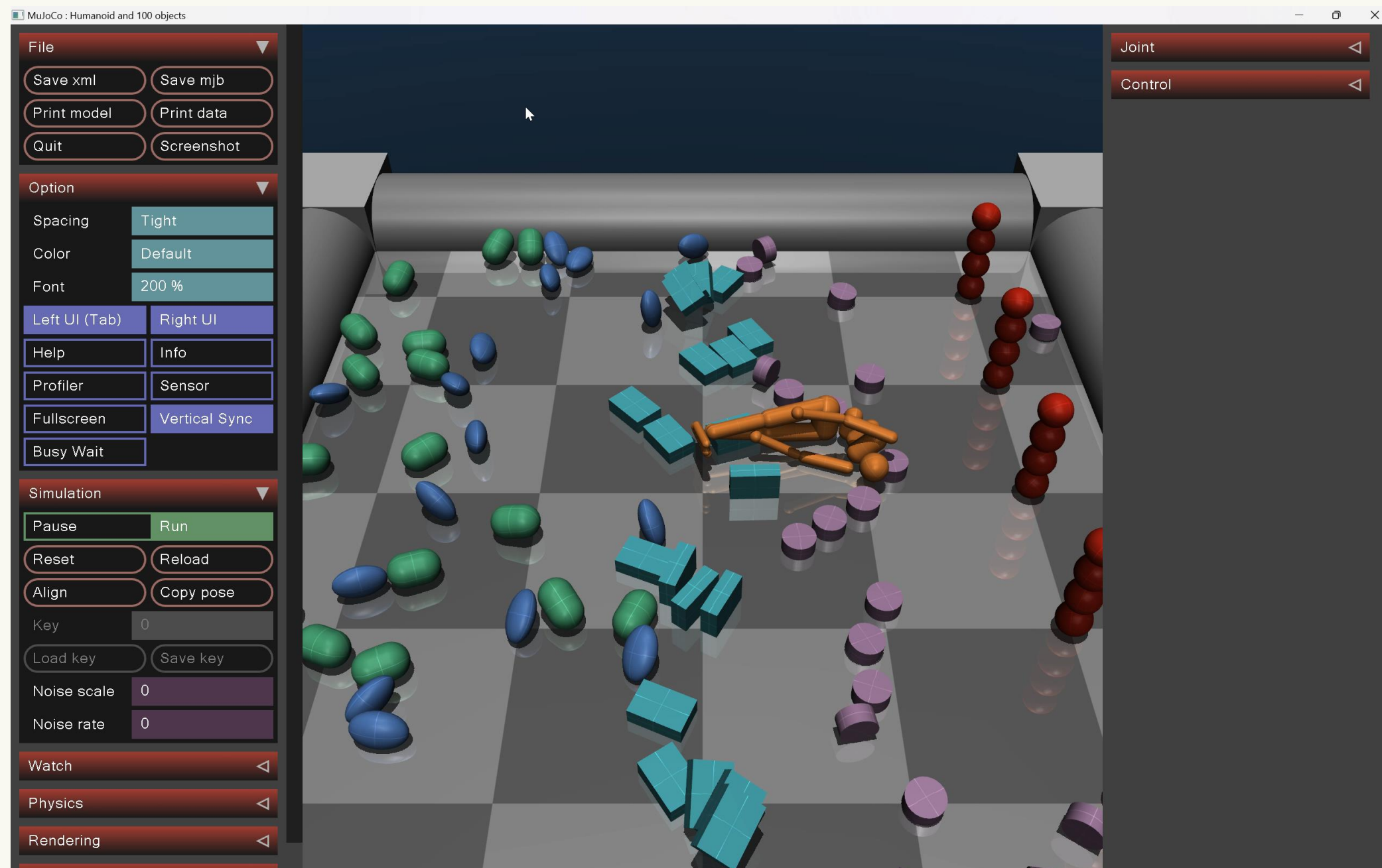
```
time to jit: 0:00:58.711923  
time to train: 0:01:47.832345
```

