December 13, 2023

Time

9AM PST

Agenda

- New Unity plugin
- Any follow up discussion on the wonderful presentations shown last month from @Alan Kent and @Kev Kirkland
- Spline Animation Proposal : Discussion on assembling a character in USD, this is an interesting topic that @Frieder Erdmann brought up the other week.

Notes

Unity Plugin

https://forum.unity.com/threads/new-openusd-packages-now-available.1524583/

- USD Importer
- USD Exporter
- USD Core package

It was a plugin package Unity. The Team rewrote the thing from scratch!

Binding C# code is not yet open source.. Unity worked a lot to improve them.

23.02 is the USD that it was based on.

Q&A

Regarding the assets, we need to check what,s a game asset look like.
We need an atomic unit.
We also need large scenes.

A bigger discussion need to be done to be sure to have the right assets.

We need to find the more compelling way to make USD a good format for games.

How is the integration done?

It is done mainly done in Importer.

An underlying question is USD is a good format for runtime?

Another question is : how to support mobile etc

Another area that needs to be discussed is the roundabout and interchange.

How do you handle levels?

Unity tried to stick to Vanilla USD. There is no custom schemas support.

Is the USD turned into native internal structures, or is the USD format kept and adjusted directly?

USD are converted to prefab game objects

Is the file watcher aware of the references between file?

It keeps tracks of assets in the asset folder.

Are you allowed to share a roadmap?

No dates are shareable.

Unity needs to check how many people will need USD as a runtime format.

Unity is very open to ear from the community.

They need to check the adoption.

Some questions are : is USD suited for my project, Can I do animation on USD, etc

Any follow up discussion on the wonderful presentations shown last month from @Alan Kent and @Kev Kirkland

Spline Animation Proposal : Discussion on assembling a character in USD, this is an interesting topic that @Frieder Erdmann brought up the other week.

https://academysoftwarefdn.slack.com/archives/C03GKF4DG7K/p1700757450973849

Hello! I'm looking into something that I'm assuming most game studios have (outside of USD) already in their pipelines and trying to see if this would map well or less well into USD:
To assemble a character, we have usually three file types:
A character file with the base skeleton with the relevant bones/joints required by the default animation system (root, hips, legs, arms, head)
N number of attachment files, these usually contain Geometry and the part of the base skeleton that they skin to, as well as sometimes additional bones/joints, that can be driven by secondary animations.
Some sort of manifest file that tells the engine to load the base skeleton with a number of attachments.
The process in the engine (at runtime) then takes the manifest file and creates an assembled character from the character file with the different attachment files merged together into one.
Ideally I'd like to achieve something similar in USD in order to give proper preview to Animators in the DCC of their choice. We have done this in the past in different DCCs with either
loading the different (FBX at the time) files and constraining them together (bone by bone) (maintaining the individual pieces, but slow to evaluate in the DCC) or
having a pre-processor merge all (FBX again at the time) files together and then load the FBX data into the DCC in a lossy process (but faster to work with in DCC)
With USD's referencing and opinion workflows, I was hoping that there would be some clever way to achieve this result, but so far, I'm struggling to put something coherent together - hence my message here: Has anyone already looked at such a workflow and what was your approach?

Charles Flèche
19 days ago
As far I know you can't list list-edit an attributes, so generating the full set of bones can't be done with USD composition itself.
Maybe that could be done with additionnal logic ?
Variants per bone set contains contains a prim that has an attribute: a list of bones
When a variant is activated it also add the relationship a custom rel attribute to your Skeleton
on recomposition the attribute that contains the rel to Prim that contain bones is notified to have changed
a script rebuilds the full skeleton
(edited)
:+1:
1

Charles Flèche
19 days ago
I am not super clear...

Charles Flèche
19 days ago
Let me see if I can make a quick example

Charles Flèche
19 days ago

```
#usda 1.0
def SkelRoot "SkelRoot" (
  variants = {
    string tool = "hammer"
    string shoe = "trainer"
  }
  prepend variantSets = ["tool", "shoe"]
)
{
  def Skeleton "Skeleton"
  {
    uniform token[] joints = []
    custom token[] main_joints = ["root", "root/hand", "root/foot"]
    custom rel addons
  }

  variantSet "tool" = {
    "hammer" {
      over "Skeleton" {
        append rel addons = </SkelRoot/Skeleton/Hammer>

        def "Hammer" {
          custom token[] addon_joints = ["root/hand/hammerbase"]
        }
      }
    }
    "compass" {
      over "Skeleton" {
        append custom rel addons = </SkelRoot/Skeleton/Compass>

        def "Compass" {
          custom token[] addon_joints = ["root/hand/hinge", "root/hand/hinge/left", "root/hand/hinge/right"]
        }
      }
    }
  }

  variantSet "shoe" = {
    "trainer" {
      over "Skeleton" {
        append rel addons = </SkelRoot/Skeleton/Trainer>

        def "Trainer" {
          custom token[] addon_joints = ["root/foot/trainer"]
        }
      }
    }
    "robotBoot" {
      over "Skeleton" {
        append custom rel addons = </SkelRoot/Skeleton/RobotBoot>

        def "RobotBoot" {
          custom token[] addon_joints = ["root/foot/boot", "root/foot/boot/hinge", "root/foot/boot/hinge/tip"]
        }
      }
    }
  }
}
```

