taichi-dev / **taichi** Public

Productive & portable high-performance programming in Python.

taichi.graphics

MIT License

18k stars 1.9k forks

Star Notifications

Code Issues 395 Pull requests 48 Discussions Actions Projects 9 Security

master Go to file

knight42 … ● 1 hour ago

View code

README.md

Taichi Programming Language

Latest Release v0.8.11 | downloads 455k | Build and Test failing | Docker Image passing | Python Coverage 43%

```
import taichi as ti
```

- Getting Started
  - Installation
  - Your first Taichi program
- Documentation
- Contacts

Taichi (太极) is an open-source, imperative, parallel programming language for high-performance numerical computation. It is embedded in Python and uses just-in-time (JIT) compiler frameworks (e.g. LLVM) to offload compute-intensive Python code to the native GPU or CPU instructions.

Advantages of Taichi:

- Built around Python: Taichi shares almost the same syntax with Python, allowing you to write algorithms with minimal language barrier. It is also well integrated into the Python ecosystem, such as NumPy and PyTorch.
- Flexibility: Taichi provides a set of generic data containers known as *SNode* (/ˈsnoʊd/), an effective mechanism for composing hierarchical, multi-dimensional fields. This can cover many use patterns in numerical simulation (e.g. spatially sparse computing).
- Performance: Through the `@ti.kernel` decorator, Taichi's JIT compiler automatically compiles your Python functions into efficient GPU or CPU machine code for parallel execution.
- Portability: Write your code once and run it everywhere. Currently, Taichi supports most mainstream GPU APIs, such as CUDA and Vulkan.
- ... and many more features! A cross-platform, Vulkan-based 3D visualizer, differentiable programming, quantized computation (experimental), etc.

Getting Started

Installation

You can easily install Taichi with Python's package installer `pip`:

```
pip install taichi
```

If you want to try out the latest features, we also provide a nightly package:

```
pip install -i https://test.pypi.org/simple/ taichi-nightly
```

The nightly package can and will break from time to time!

Supported environments

- Operating systems
  - Windows[1]
  - Linux
  - macOS
- Python: 3.6 ~ 3.9 (64-bit only)
- Compute backends
  - x64/ARM CPUs
  - CUDA
  - Vulkan
  - OpenGL (4.3+)
  - Apple Metal
  - WebAssembly (experiemental)

[1] On Windows, please install Microsoft Visual C++ Redistributable first.

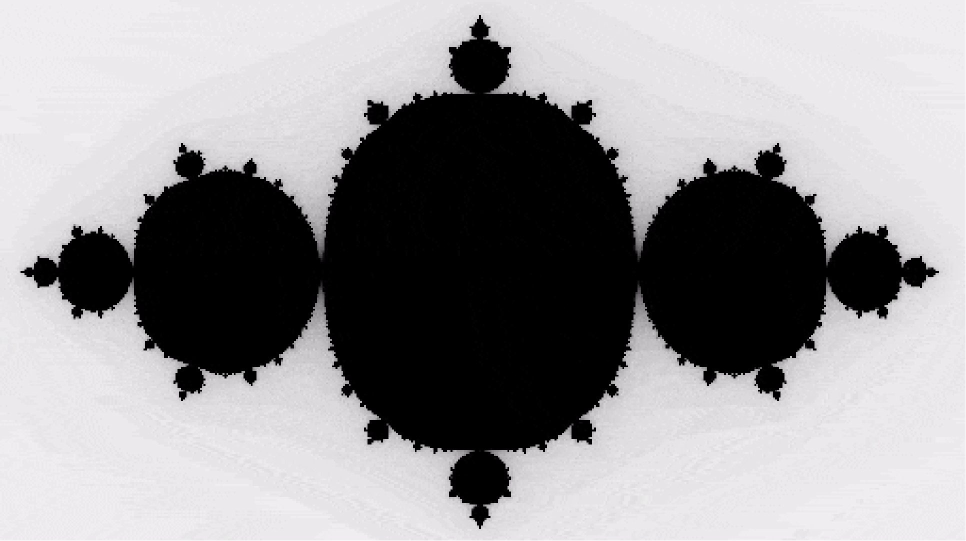
Your first Taichi program

Here's how you can program a 2D fractal in Taichi:

```
# python/taichi/examples/simulation/fractal.py

import taichi as ti

ti.init(arch=ti.gpu)

n = 320
pixels = ti.field(dtype=float, shape=(n * 2, n))


@ti.func
def complex_sqr(z):
    return ti.Vector([z[0]**2 - z[1]**2, z[1] * z[0] * 2])


@ti.kernel
def paint(t: float):
    for i, j in pixels:  # Parallelized over all pixels
        c = ti.Vector([-0.8, ti.cos(t) * 0.2])
        z = ti.Vector([i / n - 1, j / n - 0.5]) * 2
        iterations = 0
        while z.norm() < 20 and iterations < 50:
            z = complex_sqr(z) + c
            iterations += 1
        pixels[i, j] = 1 - iterations * 0.02
```

```
gui = ti.GUI("Julia Set", res=(n * 2, n))

for i in range(1000000):
    paint(i * 0.03)
    gui.set_image(pixels)
    gui.show()
```

If Taichi is properly installed, you should get the animation below 🎉:



Documentation

Taichi's documentation is available at https://docs.taichi.graphics.