<https://github.com/google/brax>

MuJoCo

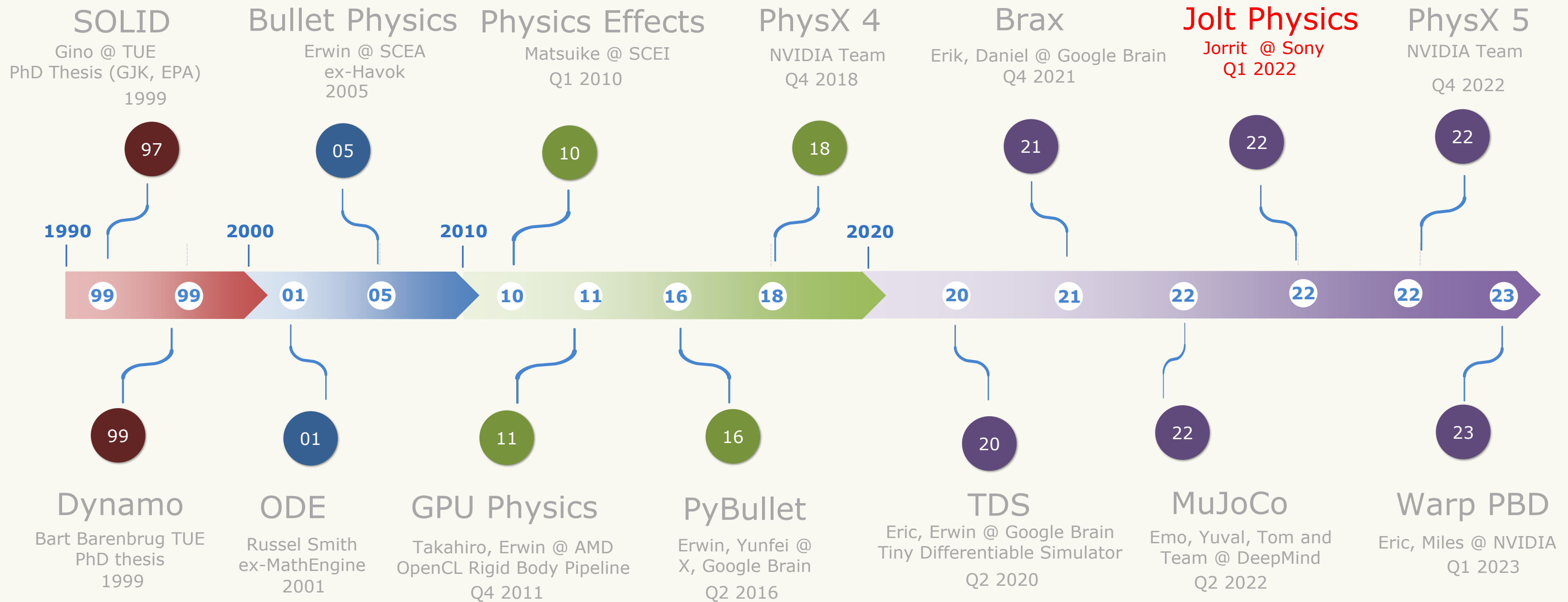


MuJoCo

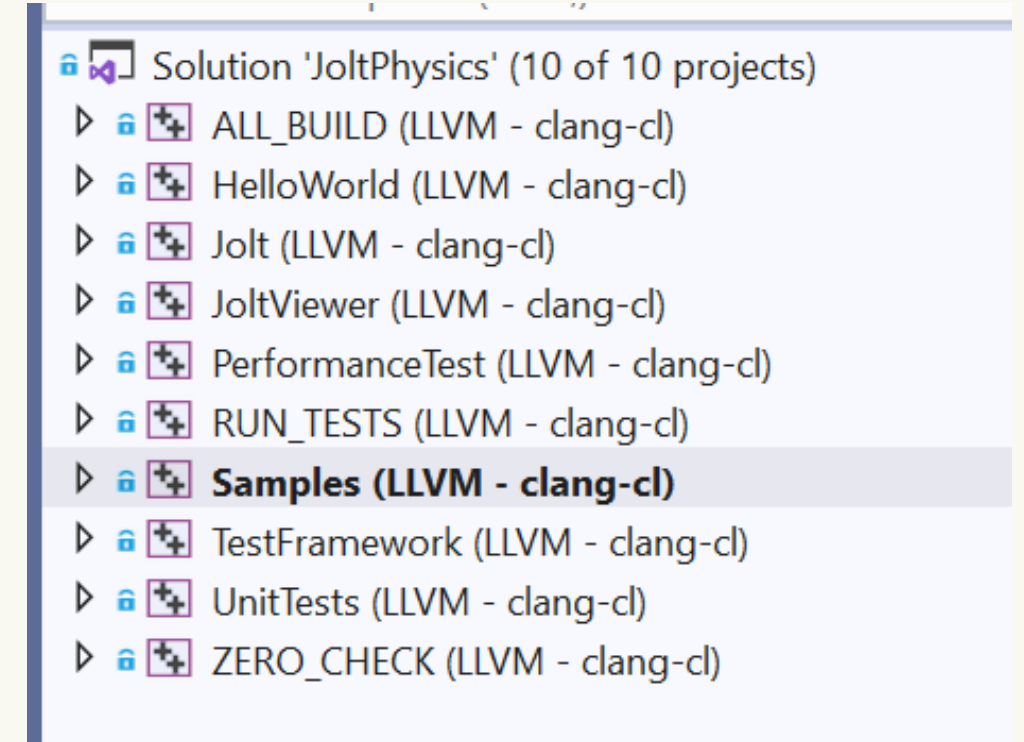


<https://github.com/deepmind/mujoco>

Jolt Physics

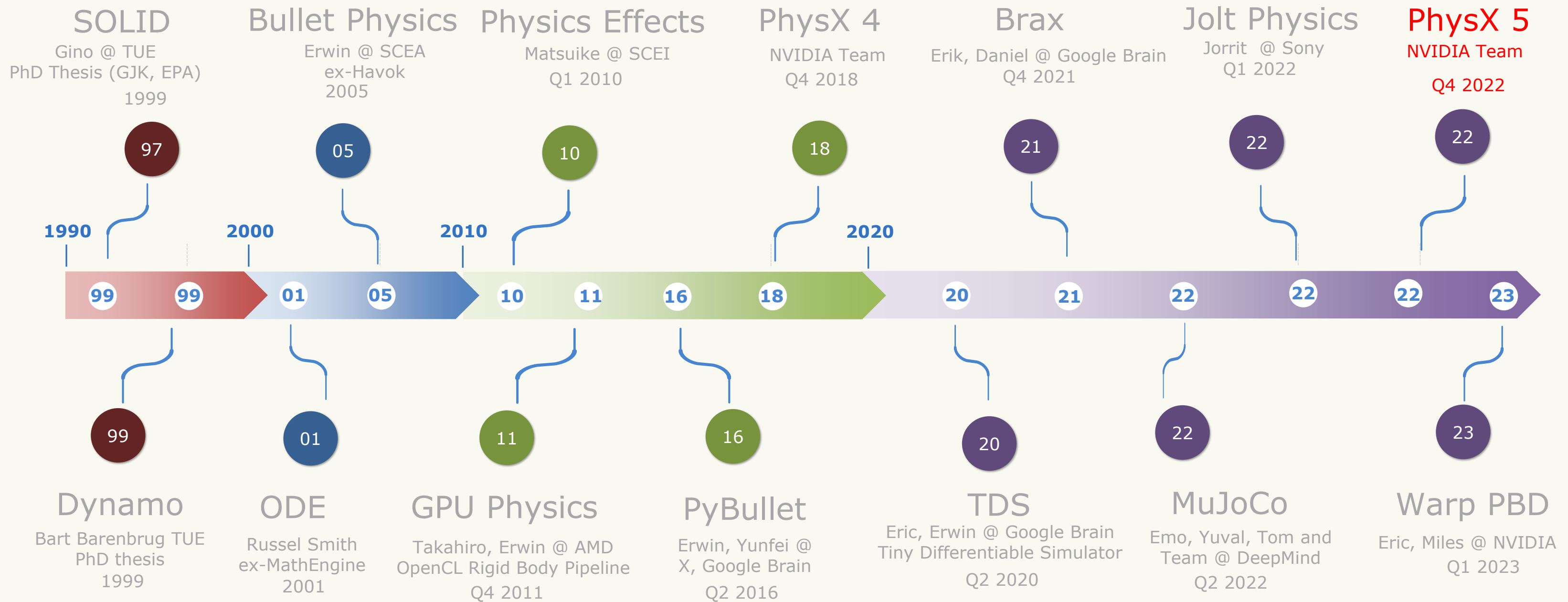


