

---

# **taichi elements documentation**

***Release 0.0.1***

**The Taichi Elements Developers**

**Feb 09, 2023**



<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Installation</b>	<b>3</b>
2.1	Install as a Python package . . . . .	3
2.2	Install as a Blender plugin . . . . .	3
<b>3</b>	<b>Using Taichi elements in Python</b>	<b>5</b>
<b>4</b>	<b>Blender nodes for Taichi elements</b>	<b>7</b>
4.1	Node System . . . . .	8
4.2	Node Sockets . . . . .	8
4.3	Nodes . . . . .	8
4.3.1	MPM Solver . . . . .	8
4.3.2	Material . . . . .	9
4.3.3	Integer . . . . .	9
4.3.4	Float . . . . .	10
4.3.5	Folder . . . . .	10
4.3.6	Emitter . . . . .	11
4.3.7	Inflow . . . . .	12
4.3.8	Simulation . . . . .	12
4.3.9	Hub . . . . .	13
4.3.10	Source Object . . . . .	14
4.3.11	FCurve . . . . .	14
4.3.12	Texture . . . . .	15
4.3.13	Disk Cache . . . . .	15
4.3.14	Gravity . . . . .	16
4.3.15	Make List . . . . .	16
4.3.16	Merge . . . . .	17
4.3.17	Frame . . . . .	17
4.3.18	Reroute . . . . .	18
<b>5</b>	<b>Miscellaneous</b>	<b>19</b>



# CHAPTER 1

---

## Introduction

---

Taichi elements is a high-performance physics engine for VFX. It can be used as either a Python package or a Blender plugin.

It supports multiple materials (e.g. water, elastic objects, sand, snow).

(TODO: write more...)



#### **2.1 Install as a Python package**

(TODO)

#### **2.2 Install as a Blender plugin**

(TODO)





---

Using Taichi elements in Python

---

We introduce the Python API through a basic 2D demo.

```
import taichi as ti
import numpy as np
from mpm_solver import MPMSolver

write_to_disk = False

ti.init(arch=ti.cuda) # Try to run on GPU

gui = ti.GUI("Taichi Elements", res=512, background_color=0x112F41)

mpm = MPMSolver(res=(128, 128))

for i in range(3):
    mpm.add_cube(lower_corner=[0.2 + i * 0.1, 0.3 + i * 0.1],
                  cube_size=[0.1, 0.1],
                  material=MPMSolver.material_elastic)

for frame in range(500):
    mpm.step(8e-3)
    if frame < 100 and frame % 2 == 0:
        mpm.add_cube(lower_corner=[0.1, 0.4],
                      cube_size=[0.05, 0.01],
                      material=MPMSolver.material_sand)
    if 10 < frame < 100 and frame % 2 == 0:
        mpm.add_cube(lower_corner=[0.4, 0.7],
                      cube_size=[0.2, 0.01],
                      material=MPMSolver.material_water)
    if 120 < frame < 300 and frame % 40 == 0:
        mpm.add_cube(
            lower_corner=[0.4 + frame * 0.001, 0.6 + frame // 40 * 0.02],
            cube_size=[0.2, 0.1],
            material=MPMSolver.material_snow)
```

(continues on next page)

(continued from previous page)

```
colors = np.array([0x068587, 0xED553B, 0xEEEEF0, 0xFFFF00], dtype=np.uint32)
particles = mpm.particle_info()
gui.circles(particles['position'], radius=1.5, color=colors[particles['material
↪ ']])
gui.show(f'{frame:06d}.png' if write_to_disk else None)
```

---

## Blender nodes for Taichi elements

---

### Contents

- *Blender nodes for Taichi elements*
  - *Node System*
  - *Node Sockets*
  - *Nodes*
    - \* *MPM Solver*
    - \* *Material*
    - \* *Integer*
    - \* *Float*
    - \* *Folder*
    - \* *Emitter*
    - \* *Inflow*
    - \* *Simulation*
    - \* *Hub*
    - \* *Source Object*
    - \* *FCurve*
    - \* *Texture*
    - \* *Disk Cache*
    - \* *Gravity*
    - \* *Make List*

- \* *Merge*
- \* *Frame*
- \* *Reroute*

## 4.1 Node System

To create a taichi-elements simulation, you need to open the Elements window. Next, you need to create a new tree and add the necessary nodes. Addon requires many nodes for simulation. The main node is the Simulation node. This node store the Simulate button. Using this button, you can start the simulation. Examples of node trees can be downloaded from here: [https://github.com/taichi-dev/taichi\\_elements\\_blender\\_examples](https://github.com/taichi-dev/taichi_elements_blender_examples)