Contacts

We use these channels to report bugs, discuss design, show off demos and send announcements on a daily basis:

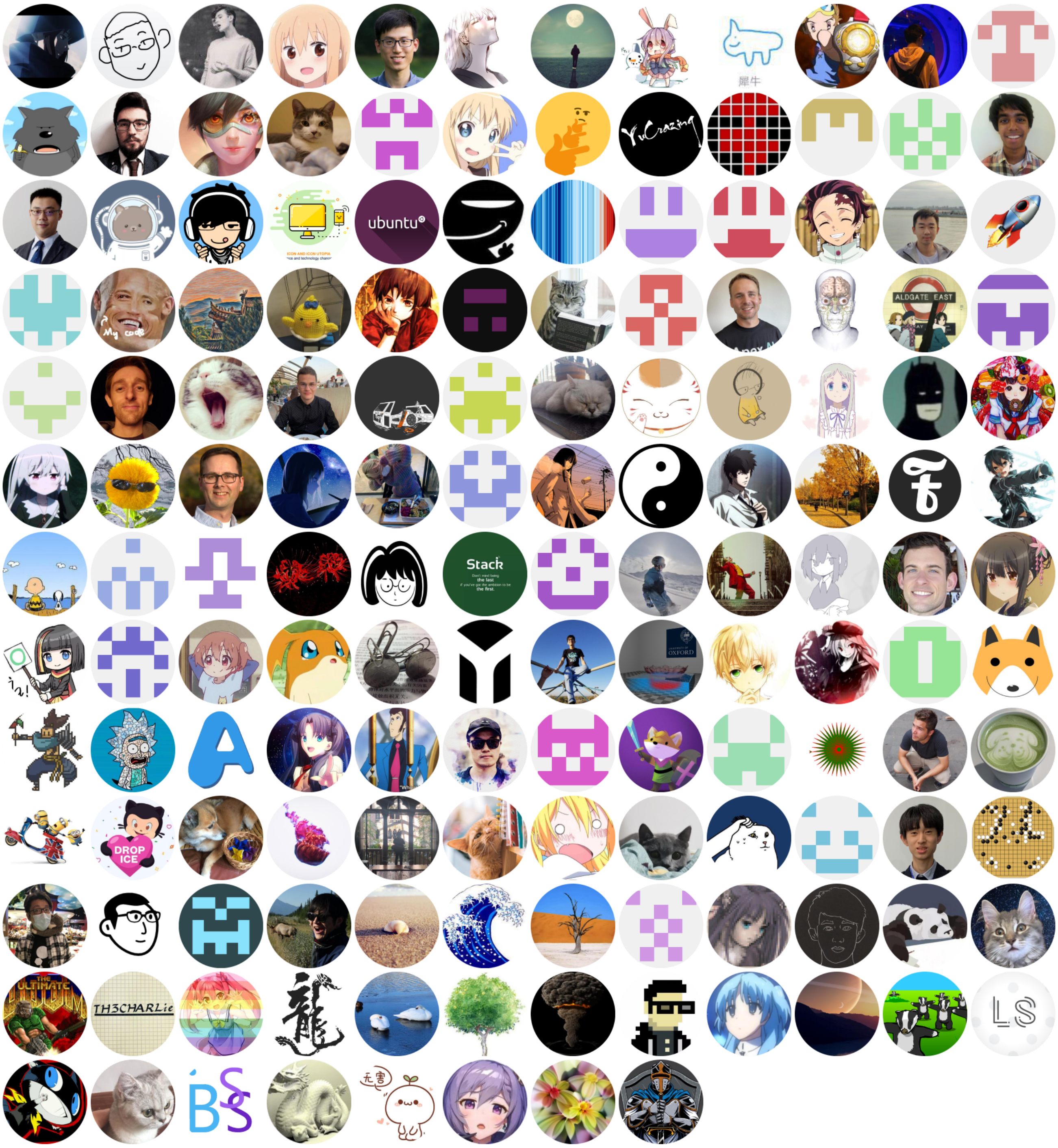
- GitHub Issues
- GitHub Discussions
- Twitter
- Taichi 中文论坛
- Slack & Wechat groups: please send us a message at contact@taichi.graphics first, thanks!

Should you spot any security issue, do not hesitate to report it by mailing to security@taichi.graphics.

Contributing

If you would like to contribute to Taichi, please check out the Contribution Guidelines.

A huge thanks to all of our amazing contributors!