Jolt Physics

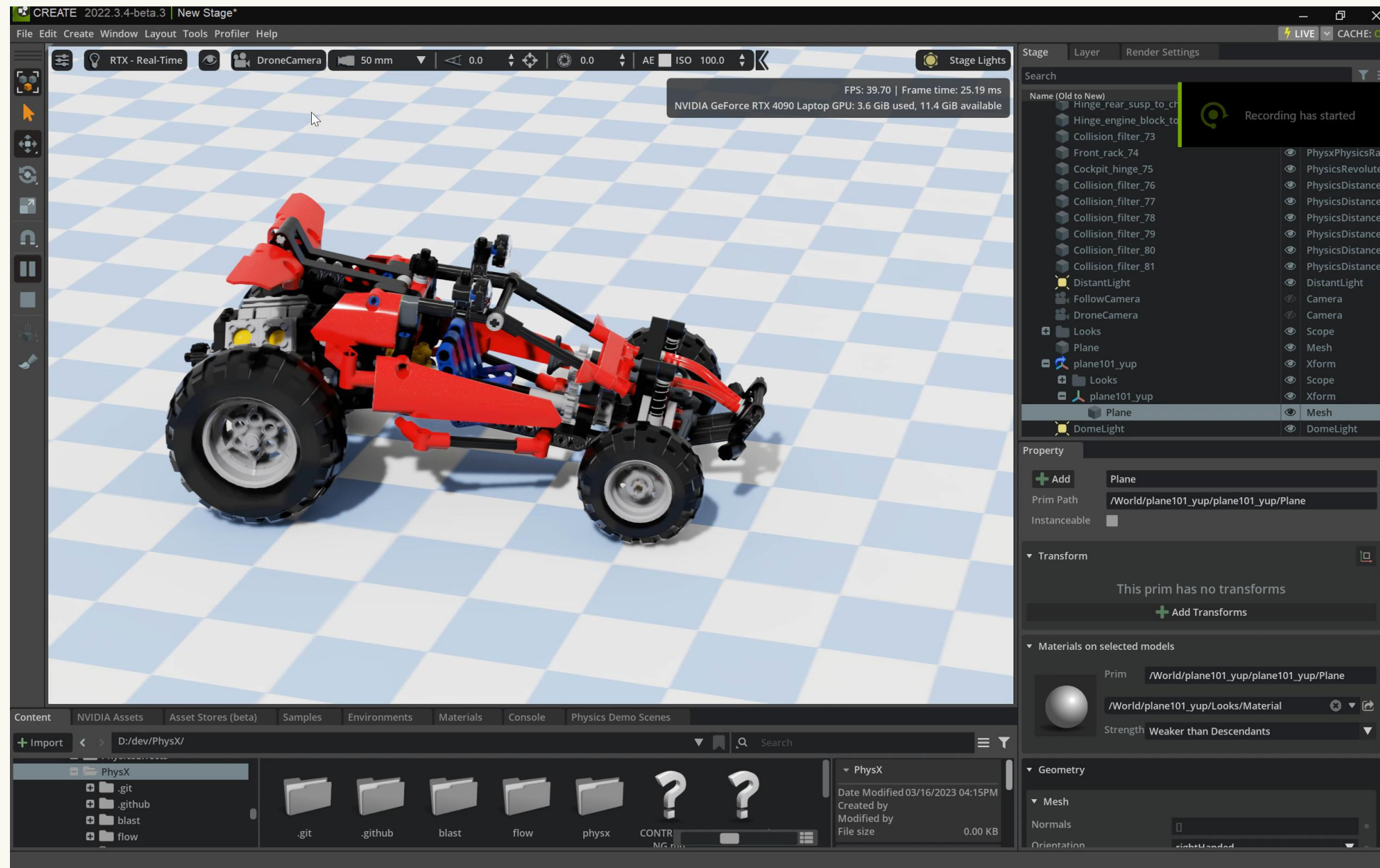


<https://github.com/jrouwe/JoltPhysics>

PhysX 5

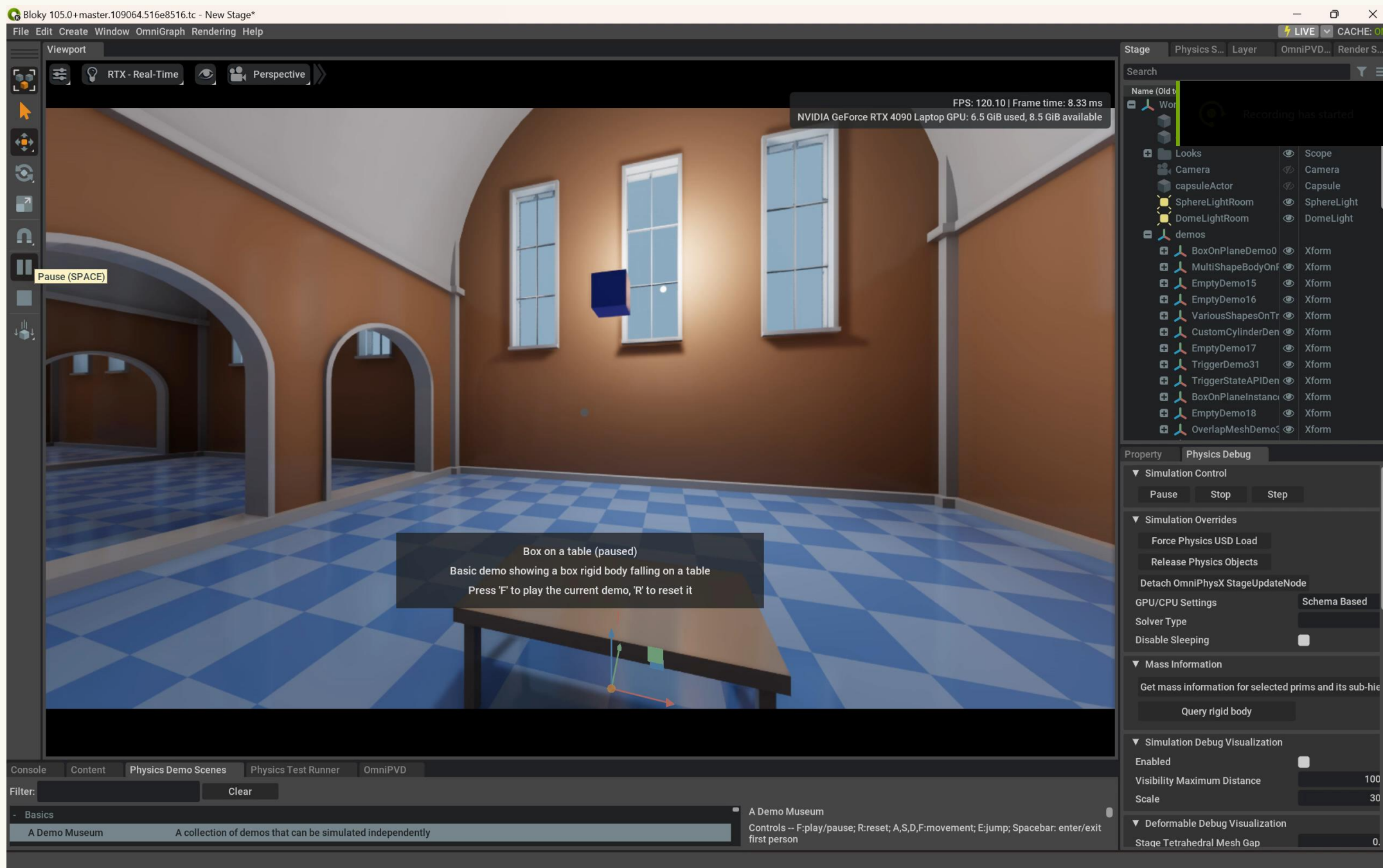


PhysX 5 in Omniverse