## 4.2 Node Sockets

Taichi Elements has the following sockets:

**Integer** - represents a single integer value. Color - gray.

**Float** - represents a single float value. Color - gray.

**Vector** - represents a single 3d vector value. Color - gray.

**Struct** - structure that stores settings. Color - green.

**Add** - dynamic socket, which is needed to create new inputs. Color - black.

**Folder** - socket stores the path to the folder. Color - orange.

**Color** - stores color values in RGB format. Color - yellow.

## 4.3 Nodes

### 4.3.1 MPM Solver

---

**Tip:** Location: Add > Solvers > MPM Solver

---

#### Description

This node tells the simulation to use the MPM method (currently the Material Point Method is the only available simulation method). This node stores the settings of the MPM solver.

#### Parameters

*It has no parameters.*

## Inputs

**Domain Object** - this socket is temporarily not working.

**Resolution** - domain resolution in voxels. The simulation will use a cubic domain. For example, if the Resolution value is 64, then the domain resolution will be 64 x 64 x 64.

**Size** - domain size in meters. The domain is created in such a way that its left, back, bottom corner (in the direction -X, -Y, -Z) is at coordinates 0, 0, 0. And if Size is 10.0, then the right, front, top corner will have a coordinate 10, 10, 10.

## Outputs

**Solver Settings** - it is a socket, which is a set of MPM solver parameters.

---

### 4.3.2 Material

---

**Tip:** Location: Add > Solvers > Material

---

## Description

This node stores information about the properties of the material. Using this node, you can specify what physical characteristics the emitter particles will have. Be it the material of water, snow, sand, etc.

## Parameters

**Material Type** - This parameter specifies what the material will be for Emitters. The following options are available: water, sand, snow, elastic.

## Inputs

*It has no inputs.*

## Outputs

**Material Settings** - This output is material settings. At the moment, from the settings there is only the type of material.

---

### 4.3.3 Integer

---

**Tip:** Location: Add > Inputs > Integer

---

### Description

This is a simple input node that provides an integer value.

### Parameters

*It has no parameters.*

### Inputs

*It has no inputs.*

### Outputs

**Integer Value** - an integer value that can be connected to any other integer socket.

---

## 4.3.4 Float

---

**Tip:** Location: Add > Inputs > Float

---

### Description

This node represents a floating point number.

### Parameters

*It has no parameters.*

### Inputs

*It has no inputs.*

### Outputs

**Float Value** - this socket provides a floating point number that can be connected to any float socket.

---

## 4.3.5 Folder

---

**Tip:** Location: Add > Inputs > Folder

---

### Description

Using this node, you can specify the folder.

### Parameters

*It has no parameters.*

### Inputs

*It has no inputs.*

### Outputs

**String Value** - this output is a string that indicates the folder.

---

## 4.3.6 Emitter

---

**Tip:** Location: Add > Simulation Objects > Emitter

---

### Description

Using this node, you can add an emitter to the simulation. Emitter is a mesh object that emits particles from its volume once.

### Parameters

*It has no parameters.*

### Inputs

**Emit Frame** - indicates the frame number in the timeline in which particles will be emitted.

**Source Geometry** - indicates a mesh object that will emit particles from its volume.

**Material** - this socket accepts material parameters (water, snow, sand, elastic).

**Color** - particle color.

### Outputs

**Emitter** - this socket is a structure that stores the settings of the emitter.

---

### 4.3.7 Inflow

---

**Tip:** Location: Add > Simulation Objects > Inflow

---

#### Description

This type of object emits particles like a faucet. An Inflow object can continuously emit particles, and can also stop the emission of particles, and then continue to emit particles.

#### Parameters

*It has no parameters.*

#### Inputs

**Enable FCurve** - This socket accepts an input animation curve that will indicate the enable value for the inflow object. If you want particles to be emitted throughout the simulation, you can omit the animation curve, but the socket must be connected to the FCurve node. But if you need to turn on and off the inflow object during the simulation, you need to specify the animation curve in the FCurve node. At a value of 0.0, inflow will not emit particles, and at a value of 1.0, continuous emission of particles will occur.

**Source Geometry** - indicates a mesh object that will emit particles from its volume.

**Material** - this socket accepts material parameters (water, snow, sand, elastic).

**Color** - particle color.

#### Outputs

**Inflow** - this socket is a structure that stores the settings of the inflow object.

---

### 4.3.8 Simulation

---

**Tip:** Location: Add > Simulation Objects > Simulation

---

#### Description

This node is a simulation in general. The simulation is launched using the Simulate operator of this node.

