Add **Physics Draw 2D** to your project and enjoy creating games with 2D drawing mechanics, where the drawn lines have Rigidbody2D and PolygonCollider2D. Let the player draw lines managed by the physics engine and create amazing puzzles in your game.

This Physics Draw 2D asset documentation contains the explanations for the implemented features, classes, methods, variables and setting adjustments.

Contact

Asset Store Page

Github

support@meadowgames.com

MeadowGames
MeadowGames.com

CONTENTS

1. OVERVIEW

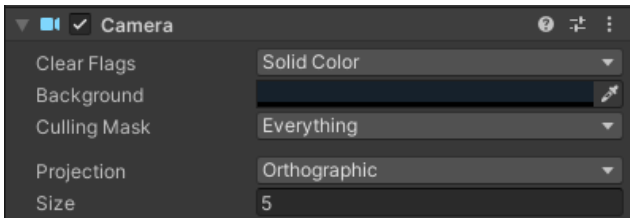
Physics Draw 2D is a simple system to boost your project with a 2D drawing mechanic, where each drawing works with the physics engine by having Rigidbody 2D and a Polygon Collider 2D.

The system can be added to any 2D game project and be configured to fit the game with simple configurations of the drawing template and the general settings.

2. Getting Started

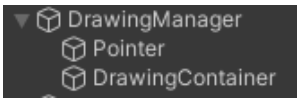
Some basic requirements should be met so the system works properly.

The Camera Projection should be Orthographic:



The Scene must contain a Drawing Manager (PD2_DrawingManager):

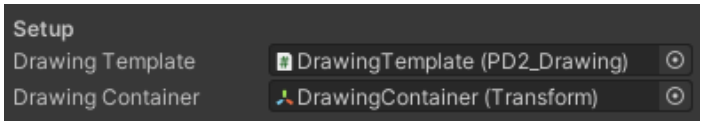
The Manager contains two child GameObjects, Pointer and DrawingContainer. There is a prefab of the Drawing Manager at “Assets/MG_PhysicsDraw2/Prefabs”.



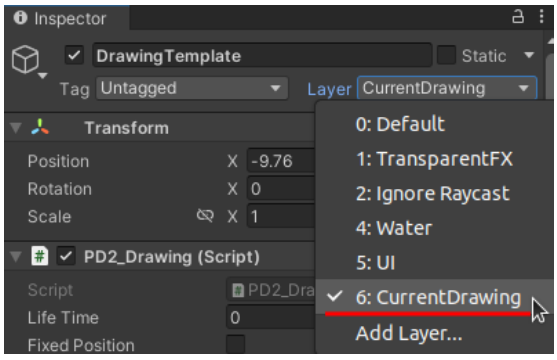
The “Drawing Template” field of the Drawing Manager is filled with the proper template:

The sample scene has a GameObject named “DrawingTemplate” on the scene for making it easier to change the Drawing settings, this object is the one used as a template in the Drawing Manager.

It is also possible to use a previously set prefab as the template, as the one available at “Assets/MG_PhysicsDraw2/Prefabs”.



Make sure a layer named “CurrentDrawing” is added:



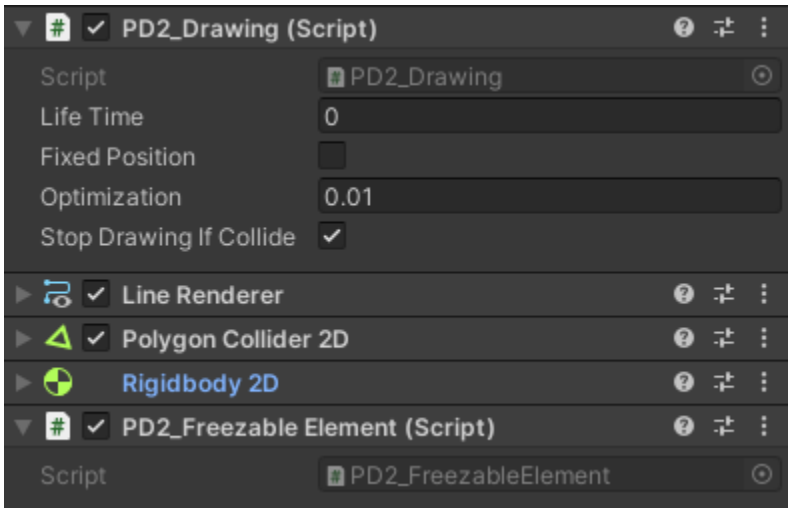
3. PD2 Components

In this section, the core components of Physics Draw 2D are explained.