<https://github.com/NVIDIA-Omniverse/PhysX>

PhysX 5 in Omniverse



PhysX, USD, MDL in Omniverse



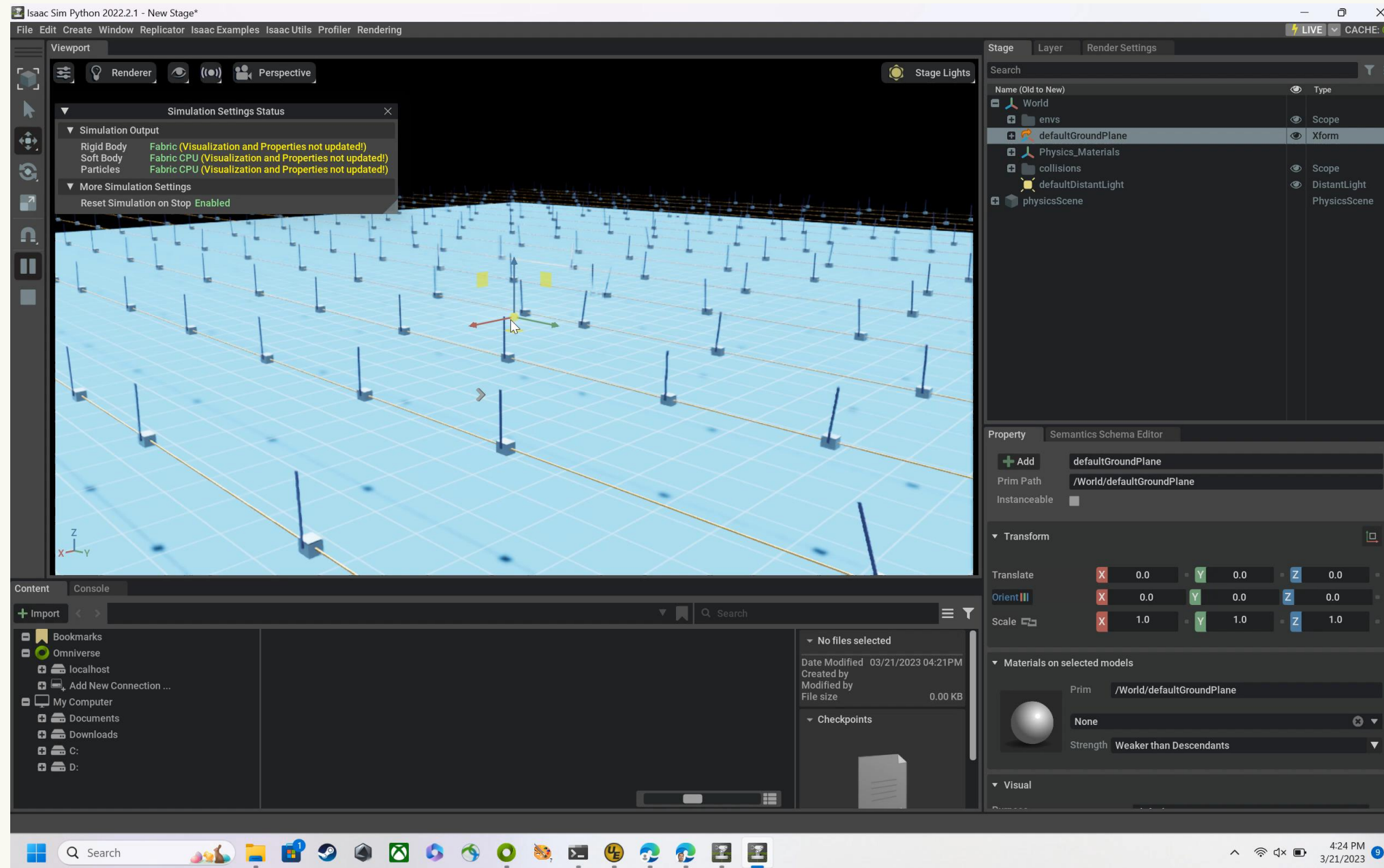
Open Standards



Top Industry Tools

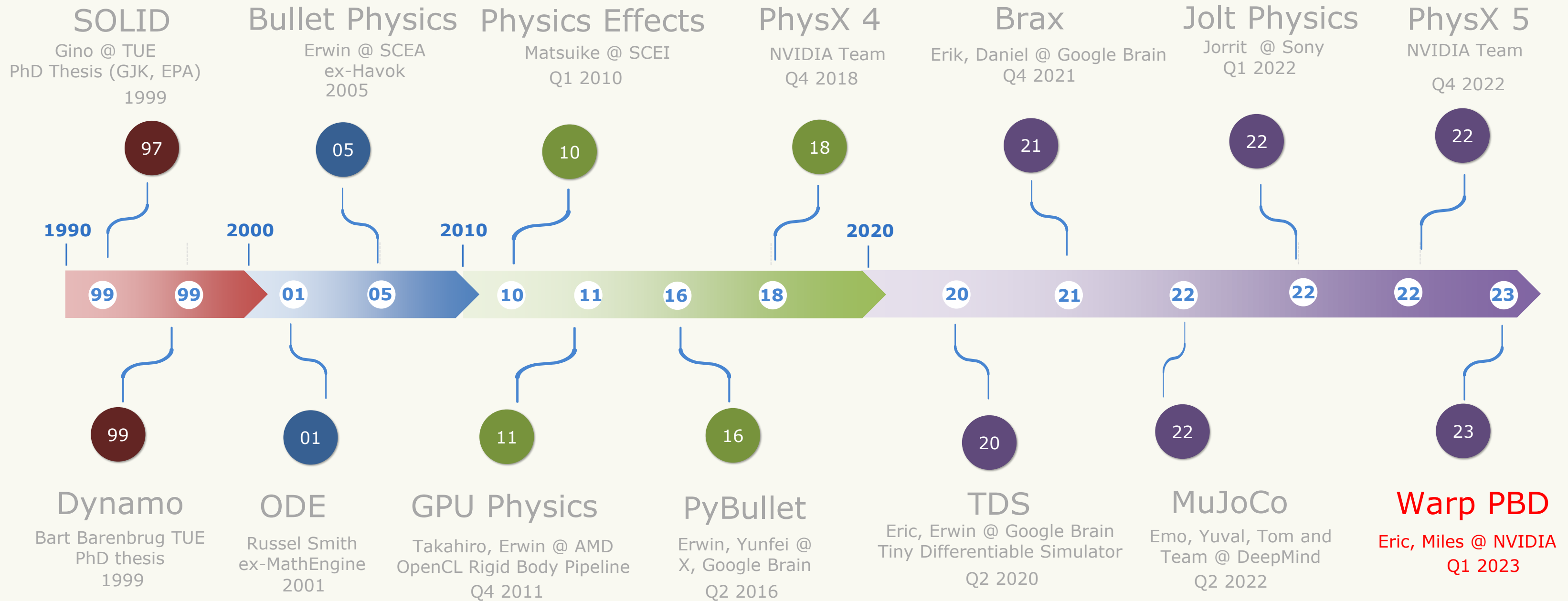
https://openusd.org/release/api/usd_physics_page_front.html

OmniIsaacGymEnvs RL training



<https://github.com/NVIDIA-Omniverse/OmniIsaacGymEnvs>

Warp (pbd)



WARP

Python-based CUDA Kernel JIT - “Shadertoy for simulation”

