Standardize Bows

An asset that simulates the movement and physics of a bow without an animator.

First of all, thank you for buying my asset. This is my first try on publishing something online. I tried to make it as detailed as possible. All of the scripts are heavily commented and there are some tutorial videos that I created in order for you to quickly start using this tool.

So here is how we are going to do it.

1. **First I will give a TL;DR on what this asset can do and how to setup it easily.**

2. **I will try to explain what this script does and how can it be developed even further.**

3. **I will inform you about the critical stuff that you should know when you are using this on your own bow models.**

4. **Lastly, I will share my contact email for support.**

Table Of Contents

TL:DR - Quick Setup

What is Standardize Bows?

This asset is developed in order to standardize the procedure of rigging and animating a bow for Unity.

Developing a game like RPG that has lots of bow models might be scary to someone that has no particular experience modeling and animating. That someone was actually me :) So I created this asset.

With this asset you can set up a bow for your game in approximately 5 minutes and the best part is that since all of the action is done in C#, you can just tweak the parameters to create the look you want.

Script Selection

Standalone Bow (Like VR)

1.Standardize Bow
2.Standardize Projectile

For Player Character With Animator(RPG)

1.Standardize Bow For Hand
2.Standardize Projectile For Hand
3. Standardize Bow For Hand Bool
4.Standardize Projectile For Hand Bool

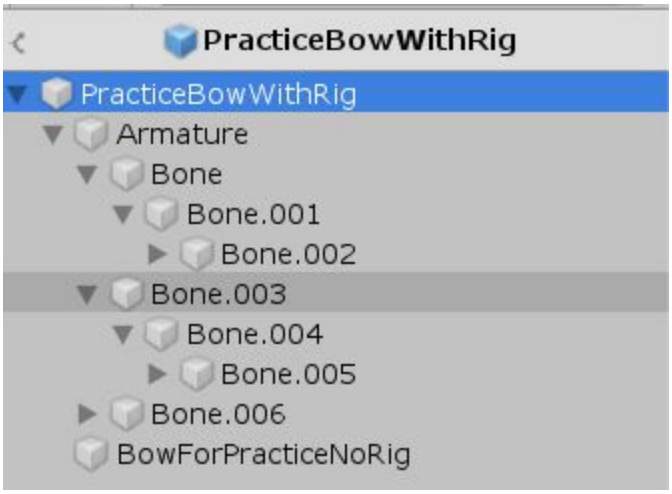
Basically Standardize Bow scripts are for bow models and the Standardize Projectiles are for projectiles. Standalone one is the easiest one to setup so we will start with that. It doesn't require a bow user to be existent in the scene. It just simulates the bow and fires off projectiles depending on the trigger or command you are giving it. The default trigger is the Left Mouse Button.

```
319     // Update is called once per frame
320     void Update()
321     {
322         // STATE 1 - Pulling the string - Default Trigger is left mouse click
323         if (Input.GetKey(KeyCode.Mouse0))
```

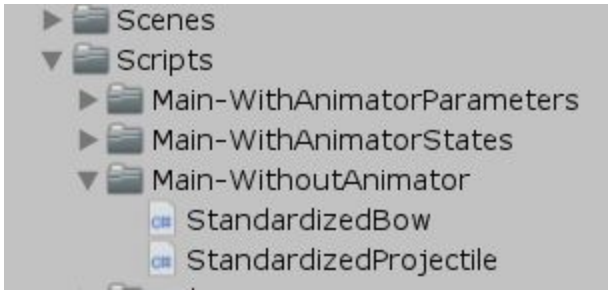
If you don't like reading, you can follow along with the tutorial videos. You should especially watch the one video where we created a simple rig for your bow. **If you follow the videos, you will have a smoother experience while using this asset.**

Main Steps For Standardized Bow Script

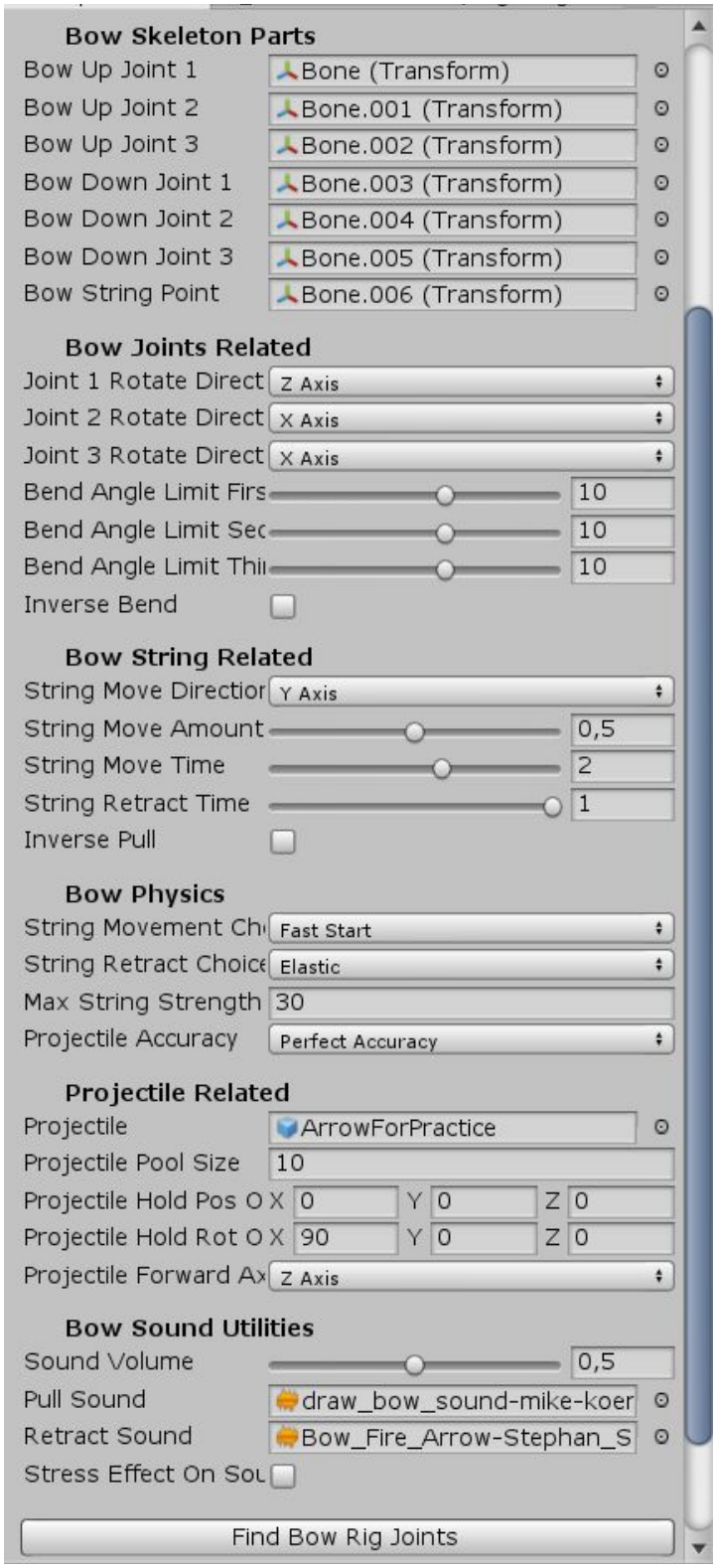
1. Make sure that you have a bow that has the rig as shown in the tutorial video. It is really short and you don't need to have any experience to do it. Just follow along with the short tutorial video if your bow doesn't have a rig. You can also use the bow prefabs and meshes that I placed in this asset for your practice.