3.1. Drawing

Each line drawn by the player contains a PD2_Drawing component, along with a line renderer, polygon collider, rigidbody 2d and a PD2_FreezableElement.

The components variables can be adjusted previously by adjusting them on the Drawing Template. The drawn lines can also have some of the variables adjusted, however, adjusting the line renderer width or points requires that the PD2_Drawing.UpdateMesh() method is called so the collider is updated.



3.1.1. PD2_Drawing

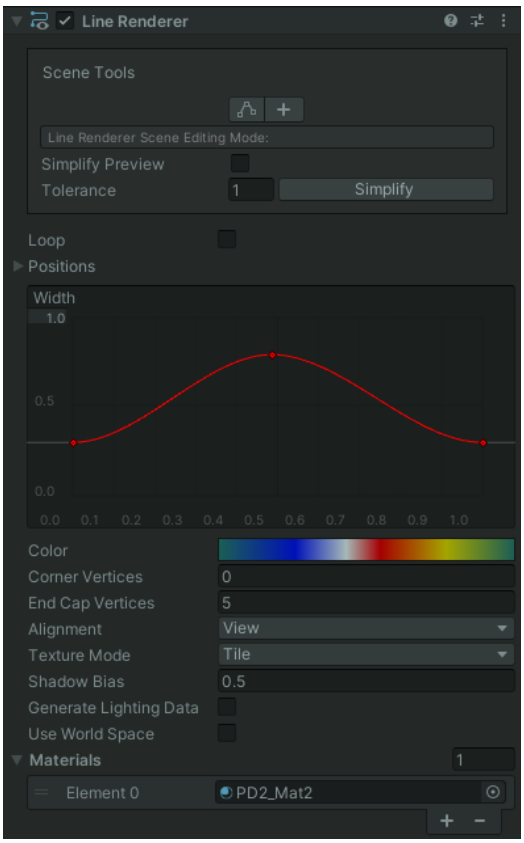
```
# Settings
float lifeTime //If > 0, delay in seconds before destroying this drawing
bool fixedPosition //line does move or is affected by gravity
bool stopDrawingIfCollide //If drawn over a collider 2D, the drawing ends
float maxLength //This line ends when it reaches the maxLength
float optimization //level of optimization of the collider, higher value leads
to less vertices

# Info
float length //current line length
```

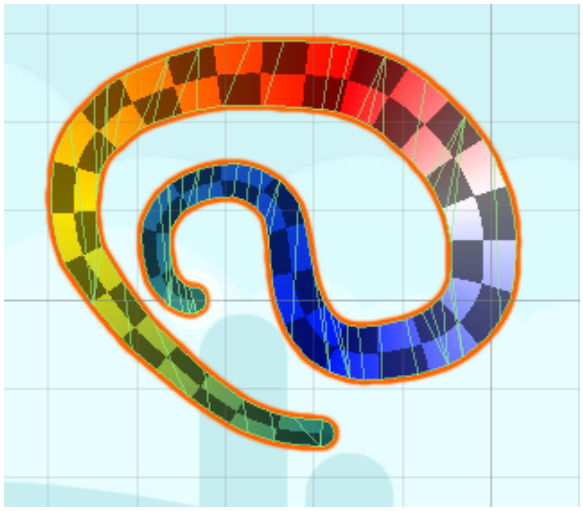
```
# Methods
static PD2_Drawing StartDrawing(Vector3) //Create new potential line. This
method must be called before adding points to the line
void AddPoint(Vector3) //Add points to the line
void FinishDrawing() //Ends the drawing and adjusts the polygon collider
void UpdateCollider(bool = false) //Update the polygon collider
void Destroy() //Remove the line
```

3.1.2. Line Renderer

You can adjust all the line renderer settings to better fit your project requirements, for example, the line Width curve, Color gradient, End Cap Vertices and Materials.