Charles Flèche
19 days ago
image.png

Charles Flèche
19 days ago
Don't know if it is clearer, but at list if you open the file above in usdview you should be able to switch variants and see /SkelRoot/Skeleton. addons to be update with the list of addons prim. Those prim contains a list of addon_joints , so with a notification handler you should be able to recompose /SkelRoot/Skeleton.joins on top of the base skeleton bones defined in /SkelRoot/Skeleton.main_joinrs

Charles Flèche
19 days ago
IIRC there was a discussion on the previous Google List on allowing list-editing uniform (non time sampled) Attributes. That definitely could be useful for this kind of workflow, as Skeleton.joints is uniform, so we could leverage's USD list-editing operations directly.

Koen Vroeijenstijn
18 days ago
When I tried to represent attachments elegantly, I ran into the issue that there is no parenting to bones. As all the joints are represented as a single prim, you cannot parent the attachment to a specific joint (connect the neck joint of the body to the neck joint of the body for example to swap out heads). For static models, I just apply the transform of the "attach to" joint to the attachment, if you then convert to native geometry, you can easily create a parent contraint, but it's not ideal. I played around with merging the skeletons, but then updating all the skin weights for the meshes gets messy. Maybe we can have a very simple "parent joint constraint" or perhaps a more expensive representation of a usdskel which does unroll into the regular scene graph. Hope I remember right, it was a while ago I looked at this. Curious if Charles' suggestion works for you, please keep us posted. (edited)
:+1:
1

Alan Kent
15 days ago
A standard way to do parenting to bones, or a "UsdSkel as prims" would be nice. E.g. how to put an arbitrary hat on a head?

Typical problem is : when you have extra joints in the jacket, how do you simulate these joints.

USD does not allow to do extra operation on the Skeletal.

Koen Vroeijenstijn haven't found yet a good way. A simple constraint would be good for games.

TJ worked on a schema for that but wasn't able to finish his work.

For Frieder, constraint is one solution. In MoBu and Maya become heavy resources
Another solution is to have one skeleton.
A very good solution would have the ability to reference another skeleton to evaluate one hierarchy.

For animation is very important to have all in one system.

Setting up something work to create asset for animation. TJ will check for that.

Attendance

- [x] François Devic, Co-Lead
- [x] TJ Trently, Co-Lead, Firewalk
- [ ] Alex Schwank - WG Co-chair, Apple
- [ ] Nick Porcino - WG Co-chair, Pixar
- [ ] Michael Min - USD Camera WG, Netflix
- [ ] Roman Zulak - USD on the web WG, NVIDIA
- [x] Aaron Luk, NVIDIA
- [ ] Adam Harder
- [ ] Alan Blevins, NVIDIA
- [ ] Alessandro Bernardi - Ubisoft - HELIX Studio
- [ ] Alex Gerveshi, AWS
- [ ] Alex Wilkie
- [ ] Alexander Kalyuzhnyy, Wizart Animation
- [ ] Allen Hastings, Foundry
- [ ] Aloys Baillet, Animal Logic
- [ ] Alson Entuna, Crytek
- [ ] Alyssa Reuter
- [ ] Andy Beers
- [ ]