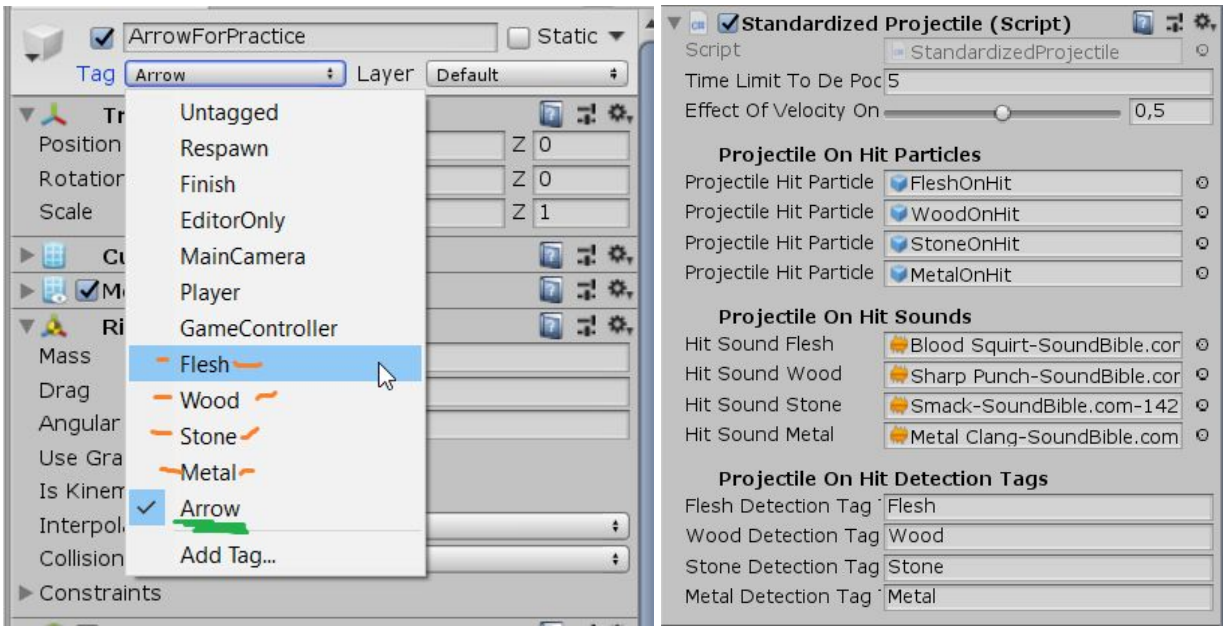
2. Attach the StandardizedBow script to your prefab that holds your armature and bow model. And add the StandardizedProjectile script to the prefab that you want your bow to fire as a projectile. If you don't have a bow or projectile prefab, just use the ones that I created for you as an example.



3. If you have followed along the tutorials, then all you have to do is to press “Find Bow Joint Rigs” in the bottom of the StandardizedBow script and save your prefab.



4. For the StandardizedProjectile the default values should be fine but in order to detect collider triggers correctly, change the tag to something other than Default. It is best to create tags for each surface(Flesh,Wood,Stone,Metal) you want to detect and your projectiles(Arrow).

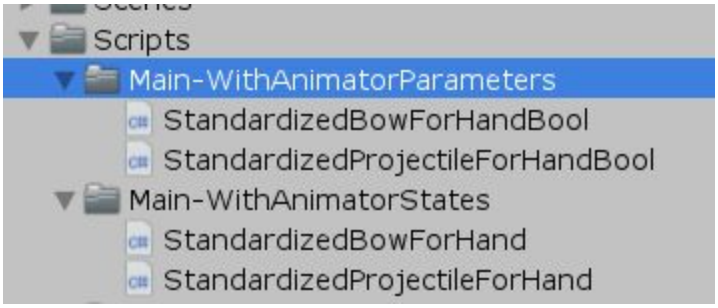


5. If you have followed my instructions correctly, you can start using your bow in PlayMode. The default joint angle values can be changed for a more natural look and if your projectile spawns on a different position and angle, play with the Position&Rotation Offset variable in the StandardizedBow script.

