Teaching Core Disciplines Through Game Development

Carl Callewaert
Unity Product Evangelist

Class Summary

Games and applications, especially those on continually evolving mobile devices, now occupy an increasing percentage of a student's day. At the same time, performance in math and engineering for that same demographic is dwindling at an alarming rate. To compete in the future and secure promising opportunities, the latter (but not necessarily the former) must change. This class will offer ways to increase class engagement while buildingthe types of games and applications almost certainly found in nearly every student's pocket.

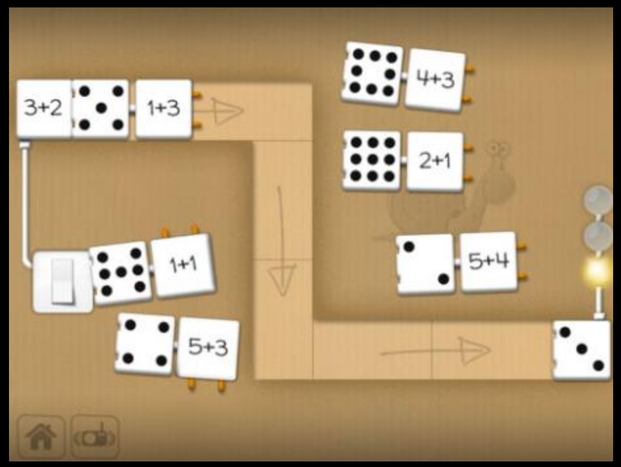
Learning Objectives

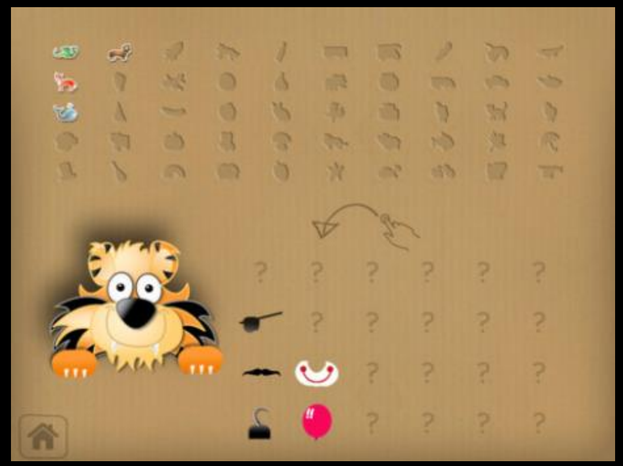
At the end of this class, you will be able to:

- Explain the content game development process for in STE(A)M curriculum
- List the step to create interactive content for STEAM curriculum
- Recall examples of succesful STEAM students projects
- Identify the power of each Autodesk and Unity tool to use in STE(AM) curriculum

