

What is 2D Soft Physics?

This is easy-to-use asset, which allows you to create 2d jelly physics objects with only two scripts. So all you need it's to add two scripts on your GameObject:

- **SpriteObject**
- **SoftObject**

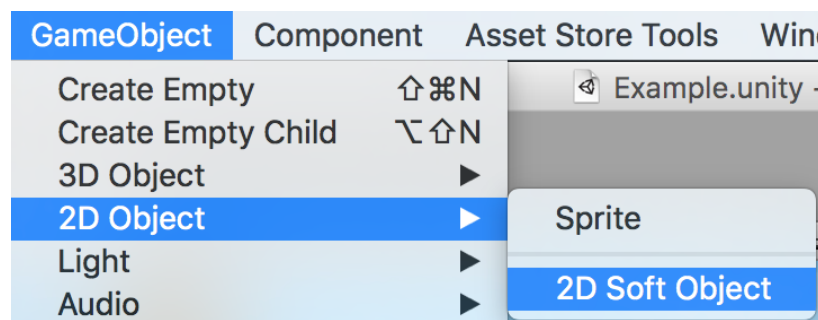
and choose a sprite to set your jelly-physics object. Also it works with 2D Toolkit.

You can setup Fill Type, Joint Radius, Distance, Frequency, DampingRatio, Mass, AngularDrag, LinearDrag, Physic Material, Layer, of physics-objects in the Inspector panel.

This works well on Personal and Pro Unity, suitable for Web, Standalone, Android and iOS platforms. And all this takes less then 3 mb on your drive.

Quick Start

As mentioned above, to create jelly-physics object, create a new GameObject, from Menu «**GameObject/2D Object/2D Soft Object**» and choose a Texture in Inspector window for «**SoftSprite**» component.



That is all. Now you can try to set different parameters of «**SoftObject**». For example - set Joint Radius, Distance, Frequency and Mass as 0.25, 0.1, 20, 5 and jelly-object will have other physic behavior!

API Help

Table of content

- **SoftSprite** script
- **SoftObject** script

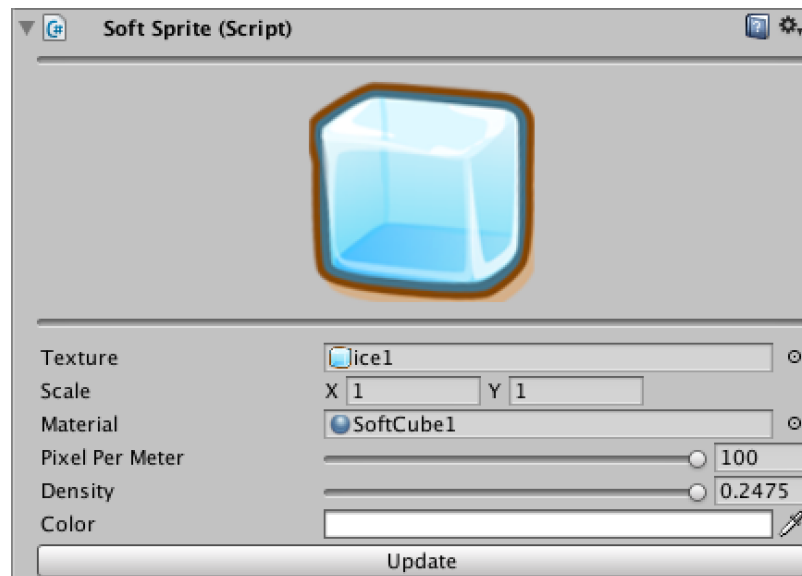
SoftSprite script

SoftSprite script creates and configures image on scene, using MeshFilter.

SoftSprite script has a few parameters:

- **Sprite** - it's «Texture», which used to create a sprite on scene;
- **Scale** - size of Texture;
- **Material** - Material of MeshRenderer;
- **Pixel Per Meter** - analogue Sprite's pixelPerMeter;
- **Density** - how often mesh quads will be repeated;
- **Color** - color of sprite.

Also if you need to update your sprite, press the button «Update».