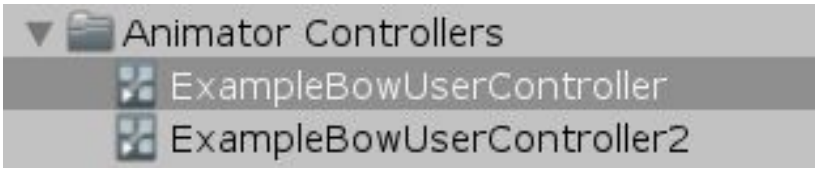
Main Steps For Standardized Bow For Hand Scripts

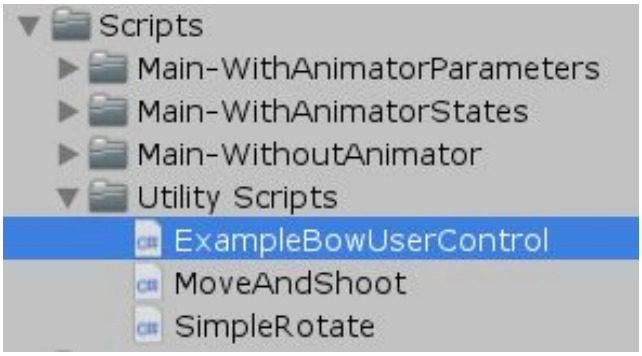
1. Again, make sure you have a bow that has the rig defined in the tutorial video. It is really short and you don't need to have any experience to do it. Just follow along with the short tutorial video if your bow doesn't have a rig. You can also use the bow prefabs and meshes that I placed in this asset for your practice.

For this part, I couldn't create a Example Scene since it requires character models and animations for it. But if you have a humanoid character in your project, you can just use them with the animations you have(Check out Mixamo if you don't).

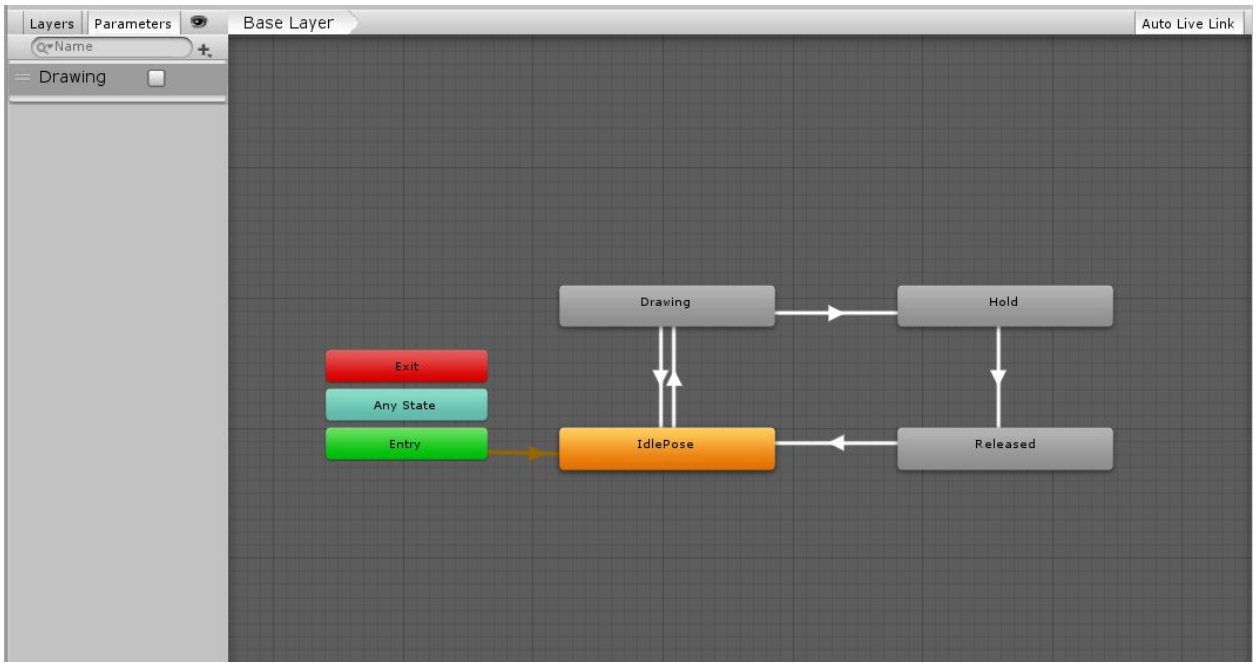


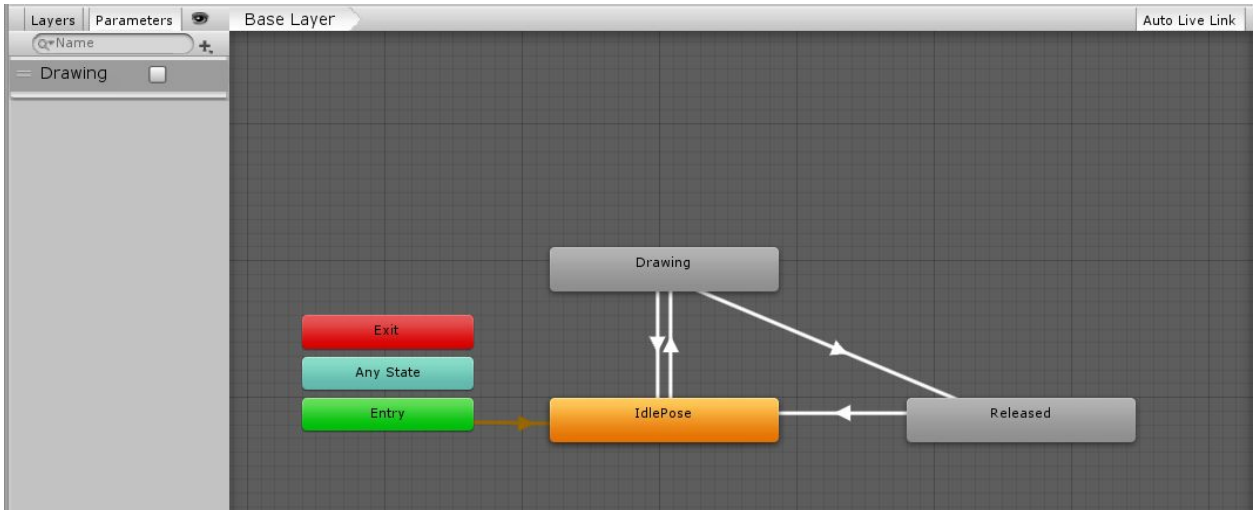
2. First, we should decide on our player animator and model. If you are planning to integrate your bow to your character with Animator States(recommended) than use StandardizedBowForHand. But if you want to use a simple Animator Controller Boolean parameter, that is fine too. Just use StandardizedBowForHandBool.