- [ ] Andy Biar, Warner Bros.
- [ ] Ana Gomez
- [ ] Anandhaiyappan, Botvfx
- [ ] Angelo Gabriel Sapaula
- [ ] Anthony Tan, Autodesk
- [ ] Anton Palmqvist
- [ ] Arash Keissami, Nira.app
- [ ] Arielle Martin, Foundry
- [ ] Ashwin Bhat - USD and MaterialX, Autodesk
- [ ] Barry Ruff
- [ ] Ben Chung-Hoon, NVIDIA
- [ ] Ben Deniz
- [ ] Bernard Kwok, Autodesk
- [ ] Bill Dwelly
- [ ] Bill Spitzak, Dreamworks Animation
- [ ] Blazej Floch
- [ ] Brian Gyss, 5th Kind
- [ ] Bruno Ebe
- [x] Bruno Landry (Unity)
- [ ] Carlos Felipe Garcia Murillo
- [ ] Carolin Colón
- [ ] Carson Brownlee, Intel
- [ ] Charleen Chu, SPI
- [ ] Charles Flèche, Ubisoft Montréal
- [ ] Chris King
- [x] Christopher Lexington
- [ ] Chris Rydalch, SideFX
- [ ] Claire Chen
- [ ] Claire Yb
- [ ] Claude Robillard
- [ ] Connor Smith, Magic Leap
- [ ] Corey Revilla
- [ ] Cory Omand, TWDS/Pixar
- [ ] Curtis Andrus
- [ ] Dan Herman
- [ ] Dan Lee
- [ ] Dan Rolinek
- [ ] Daniel Heckenberg, Animal Logic
- [ ] Daniel Lanner
- [ ] Dave Hale, Riot Games
- [ ] David Aguilar, Walt Disney Animation
- [ ] David Larsson, Adobe
- [ ] Dean Jackson, Apple
- [ ] Deke Kincaid, Digital Domain
- [ ] Dhruv Govil, Apple
- [ ] Divyansh Mishra
- [ ] Diya Joy

- [ ] Domenico Alessi
- [ ] Dominic Couture
- [ ] Doug MacMillan, Tippett Studio
- [ ] Edward Slavin, NVidia
- [ ] Élie Michel
- [ ] Eric Chadwick, Wayfair
- [ ] Eoin Murphy, NVidia
- [ ] Eric Enderton, NVidia
- [ ] Eric Majka, Psyonix/Epic Games
- [ ] Erik Ostsjo
- [ ] Étienne Archambault
- [ ] F. Sebastian Grassia, Pixar
- [ ] Fabrice Macagno, Animal Logic
- [ ] Felix Herbst, Prefrontal Cortex
- [ ] Fernando Leandro
- [ ] Francois Lord, NAD-UQAC / Rodeo FX
- [x] Frieder Erdmann, Ubisoft Massive
- [ ] Gary Jones, Foundry
- [ ] Geoff Evans, NVIDIA
- [ ] Georgie Challis
- [ ] Gordon Bradley, Autodesk
- [ ] Gordon Cameron, Epic Games
- [ ] Guido Quaroni, Adobe
- [ ] Guillaume Laforge, Autodesk
- [ ] Guy Martin, NVIDIA
- [ ] Hamed Sabri
- [ ] Hendrik Helpach
- [ ] Henrik Edstrom, Autodesk
- [ ] Henry Vera, DNEG
- [x] Ife Olowe
- [ ] James Pedlingham, Foundry
- [ ] Jason Rosson
- [ ] Jeff Bradley, Dreamworks
- [ ] Jenna Bell, Disney / Invisible Thread
- [ ] Jennifer Horowitz, Maxar
- [ ] Jenny Zhang
- [x] Jeremiah Zanin, Santa Monica Studio
- [ ] Jeremy Cowles - USD Assets WG Chair, Valve
- [ ] Jerran Schmidt, NVIDIA
- [ ] Jerry Gamache
- [ ] Jesse Barker
- [ ] Jesse Ng, Metropolitan Museum of Art
- [x] Jessica Wang, Pixar
- [ ] Joe Hultgren
- [ ] John Burnett, Bonfire Studios
- [ ] John Creighton, Apple
- [ ] John Hood, SPI