Warp Goals

1. Make it easy to write GPU graphics + simulation code from Python
2. Lower barrier to entry, faster iteration time
3. Rich built-in math, geometry library
4. Simple interop with existing C++/CUDA code
5. Fully differentiable

```
import warp as wp

@wp.kernel
def calculate_reward_wp(obs_buf: wp.array(dtype=float),
                       rew_buf: wp.array(dtype=float),
                       reset_buf: wp.array(dtype=int),
                       timeout_buf: wp.array(dtype=int),
                       progress_buf: wp.array(dtype=int),
                       termination_height: float,
                       num_obs: int,
                       max_episode_length: int):

    tid = wp.tid()

    timeout_buf[tid] = 0
    reset_buf[tid] = 0
    progress_buf[tid] = progress_buf[tid] + 1

    up_reward = 0.1 * obs_buf[tid * num_obs + 27]
    heading_reward = obs_buf[tid * num_obs + 28]
    height_reward = obs_buf[tid * num_obs] - termination_height
    progress_reward = obs_buf[tid * num_obs + 5]

    rew_buf[tid] = progress_reward + up_reward + heading_reward + height_reward

    # reset agents
    if progress_buf[tid] >= max_episode_length:
        reset_buf[tid] = 1
        timeout_buf[tid] = 1

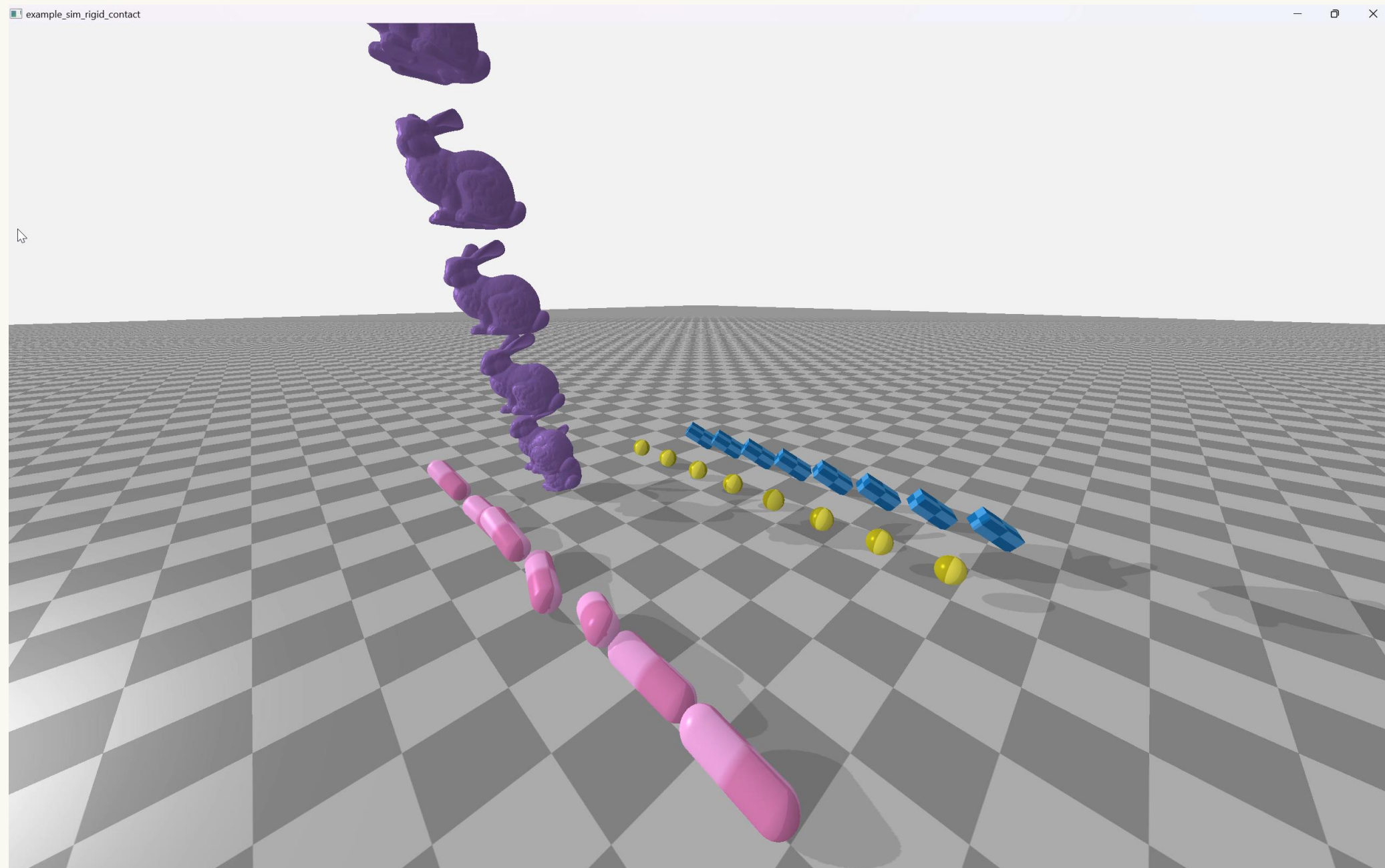
    if height_reward < 0.0:
        reset_buf[tid] = 1

    if reset_buf[tid] == 1:
        progress_buf[tid] = 0
```

Ant RL Reward Function in Warp - torch.where() begone!