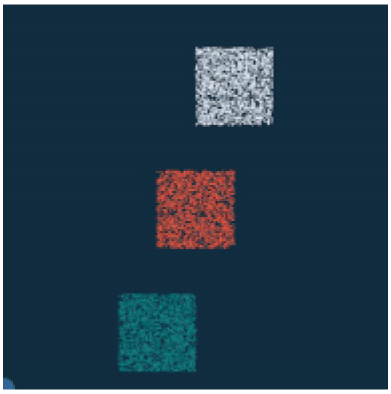
Contributor avatars are randomly shuffled.

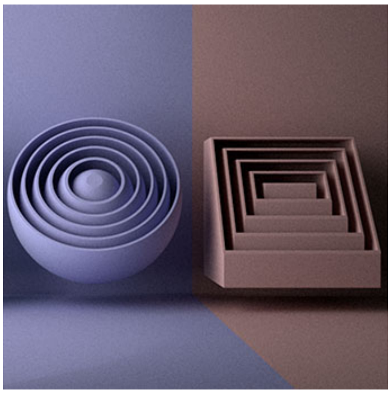
Resources

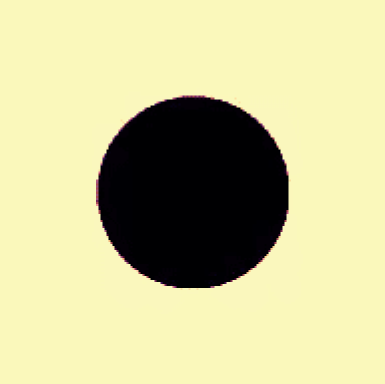
Demos

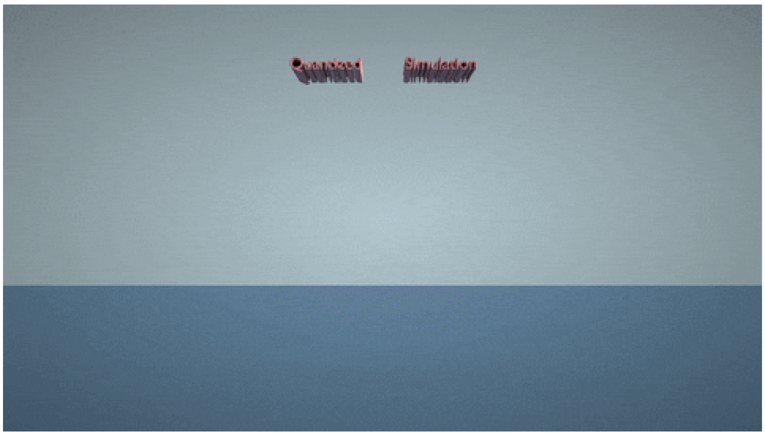
- Taichi examples
- Advanced Taichi examples

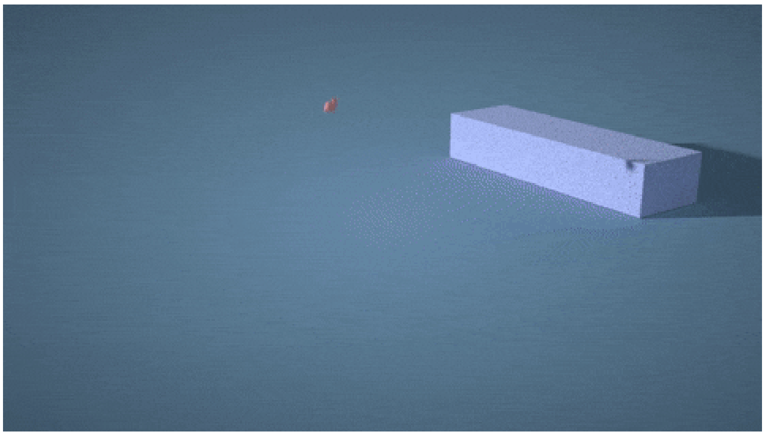
- DiffTaichi
- Taichi elements
- Taichi houdini
- More...











Lectures & talks

- **SIGGRAPH 2020 course on Taichi basics**: YouTube, Bilibili, slides (pdf).
- Chinagraph 2020 用太极编写物理引擎: 哔哩哔哩
- GAMES 201 高级物理引擎实战指南2020: 课件
- 太极图形课第一季：课件
- TaichiCon: Taichi developer conferences
- More to come...

If you use Taichi in your research, please cite related papers:

- **(SIGGRAPH Asia 2019) Taichi: High-Performance Computation on Sparse Data Structures** [Video] [BibTex] [Code]
- **(ICLR 2020) DiffTaichi: Differentiable Programming for Physical Simulation** [Video] [BibTex] [Code]
- **(SIGGRAPH 2021) QuanTaichi: A Compiler for Quantized Simulations** [Video] [BibTex] [Code]

Releases 94

v0.8.11 Latest
13 days ago

+ 93 releases

Used by 215

+ 207

Contributors 155

+ 144 contributors

Languages

● **C++** 64.7% ● **Python** 32.6% ● **C** 0.8% ● **CMake** 0.7% ● **GLSL** 0.5% ● **Cuda** 0.4%
● **Other** 0.3%