#### Parameters

*It has no parameters.*



## Inputs

**Frame Start** - frame on the timeline from which the simulation begins. If you plan to create a particle system from the simulation cache, it is better to set this value to 2, since 0 and 1 frames cannot be displayed correctly (this is a limitation of the blender particle system).

**Frame End** - frame number on the timeline in which the simulation ends.

**FPS** - simulation frame rate.

**Solver** - this socket accepts solver settings as input. At the moment, only MPM Solver can be connected.

**Hubs** - This socket accepts a hub or a list of hubs as an input. The hub node is described below.

## Outputs

**Particles** - this output represents particle simulation data.

---

## 4.3.9 Hub

---

**Tip:** Location: Add > Simulation Objects > Hub

---

## Description

This node is the connecting link between emitters and force fields. At the moment, only one force field is supported in the simulation. In the future, the use of different force fields for individual emitters is possible.

## Parameters

*It has no parameters.*

## Inputs

**Forces** - input for force fields. At the moment, it is possible to connect only one force field. In the future, the capabilities of the simulator and this node will expand, so that it is possible to connect several force fields.

**Emitters** - socket to connect emitter or emitter list. Those emitters that are not connected to any hub object will not participate in the simulation.

## Outputs

**Hub Data** - these are the settings of the hub object.

---

### 4.3.10 Source Object

---

**Tip:** Location: Add > Source Data > Source Object

---

#### Description

Allows you to select and use an mesh object from the scene in the simulation.

#### Parameters

**Object** - the name of the object to use.

#### Inputs

*It has no inputs.*

#### Outputs

**Source Geometry** - geometric data from the mesh of the object.

---

### 4.3.11 FCurve

---

**Tip:** Location: Add > Source Data > FCurve

---

#### Description

Animation curve. You can use it to specify animation for the Enable parameter of the Inflow object. To do this, create a Custom Property on any scene object and animate this custom property. Next, you can specify an animation curve using this node.

#### Parameters

**Action** - name action from the blend file.

**FCurve Index** - index of the animation curve. If this index is specified correctly, the name of the animation curve will be displayed below.

#### Inputs

*It has no inputs.*

## Outputs

**FCurve Values** - animation curve values.

---

### 4.3.12 Texture

---

**Tip:** Location: Add > Source Data > Texture

---

## Description

Allows you to select a texture from a blend file. At the moment, this node cannot be used anywhere. In the future, it is planned to expand the capabilities of the simulator and it will be possible to use this node.

## Parameters

**Texture** - the name of the texture to use.

## Inputs

*It has no inputs.*

## Outputs

**Texture** - an object that represents data and texture parameters.

---

### 4.3.13 Disk Cache

---

**Tip:** Location: Add > Output > Disk Cache

---

## Description

This node saves the simulation to disk.

## Parameters

**Particle System** - import particles from the cache and create a particle system based on them.

**Particle Mesh** - import particles from the cache and create a mesh based on them. The created mesh will only have vertices.

## Inputs

**Particles** - this input receives a list of particles from the Simulation node.

**Folder** - path to save and import cache.

## Outputs

*It has no outputs.*

---

### 4.3.14 Gravity

---

**Tip:** Location: Add > Force Fields > Gravity

---

## Description

Gravitational force field.

## Parameters

*It has no parameters.*

## Inputs

**Speed** - temporarily not working.

**Direction** - direction and force of gravity.

## Outputs

**Gravity Force** - structure that represents gravity settings.

---

### 4.3.15 Make List

---

**Tip:** Location: Add > Struct > Make List

---

## Description

Combines several structures (nodes) into one list.

### Parameters

*It has no parameters.*

### Inputs

**Add Element** - dynamic socket with which you can connect many nodes.

**Element** - list item.

### Outputs

**Elements** - list of input structures.

---

## 4.3.16 Merge

---

**Tip:** Location: Add > Struct > Merge

---

### Description

Combines lists of structures that are formed using the Make List node.

### Parameters

*It has no parameters.*

### Inputs

**Merge Lists** - dynamic socket with which you can connect many nodes.

**List** - list item.

### Outputs

**Elements** - merged lists items.

---

## 4.3.17 Frame

---

**Tip:** Location: Add > Layout > Frame

---

### Description

Standard frame blender node.

### Parameters

*It has no parameters.*

### Inputs

*It has no inputs.*

### Outputs

*It has no outputs.*

---

## 4.3.18 Reroute

---

**Tip:** Location: Add > Layout > Reroute

---

### Description

Standard reroute blender node.

### Parameters

*It has no parameters.*

### Inputs

*It has no inputs.*

### Outputs

*It has no outputs.*

## CHAPTER 5

---

Miscellaneous

---

(TODO)