Warp: A High-performance Python Framework for GPU Simulation and Graphics

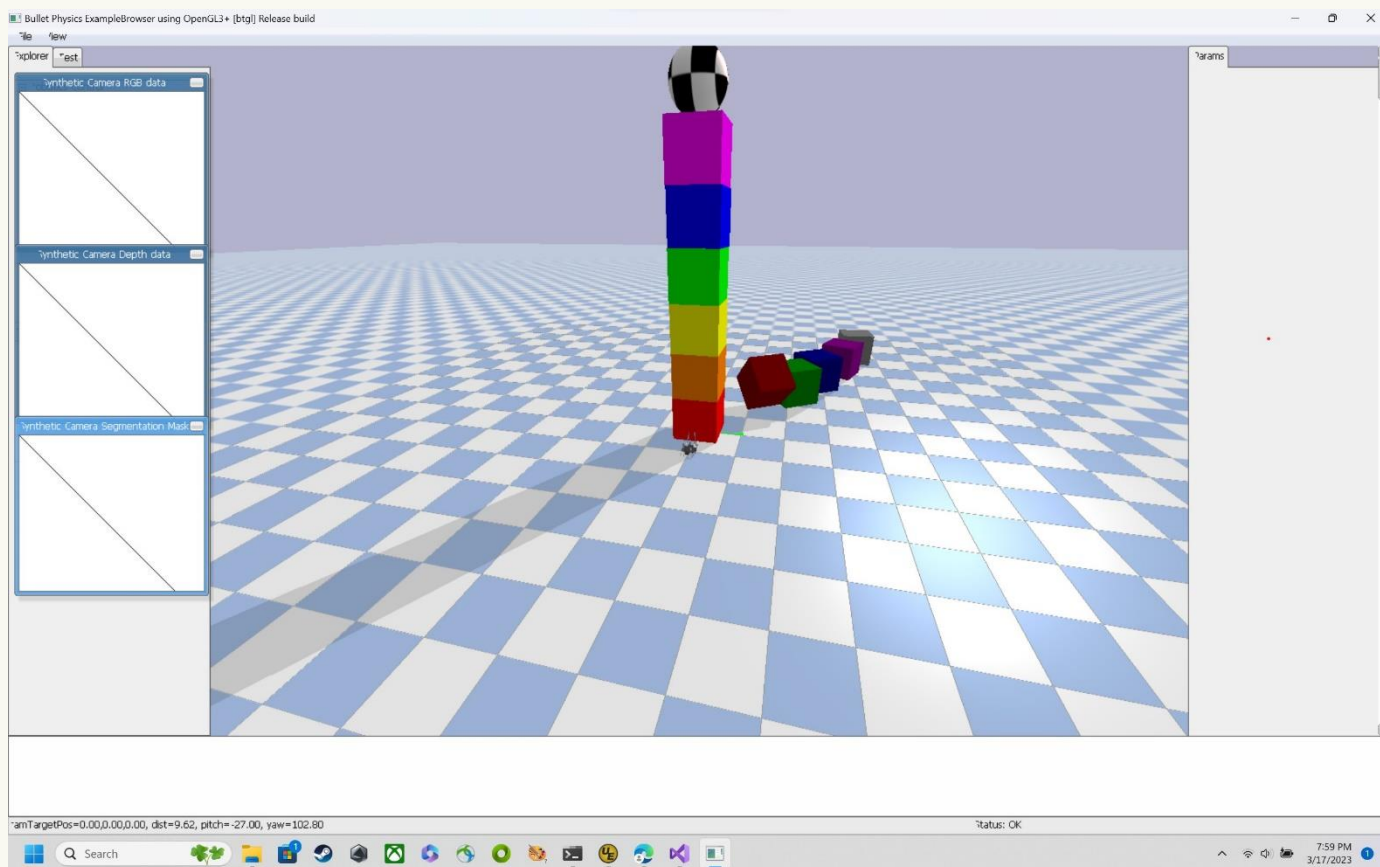
Warp



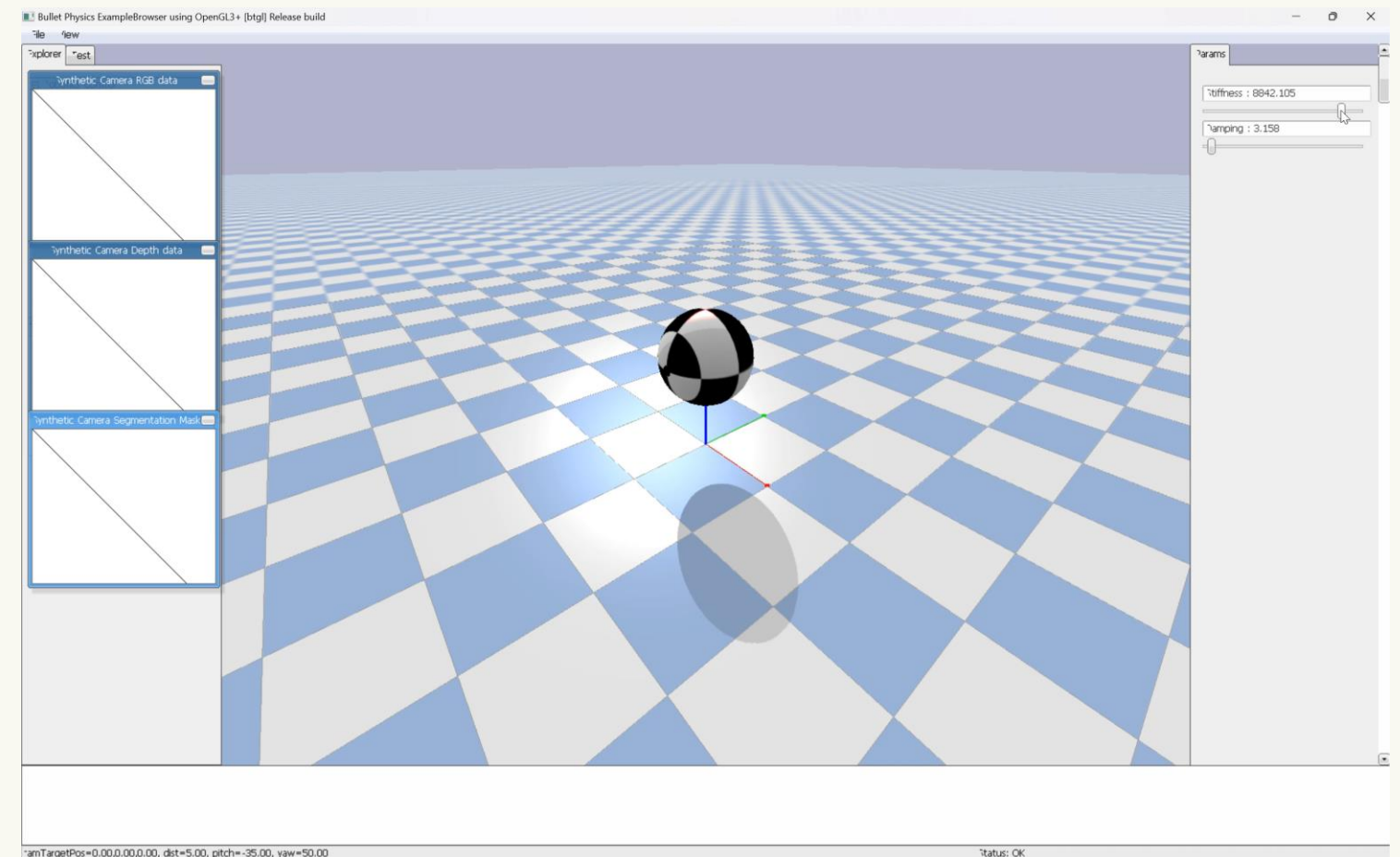
<https://github.com/NVIDIA/warp> (and <https://github.com/eric-heiden/warp>)

ChatGPT

"can you create a pybullet script with a ground plane, a box and a sphere on top."+"can you add 10 more boxes on top?"+"can you move the 5th box 0.3 units along the x axis?"+"can you add a quadruped robot next to the boxes?"