`private void Awake()` - cache a components, create or update a MeshFilter;

`private void CreateSpriteMaterial()` - create a Material of sprite;

`private void CreateMesh()` - create a MeshFilter of sprite;

`private void UpdateMesh()` - update a MeshFilter of sprite;

`private void UpdateMaterialTexture(Texture texture)` - update a texture of material, argument - texture, which sets to material;

`public void ForceUpdate()` - update mesh of sprite.

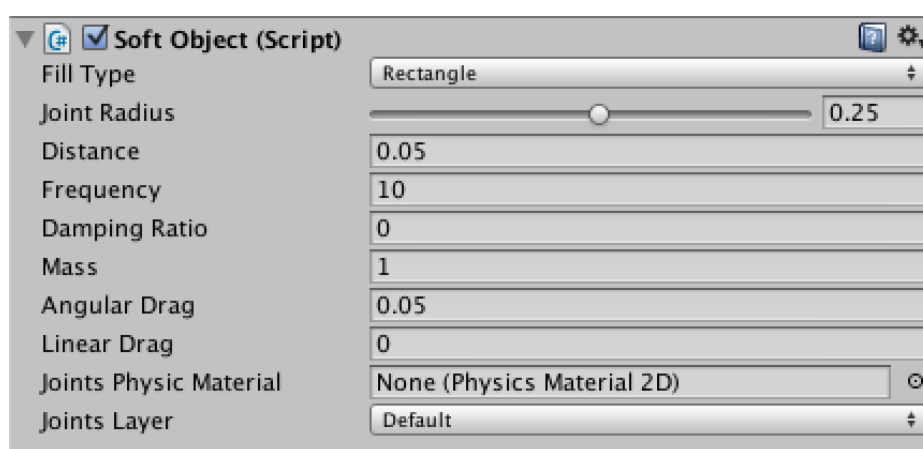
SoftObject script

SoftObject script creates and configures joints to make jelly-physics effect.

SoftObject script has a few parameters:

- **Fill Type** - type of filling by CircleCollider2D's with joints, can be Rectangle or Ellipse;
- **Joint radius** - the radius of child CircleCollider2D with joints;
- **Distance** - the distance that the spring should attempt to maintain between the objects;

- **Frequency** - at which the spring oscillates while the objects are approaching the separation distance you want (measured in cycles per second): In the range 0 to 1,000,000 - the higher the value, the stiffer the spring;
- **DampingRatio** - the degree to which you want to suppress spring oscillation: In the range 0 to 1, the higher the value, the less movement;
- **Use Auto Mass** - automatically calculate mass value, instead manual settings;
- **Mass** - mass of this object;
- **AngularDrag** - drag coefficient affecting rotational movement;
- **LinearDrag** - drag coefficient affecting positional movement;
- **Joints Physic Material** - physics material for joints;
- **Joints Layer** - layer of joints;



```

private void Awake() - cache components, initialize and configure joints;
private void CacheObjects() - cache components;
private void Initialize() - initialize and configure joints;
private void FillGrid(int jointsHorizontalCount, int jointsVerticalCount, float mass,
Vector2 scaledSize, Vector2 offset) - initialize and configure joints for Rectangle Fill
Type;
private void FillCircle(int jointsHorizontalCount, int jointsVerticalCount, float mass,
Vector2 scaledSize) - initialize and configure joints for Ellipse Fill Type;
private void CreateJoint(int id, float mass, Vector2 jointPosition, float[] weights) - create
configured joint;
private void CalcVertsWeight() - calculate total joints wights for each vertex;
private Vector2 GetMeshQuadSize() - return size of first quad in mesh;
private Vector2 IgnoreCollisions() - ignore collisions of joints;
private void FixedUpdate() - update MeshFilter;
private void DestroyJoints() - destroy joints;
public void ForceUpdate() - delete and create joints and set joints parameters;
public void UpdateParams() - set joints parameters.

```

Tips

- Choose `SoftObject.FillType` for different jelly objects forms: rectangle or ellipse form;
- You can also choose different value for `SoftSprite.Density` to change Mesh vertex count and `SoftBody` behavior;
- For better result setup `SoftSprite.Density` the same as `SoftObject.JointRadius`.

Please let me know if you have any questions.

E-mail: unitymedved@gmail.com