Examples of Teaching Core Disciplines Through Game Development

Examples





Examples

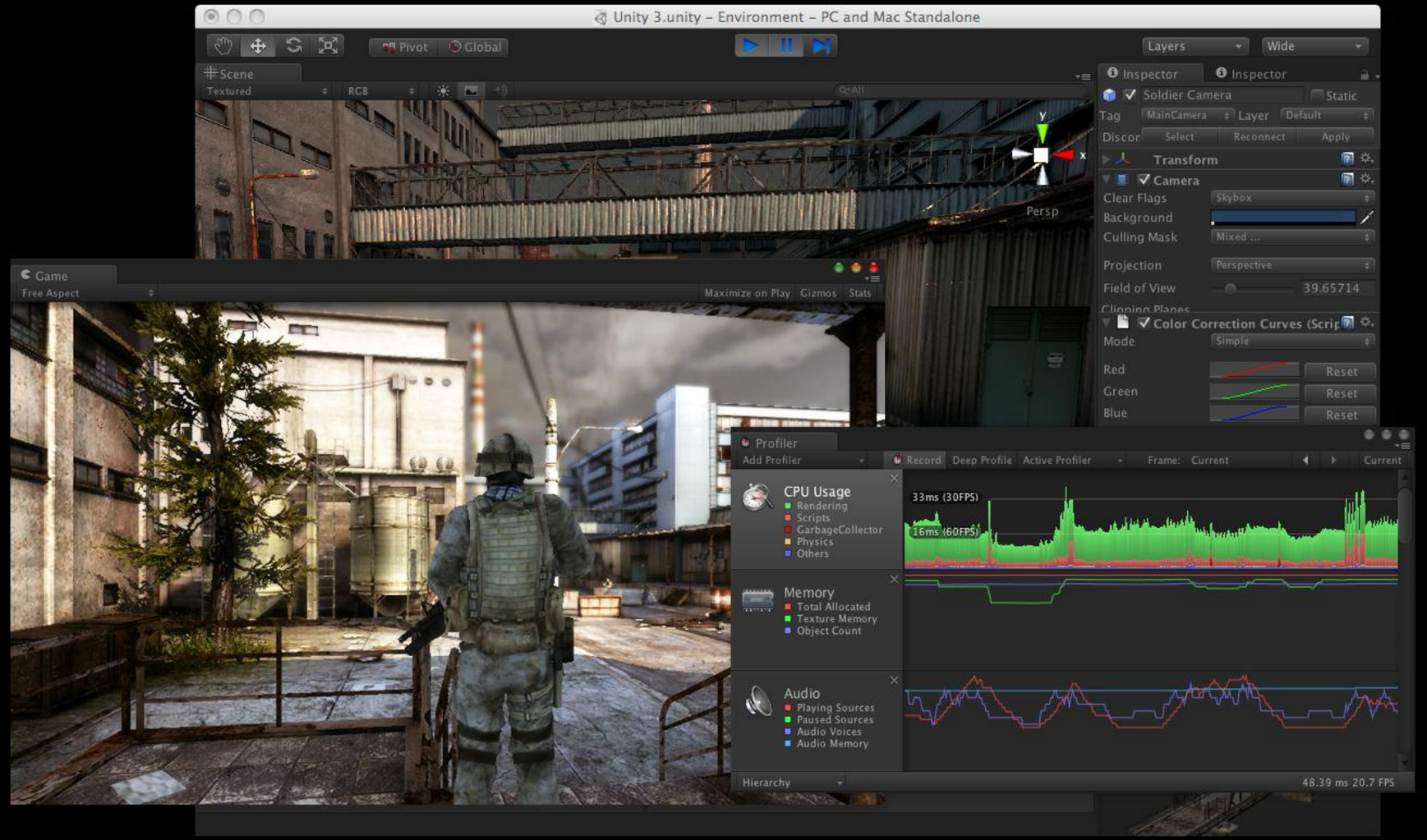
Band Together – Backflip Studios

What is Unity

What is Unity

What is Unity

What is Unity



What is Unity

What is Unity

Developers!!!

1,200,000 developers
- 300,000 monthly active
- 53% of mobile game developers
- 2% of developers of any kind