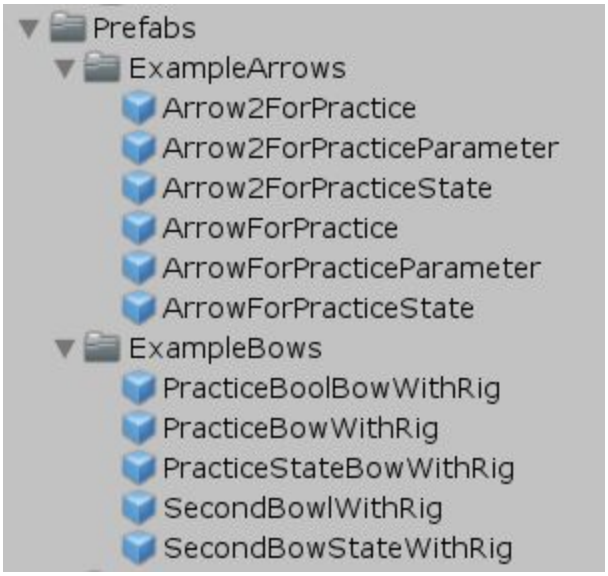


For each of them, you will be using a Mecanim system that looks like the ones that I created as an example to you. You can also use your existing Animator with just changing related variables in inspector of the script. I also created a ExampleBowUser script for your player character which triggers these animation states.

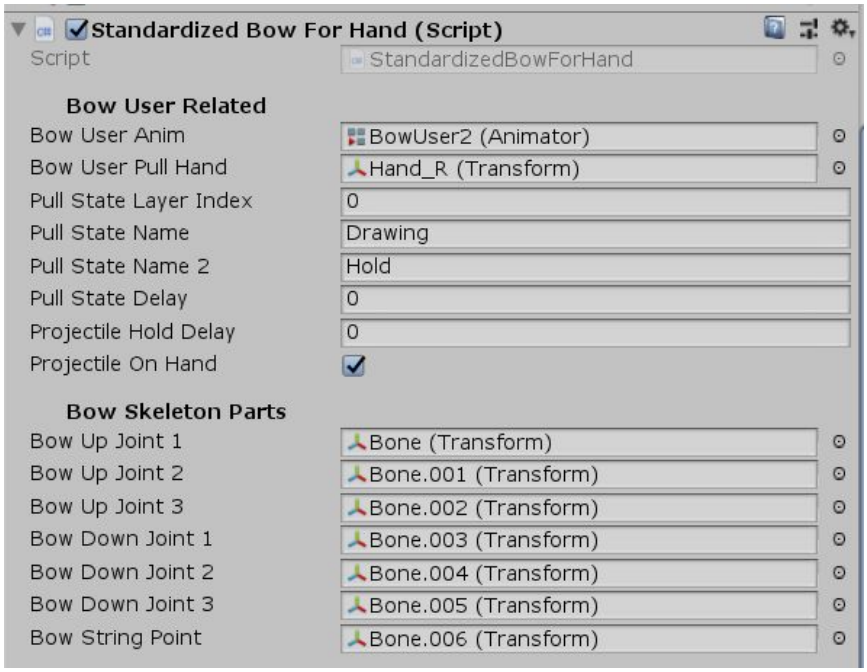




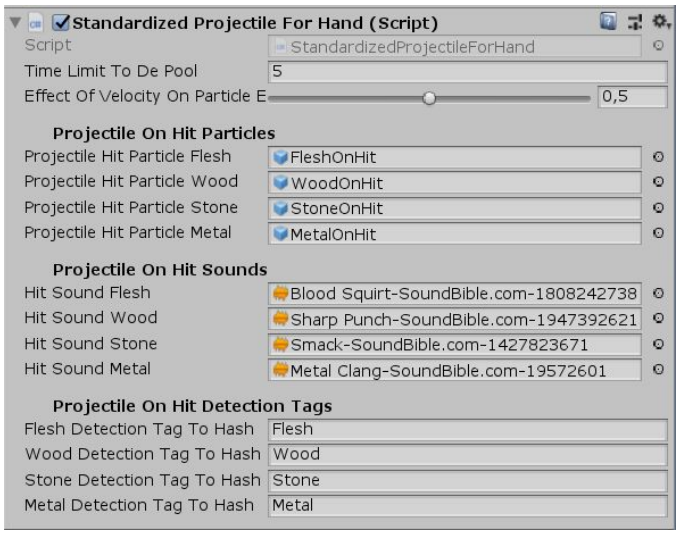
3. After setting your bow user character, add the selected bow script to your bow prefab and projectile script to your projectile prefab. If you don't have a bow model or projectile model, you can use the ones that I created as an example in prefabs folder. Don't forget to parent the bow to the hand that carries it in your character too.