- [ ] John Mertic, Linux Foundation
- [ ] Jon Creighton, Apple
- [ ] Jon Wade, Spotify
- [ ] Jonah Friedman, Autodesk
- [ ] Jonathan Gerber
- [ ] Jonathan Stone
- [ ] Jonghwan Hwang
- [ ] Jordan Soles, Rodeo FX
- [ ] Jordan Thistlewood, Epic
- [ ] Joshua Miller
- [ ] Joseph Goldstone
- [ ] JP Mackel
- [x] JT Nelson, Pasadena Open Source Consortium/SoCal Blender group
- [ ] Julien Dubuisson
- [x] Kev Kirkland
- [ ] Kevin Bullock
- [ ] Kelvin Chu, Riot Games
- [ ] Kimball Thurston, Weta
- [x] Koen Vroeijenstijn, Activision / Infinity Ward
- [ ] Kristof Minnaert, Remedy Entertainment
- [ ] Kurtis Schmidt
- [ ] Laura Scholl
- [ ] Larry Gritz, SPI
- [ ] Lee Kerley, SPI
- [ ] Levi Biasco, Santa Monica Studio
- [ ] Louis Marcoux, NVIDIA
- [x] Lucas Morante, Illusorium
- [ ] Luca Scheller, RiseFX
- [ ] Luiz Kruel, R* NYC
- [ ] Luke Titley
- [ ] Manuel Köster, Crytek
- [ ] Mark Alexander
- [ ] Mark Elendt, SideFX
- [ ] Mark Final, Foundry
- [ ] Mark Masson
- [ ] Mark Manca
- [ ] Mark Sisson
- [x] Mark Tucker, SideFX
- [ ] Marteinn Oskarsson, Sony Imageworks
- [ ] Martin Karlsson
- [ ] Mathieu Bertrand
- [ ] Mathieu Mazerolle, Foundry
- [ ] Matias Codesal, NVIDIA
- [ ] Matt Johnson, Epic Games
- [ ] Matt Kuruc, NVIDIA
- [ ] Matthew Levine, WDAS
- [ ] Matthew Low, DWA

- [ ] Michael B. Johnson, Apple
- [ ] Michael Blain, Unity
- [ ] Michael Buckley
- [ ] Michael Kass, NVidia
- [ ] Michael Min
- [ ] Mika Vehkala, Remedy Entertainment
- [ ] Mikko Haapoja, Shopify
- [ ] Nat Brown
- [ ] Natasha Tatarchuk, Unity
- [ ] Neil Chodorowski
- [ ] Niall Redmond, Foundry
- [ ] Nicolas Popravka, Soul Machines
- [ ] Nicolas Savva
- [ ] Nishanth Singaraju
- [ ] Nishith Singhai
- [ ] Oliver Dunn
- [ ] Oscar Sebio, Autodesk
- [ ] Paolo Selva, Weta
- [ ] Paul Baaske, Jellyfish Pictures
- [ ] Paul Molodowitch, NVIDIA
- [ ] Patrick Palmer
- [ ] Peter Arcara
- [ ] Pete Segal
- [x] Phil Sawicki, NVIDIA
- [ ] Prapanch Swamy, Disney / Invisible Thread
- [ ] Pier Paolo Ciarravano, MPC
- [ ] Pierre-Luc Bruyere
- [ ] Quentin Birrer
- [ ] Ramesh Santhanam
- [ ] Rebecca Hallac
- [ ] Richard Frangenberg
- [ ] Richard Kerris, nVidia
- [ ] Richard Lei, Weta
- [ ] Rob Pieké
- [ ] Rob Stauffer, SideFX
- [x] Robert Krupa, Elemental Games
- [ ] Robin Rowe, CinePaint
- [ ] Rohit Khonde
- [ ] Rory Woodford, Foundry
- [ ] Ryan Stelzleni
- [ ] Scott Geffert, Metropolitan Museum of Art
- [ ] Scott Keating
- [ ] Sean Looper, AWS
- [ ] Sean McDuffee, Intel
- [ ] Seb Schmidt, Weta
- [ ] Sebastian Herholz, Intel
- [ ] Sebastian Grassia, Pixar

- [ ] Sebastian Rath, Snowtrack Montréal
- [ ] Sebastien Dalgo, Unity
- [ ] Sergei Shaykin, Apple (usdzconvert)
- [x] Sergio Rojas, Different Dimension
- [ ] Serguei Kalentchouk, Netflix
- [ ] Shane Davis, SideFX
- [ ] Shawn Dunn, Epic Games
- [ ] Simon Haegler, Esri
- [ ] Silvia Palara
- [ ] Sneha Jaikumar
- [ ] Spencer Luebbert
- [ ] Stefan Habel, Foundry
- [ ] Stephan Leroux, Shopify
- [ ] Steve Agland, Animal Logic
- [ ] Steve Hwan, DD
- [ ] Steve LaVietes
- [ ] Steven Anichini, Disbelief
- [ ] Sue Sauer, Sunrise Productions
- [ ] Sylvain Trottier, NVIDIA
- [ ] Thibault Lambert
- [ ] Thomas Chollet
- [ ] Thomas Kumlehn
- [ ] Tiago Carvalho
- [ ] Tim Armstrong
- [ ] Tim Fowler
- [ ] Tzung-da Tsai
- [ ] Vadim Slyusarev
- [ ] Varun Talwar
- [x] Wayne Wu
- [ ] Will Telford, NVIDIA
- [ ] Xiaoxi Liu, Unity
- [ ] Yassine Mankai
- [ ] YJ Jang