Mind that adjustments on the width or points require that the PD2_Drawing.UpdateMesh() method is called so the collider is updated.





3.1.3. Polygon Collider 2D and Rigidbody 2D

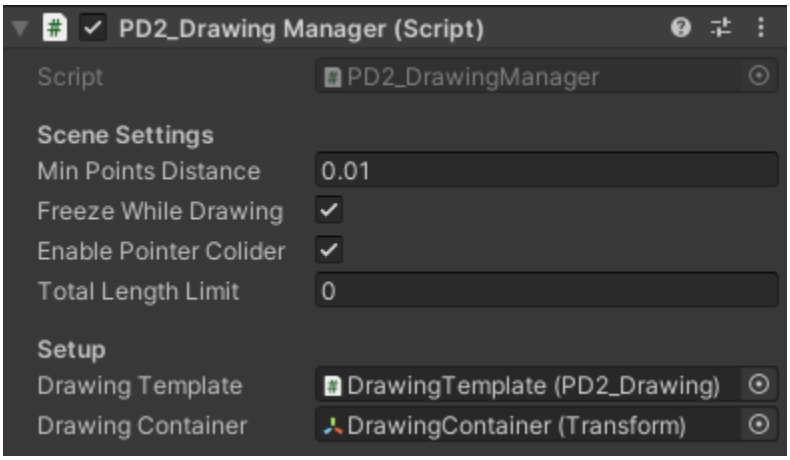
The Polygon Collider 2D and Rigidbody 2D are already setup to make the project work as expected, however, you can still change some settings without affecting the core system:

Polygon Collider 2D: Density and Material

Rigidbody 2D: Material, Use Auto Mass, Linear Drag, Angular Drag, Gravity Scale and Collision Detection

3.2. Drawing Manager

The drawing manager is responsible for the general system settings, events and managing the drawn lines.




```
# Scene Settings
float minPointsDistance //Minimum distance between the drawing points
bool freezeWhileDrawing //Makes all the freezable objects stop their movement while a line is being drawn
bool enablePointerCollider //Enables the pointer collider while a line is being drawn. It makes the drawn line constrained by the scene colliders
float maxLength //Maximum total length (summing all the drawings lengths)

# Setup
PD2_Drawing drawingTemplate //Template GameObject in scene or as prefab used to instantiate new drawings
Transform drawingContainer //Transform that organizes all the drawings as children. New lines are instantiated as child of this Transform

# Event Manager
PD2_EventManager eventManager //Events: OnUpdate, OnPointerDown, OnPointer, OnPointerUp

# Info
List<PD2_Drawing> drawingList //List of all the drawings in scene
```

```
PD2_Drawing currentDrawing //The line that is being drawn
float totalLength //Total length summing all the line lengths

# Methods
void OnPointerDown() //Start new line and manages the pointer down actions
void OnPointer() //Add points to the line and manages the actions while pointer
is held down
void OnPointerUp() //Ends the drawing and manages the pointer up actions
void AddDrawing(PD2_Drawing) //Add new drawing to the draingList
void RemoveDrawing(PD2_Drawing) //Remove drawing from the draingList and scene
void RemoveDrawing(int) - //Remove drawing from the draingList and scene
void RemoveDrawingAll() - //Remove all drawings from the draingList and scene
```

3.3. Pointer

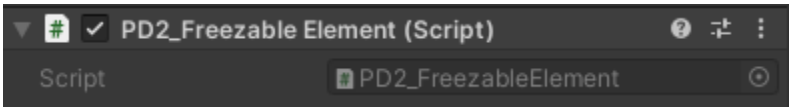
The pointer Target Joint 2D component can be adjusted so the drawing gets smoother. The basic adjustment is the Frequency.

On the start of a new Drawing, the size of the pointer is adjusted according to the end of the line width.

3.4. Freezable Element

Any scene Object with a Rigidbody 2D and needs to be freezed while the player is drawing should be added a PD2_FreezableElement component.

By default, the drawn lines already contain this component.



4. Sample Scene

The sample scene contains the needed components already set up and two circular objects with different physics materials and a Freezable Element component to test the Physics Draw 2D.