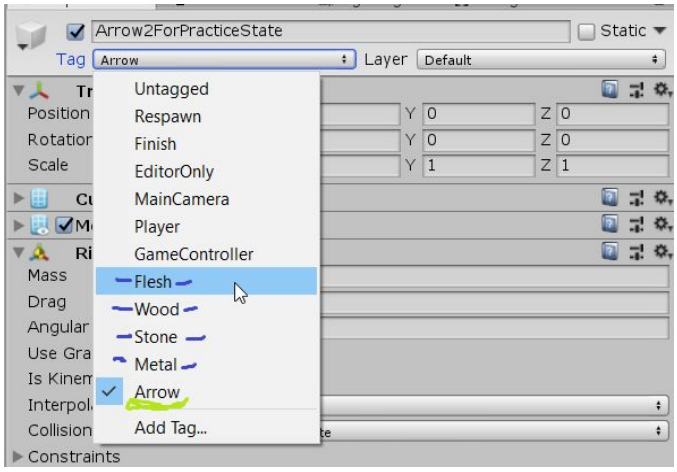


4. After adding your bow script, make sure you assign the bow user animator and the transform of the hand you want your player to pull the string. Also if you followed the rigging tutorial I made, just use the "Find Bow Rig Joints" in the bottom to assign the joints of the bow.



5. For the projectile script the default values should be fine but in order to detect collider triggers correctly, change the tag to something other than Default. It is best to create tags for each surface(Flesh,Wood,Stone,Metal) you want to detect and your projectiles(Arrow).



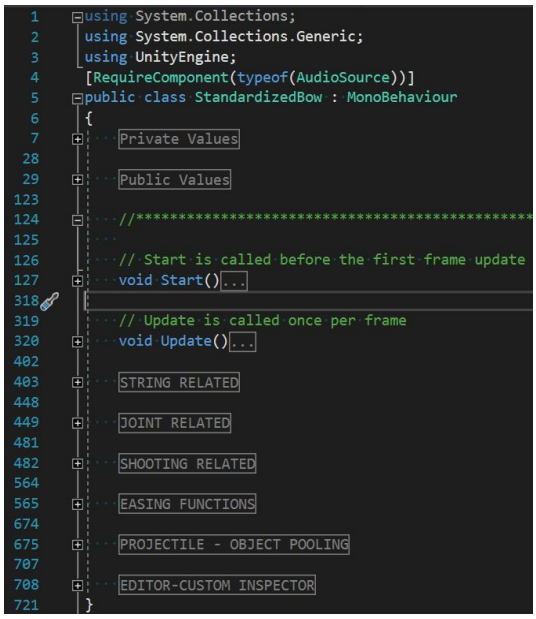


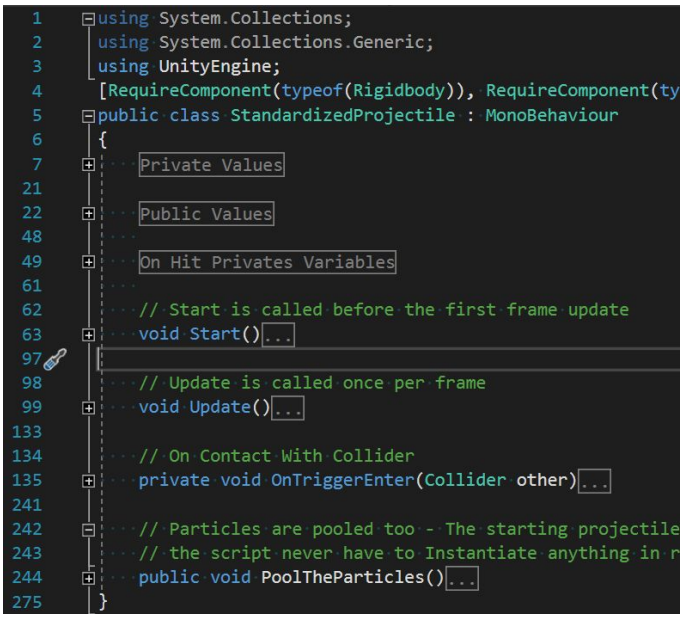
6. Depending on the name of your Animator States or Parameters, do the required changes in the bow script inspector. Try using a reasonable delay amount for your bow to sync with the player animator state. When should the bow string and projectile triggered after your player starts animating.

The default joint angle values can be changed for a more natural look and if your projectile spawns on a different position and angle, play with the Position & Rotation Offset variable in the bow script. Try to play with them and find the values that suits you.

Variables & Workflows

What do the variables mean in these scripts?





All of the functionalities in these scripts are clearly organized and almost all of the public variables have Tooltips on them so you can just hover over them and read their description. I also tried to make their names as obvious as possible to the user.

StandardizedBow Variables

Bow Skeleton Parts: Joints of the armature that the script uses. They must be assigned. You can do it manually or if you have followed the tutorial just press the find bow rig joints button in the inspector.

Bow Joints Related: Which axis direction the joints should be rotated. Each one has its own specific one. They can have different axis values because of the way they are modeled in the modelling software. If you followed my tutorials, just leave them as default or play around with them and find the right one. Bend angle limits are pretty self-explanatory. They determine how much the joints will be rotated. And inverse bend bool, rotates them on the opposite direction.

Bow String Related: Same axis choice for the string. String Move Amount determines how far the string is going to be pulled. String Amount Time is the time amount it takes to reach that point. It also affects the speed of the joint rotation. Inverse pull bool, moves the string on the opposite direction.

Bow Physics: String Movement Choice is the easing choice for the string movement and joint rotations. The choices are self-explanatory. Try them and find the best one that suits you but I would recommend the medium stable or fast start.

String Retract Choice is the easing choice for the backwards string movement and joint rotations. You can choose elastic or bounce. I would recommend elastic since this is a string.

Max String Stress is the value of the maximum stress built in stress. Longer you hold the string, a variable named currentStressOnString gets higher and stops when reaching this value. It affects how strong the projectile will be fired. If you are doing a game where projectiles deal damage, you can also use that value to interact with the damage.

Projectile Accuracy is self-explanatory too. Unstable will fire off projectiles that are distorted from their straight trajectory and perfect accuracy will always fire projectiles in a straight line.

