What is Projectile Motion Pro?

It is a library of physically-based launch force calculation algorithms for Unity that calculates real force to launch rigid bodies that perform projectile motions and hit / pass through given points.

Currently supported parameters (specify any of the following): time of flight, max height, launch angle, launch speed, and coefficient a (coefficient of the quadratic function f(x) = ax2 + bx + c).

Play the demos

Online

Click here to play the online version (WebGL). This is more convenient than play in editor since you don't need to change the presets back and forth. Note that the online version will keep up-to-date and may not be the same version as yours.

In editor

The demo scenes are under the folder "Blobcreate/Projectile Motion Pro/Demos". Before playing, back up your layer settings and physics settings, and apply the layer preset "PMPLayers" and physics preset "PMPPhysics" under ".../Demos/Other Assets/Settings". Detailed steps:

1. (At the top right of Unity editor) select "Layers > Edit Layers...",
2. Click the second icon in the top right of the inspector,
3. Click "Save current to..." button, save your current layers,
4. Now your layers are backed up, click that icon again and then choose "PMPLayers" in the pop up window.

Setting up the physics is similar, select "Edit > Project Settings...", select "Physics", click the second icon in the top right, back up your current settings, and apply "PMPPhysics" preset.

The demo scenes use URP. If your project uses URP but there are rendering problems with demo scenes you can use "PMP-URP-HighQuality" render pipeline settings. If you are using other render pipeline, change the shaders of the materials to the equivalent and you are good to go.

Check out and explore

Which scene demonstrates which algorithm? The relationship is shown in the table below:

Method	Scene Index(es) / `class(es)`
VelocityByA	02 / `JumpAttacker` `ProjectileLauncher`
VelocityByAngle	03 / `CannonLike`
VelocityByTime	02 / `Defender`
VelocityByHeight	00 / `JumpTester`, 01 / `NMJump`, 02 / `JumpAttacker`
AnglesBySpeed	03 / `CannonLike`
VelocitiesBySpeed	03 / `CannonLike`

> **Note**
>
> The script filenames of the classes are `class` + `.cs`. You can find them under the folder ".../Demos/Scripts".

Remarks

Projectile Motion Pro provides various algorithms to meet the needs of different scenarios:

Method	Example use case
VelocityByA	If you want a way to automatically increase the flight time of projectiles with the increase of distance and works at any elevation angle, use this method.
VelocityByAngle	Launch anything to a target with a specific elevation angle.
VelocityByTime	Great for RTS games when you want to use some projectile weapon to accurately hit moving targets.
VelocityByHeight	You can use it to replace the default jump behavior of Unity's Off-Mesh Link / NavMesh Link to achieve realistic jump for AI units (enemies, NPCs, player, etc.). Also great for jump pad mechanics.
AnglesBySpeed	Great for simulations of weapons with a fixed launch speed.
VelocitiesBySpeed	A extended version of AnglesBySpeed. It is more convenient than AnglesBySpeed when the rotation animation is not separated into y axis and x axis, or when there is no animation needed.

Two lines of code to launch

The integration to your own scripts is very simple (take `VelocityByTime` for example):

```
// Take VelocityByTime for example. It is very
// useful for accurately striking moving objects.
var f = Projectile.VelocityByTime(myRigid.position, predictedPos, flyTime);
myRigid.AddForce(f, ForceMode.VelocityChange);
```

> **Note**
>
> You should also include `Blobcreate.ProjectileMotionPro` namespace to your scripts.

For detailed description of each method, please see the Scripting Reference.