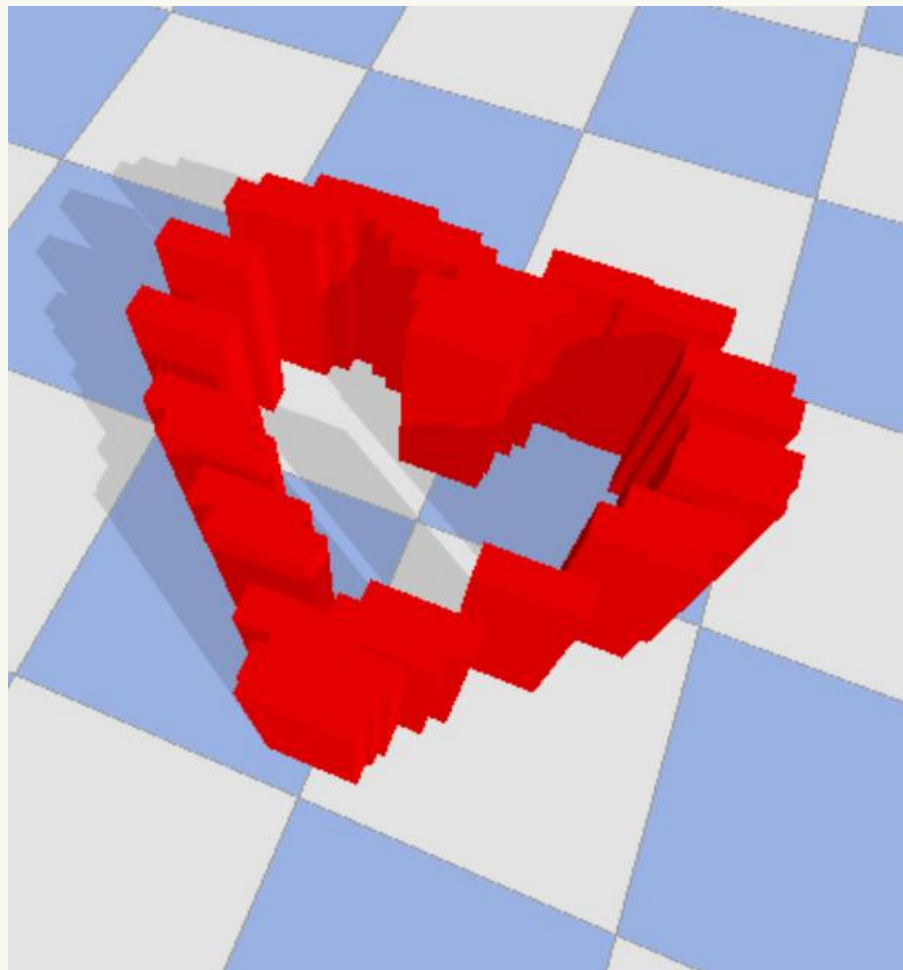


“create a pybullet script with a sphere falling on a ground plane. Add a slider to control contactStiffness and contactDamping”



ChatGPT PyBullet

“Place 100 dominoes in a shape of a heart”



```
1 import pybullet as p
2 import time
3 import math
4
5 # Define the size and position of the dominoes.
6 domino_width = 0.3
7 domino_height = 1.5
8 domino_depth = 0.1
9 spacing = 0.05
10 domino_mass = 0.1
11 domino_friction = 0.5
12 domino_restitution = 0.1
13 start_pos = [-4, 0, 0]
14
15 # Create a function to add the dominoes to the simulation.
16 def add_domino(x, y, z):
17     boxId = p.createCollisionShape(p.GEOM_BOX, halfExtents=[domino_width/2, domino_depth/2, domino_height/2])
18     boxVisualId = p.createVisualShape(p.GEOM_BOX, halfExtents=[domino_width/2, domino_depth/2, domino_height/2], rgbColor=[1,0,0,1])
19     dominoId = p.createMultiBody(domino_mass, boxId, boxVisualId, [x, y, z])
20     p.changeDynamics(dominoId, -1, lateralFriction=domino_friction, restitution=domino_restitution)
21     return dominoId
22
23 # Define the positions of the dominoes in the shape of a heart.
24 heart_pos = []
25 for t in range(0, 360, 10):
26     x = 16 * math.sin(t) ** 3
27     y = -13 * math.cos(t) + 5 * math.cos(2*t) + 2 * math.cos(3*t) + math.cos(4*t)
28     heart_pos.append([start_pos[0] + x * spacing, start_pos[1] + y * spacing, start_pos[2]])
29
30 # Create a simulation environment and set the gravity.
31 p.connect(p.GUI)
32 p.setGravity(0, 0, -10)
33
34 # Create a ground plane.
35 planeId = p.createCollisionShape(p.GEOM_PLANE)
36 p.createMultiBody(0, planeId)
37
38 # Add the dominoes to the simulation.
39 for pos in heart_pos:
40     add_domino(pos[0], pos[1], pos[2])
41
42 # Run the simulation.
43 for i in range(10000):
44     p.stepSimulation()
45     time.sleep(1./240.)
46
47 # Disconnect the simulation environment.
48 p.disconnect()
```



March 20-24, 2023
San Francisco, CA

Thank You! Questions?

<https://twitter.com/erwincoumans>

Don't forget about the speaker survey.

#GDC23