Projectile Related: Projectile is the prefab that has the StandardizedProjectile script attached to. Projectile Pool Size is the size of the object pooling. It is done in order to avoid Instantiating projectiles in run time. Assign this value according to how fast your bow is going to be firing off projectiles. Particles of the projectiles are pooled with them too.

Projectile Hold & Rot Offset are the offset Vector3 of the projectile when it is trying to follow the string. If your projectile model's origin point is not on the place where you actually hold them, you should tweak these values to spawn the projectile in the correct position.

Projectile Forward Axis shouldn't be changed unless you see that your projectile is being fired off to the incorrect side. It happens because that your projectile model doesn't look at the Z-Axis which is the forward axis. And there is no clean way of changing the axis of a model in Unity so be careful. If you can't change the axis in a modelling software, read this post from Unity: https://docs.unity3d.com/Manual/HOWTO-FixZAxisIsUp.html

Bow Sound Utilities: Sound Volume of the pull and retract sounds are determined by a slider. If you want the stress built in string to affect the sound then you can set the StressEffectOnSound bool correct, but it might give bad results depending on the audio clip you are using.

StandardizedProjectile Variables

Time Limit To De Pool: How long should the bow wait before depooling the projectiles that have been fired off. Something around [5-10] is okay.

Effect Of Velocity On Particle Emission: Depending on the velocity that projectile has in the moment of collision, the particles can be emitted much more, creating the feeling of a stronger impact.

Projectile On Hit Particles: These are the default particle effects that is being pooled when a projectile is collided with a object that has the tag of the specific detection tags below. If an object has none of the specific tags decided below, then it doesn't spawn any particles.

Projectile On Hit Sounds: Just like on hit particles, but sound effects.

Projectile On Hit Detection Tags: These are the tags that your projectiles will be looking for when they are collided with some object. All of these tags are converted into hash values so it is not that CPU expensive. I created 4 tags for you as a default, but you can add or remove any other tags you want from script.

StandardizedBowForHand Scripts Variables

There is one important difference with these scripts and the StandardizedBow script. It is the bow user related part.

Bow User Related: First variable is the animator component of the bow user that will be triggering the bow scripts. Second variable is the string pull hand of the bow user. This is a transform variable so that the string knows where to be. Both of them must be manually assigned.

Pull State Layer Index is the index of the Animator Controller Layer where the bow script should be looking to react. First layer is 0. If your shooting animations are in the third tab, then assign 2 as the index. Next we have Pull State Names for the StandardizedBowForHand and Pull Param Name for the StandardizedBowForHandBool scripts.

Pull State Names are the names of the states in your bow user Animator Controller. Whatever name you assigned to the states that you want bow to react, name them accordingly. Using one state is fine too.

Pull Param Name is the name of the boolean parameter that your Animator Controller uses to trigger the shooting animations on your bow user. Name them accordingly.

Last public variable is the Projectile On Hand. If it is true, it spawns the projectile in the string pull hand you assigned above. If it is false, it spawns the projectile on top of the string. Decide according to your needs.

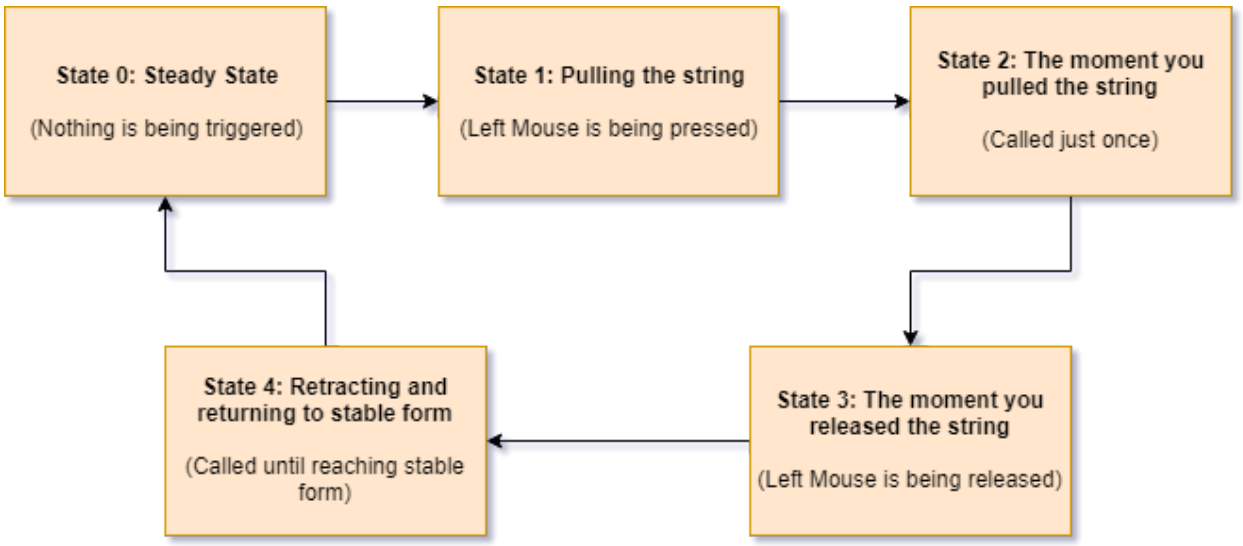
StandardizedProjectileForHand Scripts Variables

There are no differences between them with the regular StandardizedProjectile script but they should be used with their pair bow scripts because of the way that variables are being cached and pooled. So use the one that matches with bow script name.

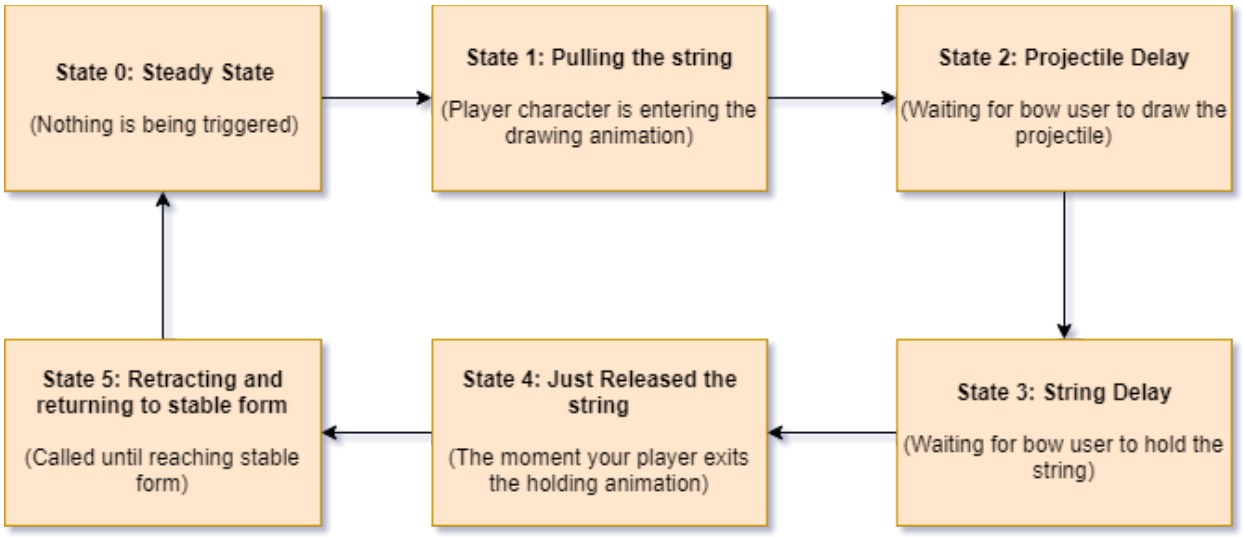
How can I add and remove functionality from this simulation?

The Update function is written in a way that every if condition represents a specific state of the simulation. Which means that even if you don't have the time to read them all, you can just add whatever functionality you want to the specific location and it will be done. They are clearly commented in the script but here is the workflow of the update functions with a graphical representation.

State Structure of the StandardizedBow Script



State Structures of the StandardizedBowForHand Scripts



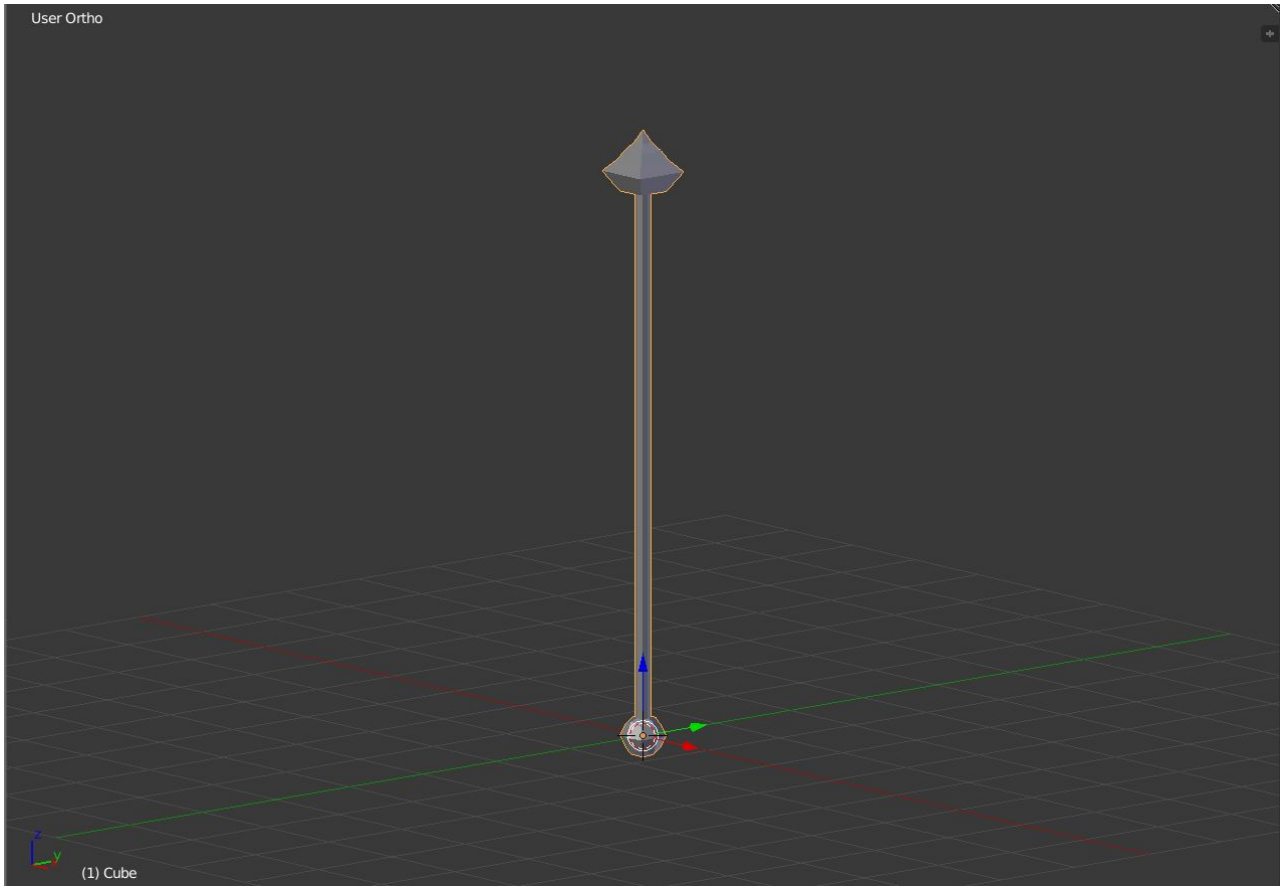
FAQ & Some Advice

My projectile's rotation in air is incorrect. How can I solve this?

This is an issue that happens because your projectile model forward axis is not aligned with the projectile itself. If you have a modelling software like Blender, just fix this issue by rotating the model in edit mode. Otherwise read this post by Unity, there is no other solution.

https://docs.unity3d.com/Manual/HOWTO-FixZAxisIsUp.html

or;



My bow acts weird. What is happening?

It is most likely because of the way it is modeled and the placement of the edges. But first be sure that you correctly assigned the axis values in the bow script. Try bunch of different combinations with different angle limits.

If the bow still acts weird, than you should use a modelling software to try to rig it again and eliminate the mesh parts that you think it might be causing this issue(Like dissymmetrical origin placement and upper-lower half of the bow).

Why do we use another software for this asset?

Even though this asset is in order to standardize the procedure of animating bow models, the procedure itself is bound to a modelling software with some kind of use. For us, it is just creating the same symmetrical rig that is in the center and has 3 bones top, 3 bones bottom and 1 on the string for every bow model that we want to use.

It is much easier since we eliminate the Inverse Kinematics and weight painting parts. It is designed in a way that, people who has no time and experience in these modelling and animation stuff(me) can easily animate a bow model that they liked.

By the time you created this same rig for 3 times and prepared your prefabs, you will see that you can actually complete the whole procedure in 2 or 3 minutes. You can't avoid Blender or Maya completely, so let's make it easy for ourselves :)

Is it really that important what I do the in the modelling software?

There are only two things that you should be careful about. First one is the axis placement. The bow joints can be fixed in Unity but the projectile issue becomes a real headache. It would best for you to remember the axis placements that I explained in the rigging tutorial video, so that the script starts working properly from the get go.

Second one is the scaling. You might see that your objects deform and act weird once you press reset transform in Unity. That is because of the difference in the scale measurement. You can avoid this situation by normalizing them. If you are using Blender, just press Ctrl+A and select Rotation & Scale after you finished completely with your model.

How performant is this script and what type of games I can use it on?

Rotating Skinned Mesh Joints is not the most performant way of achieving this result. I wouldn't recommend you to keep more than 100 active and shooting bows in your current scene. Which means that this asset is not suitable for RTS type of games where every soldier has one of these scripts. It would be very illogical.

Where this script really shines is the customizability aspect of it. You can change the values depending on the stats of your main character or skill you are using like in a RPG game. You can use this for a bow shooting game where you are the only one who is shooting the bow like in a survival game or VR.

In a game like RTS where all of the NPC is repeating the same shooting procedure, you wouldn't use features of this scripts. That requires something like ECS, Job System, Vertex Deformation with Shaders(will talk about it below) or Animation Baking.

Use what you have wisely.

Are there any other updates or features coming to this asset?

Of course there are :) This is something that I have been currently using in my own indie hobby project. So whatever I will probably add into my game, I will most likely share and update it in here too. At least the ones that are going to help other people.

First thing I will be looking into will be the Vertex Deformation with a shader. It would be much more performant and suitable for high amounts of instances. I didn't write it that way thinking that it would be hard for me and users who don't know how to write shaders to get most out of it. Because as long as an asset cannot be integrated into your project, it is a waste of time.

But as I researched into it more, I found some acceptable ways to create procedure that users can still configure the same values we have seen in this script in a more performant way. So, I am currently studying on that. When I have something that is acceptable to share in the Asset Store, I will hopefully add that to this package.

Second thing is the given bow models with the asset package. I just put them as a placeholder models. They are my own models and that's why they look terrible. I am planning to commission an artist friend of mine to create a set of bow and arrow models for this package depending on my financial state which is not very stable :)

Personal Message

Who am I?

Hey, it is the end of the document and only now I am introducing myself. How rude. My name is Ahmet Taha Aydemir. It connects perfectly so I use my name as Ahmetahaydemir. I am an industrial engineer.

Coding is not that involved in our professional lives. We mostly deal with mathematical programming stuff and optimization algorithms. But coding is something that I have been doing ever since I was a child because I thought it was cool. And now, here we are.

Why did I published this?

As I said above, I am currently developing an indie hobby project. This asset is something that I am currently using for my own project. I thought that by sharing it, I might help someone else's project and help myself financially.

I have always been fascinated with RPG games like Morrowind, Vampire: The Masquerade – Bloodlines, Neverwinter Nights, and Dragon Age: Origins. So I am trying to create a game that I can see in the Steam page and be proud that some other person in the other end of the world maybe played my game and felt something similar to what I felt when I was a child.

I know that it is delusional to think that someone can achieve by themselves what those studios have created with hundreds of talented people. But most of the indie game developers that I have met or seen are pretty delusional, so I guess it is just in our nature :)

I would be stoked to see my asset being used in some kind of project. If you happen to do, I would love to see or play it. Good luck on your projects.

Thanks for your interest

I started to form this package after I asked whether if anyone would be interested in this asset. After seeing the positive comments, I decided that I should publish it. ~~I want to thank everyone that showed support. I hope that 5$ is not that much :) I didn't want to make it free for some reasons and 5$ is the lowest price, so I hope you can understand.~~ Here is the post:

(It is open source now, https://github.com/ahmetahaydemir/StandardizeBows)

https://www.reddit.com/r/Unity3D/comments/aprz21/since_i_have_no_experience_in_animation_i/

Contacts

Again, thank you for purchasing my asset. I know it is not perfect, but I tried my best and put lots of time in it. And I am planning to put even more :) Have a good day.

If you want to contact me or need support:

Personal Email: ahmetahaydemir@gmail.com
Support Email: ahmetahaydemirian@gmail.com

Youtube: https://www.youtube.com/channel/UCg0QBQFEAHstDN3rXLveg5w

PayPal Link (I finally got one, any support is appreciated:)) : **paypal.me/ahmetahaydemir**

Since you have managed to finish reading all, I won't turn on the tap.

↓