Develop Award: Best Engine

What is Unity

Shadowgun

Dead Trigger

CSR Racing

Kingdoms of Camelot

Ski Safari

Triple Town

Beat Sneak Bandit

Road Warrior Racing

What is Unity

Captain Antarctica

Jelly Defense

Temple Run

Heroes Call

Frontline Commando

Subway Surfers

Blood & Glory

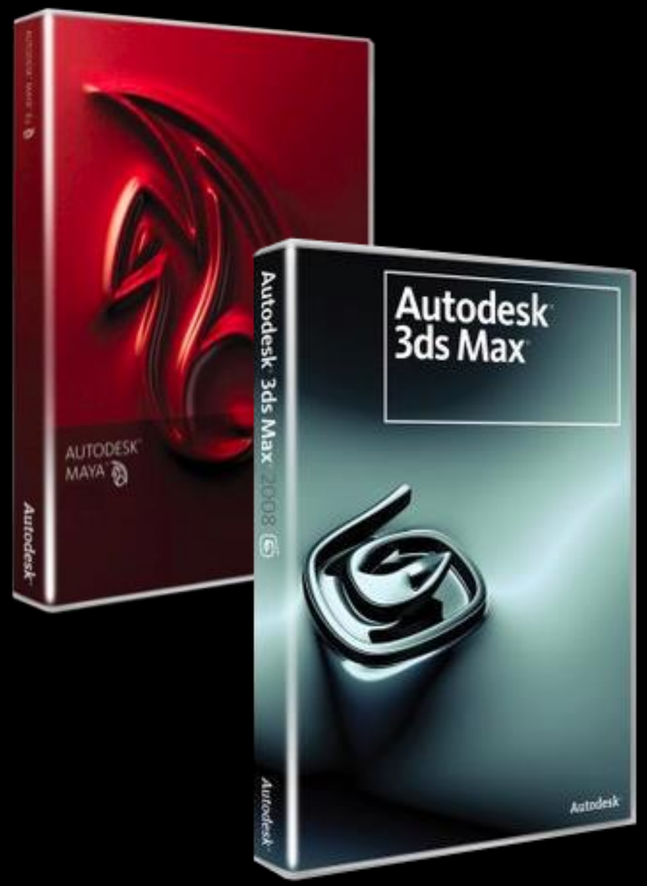
What is Unity

What is Unity

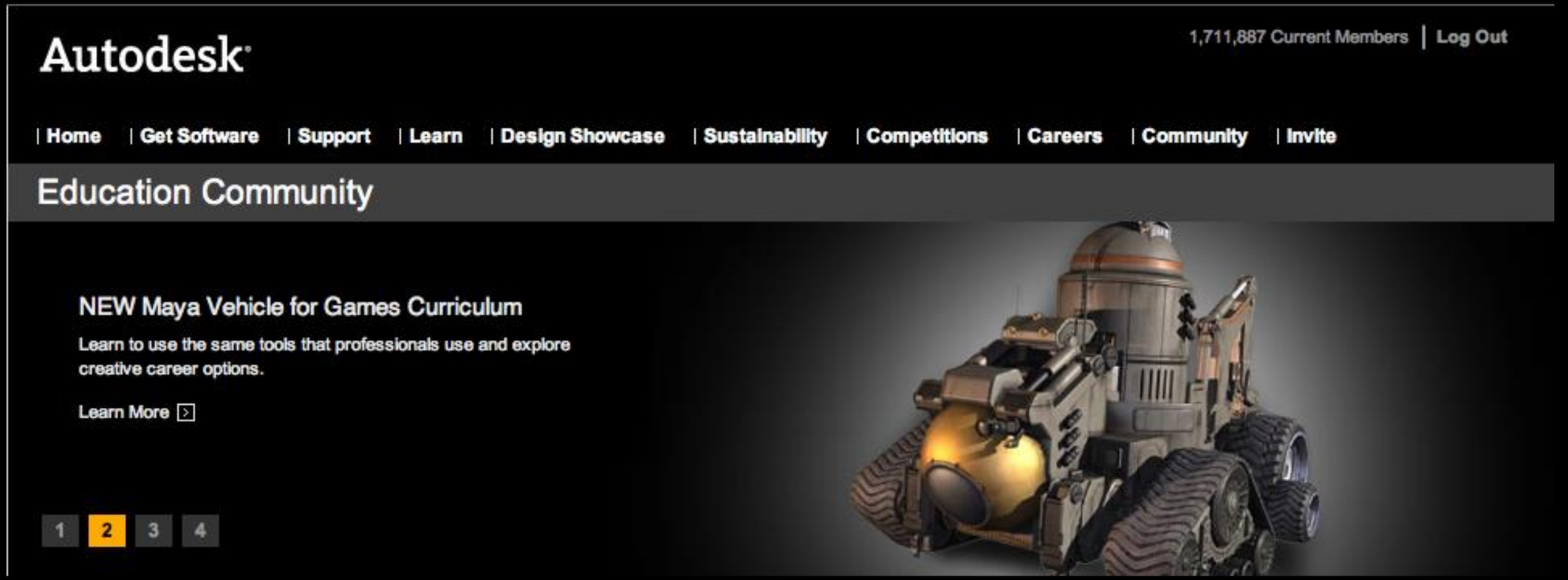
Unity & Autodesk

Unity & Autodesk

- Streamed lined Asset Pipeline
- Drag and drop Pipeline with Revit, 3dsMax and Maya
- Live-link to content tools (auto updates)
- Support of animation, riggings, mesh,..
- Assets Server gives teams the ability to
  - collaborate on project while maintaining
  - Version control and asset history

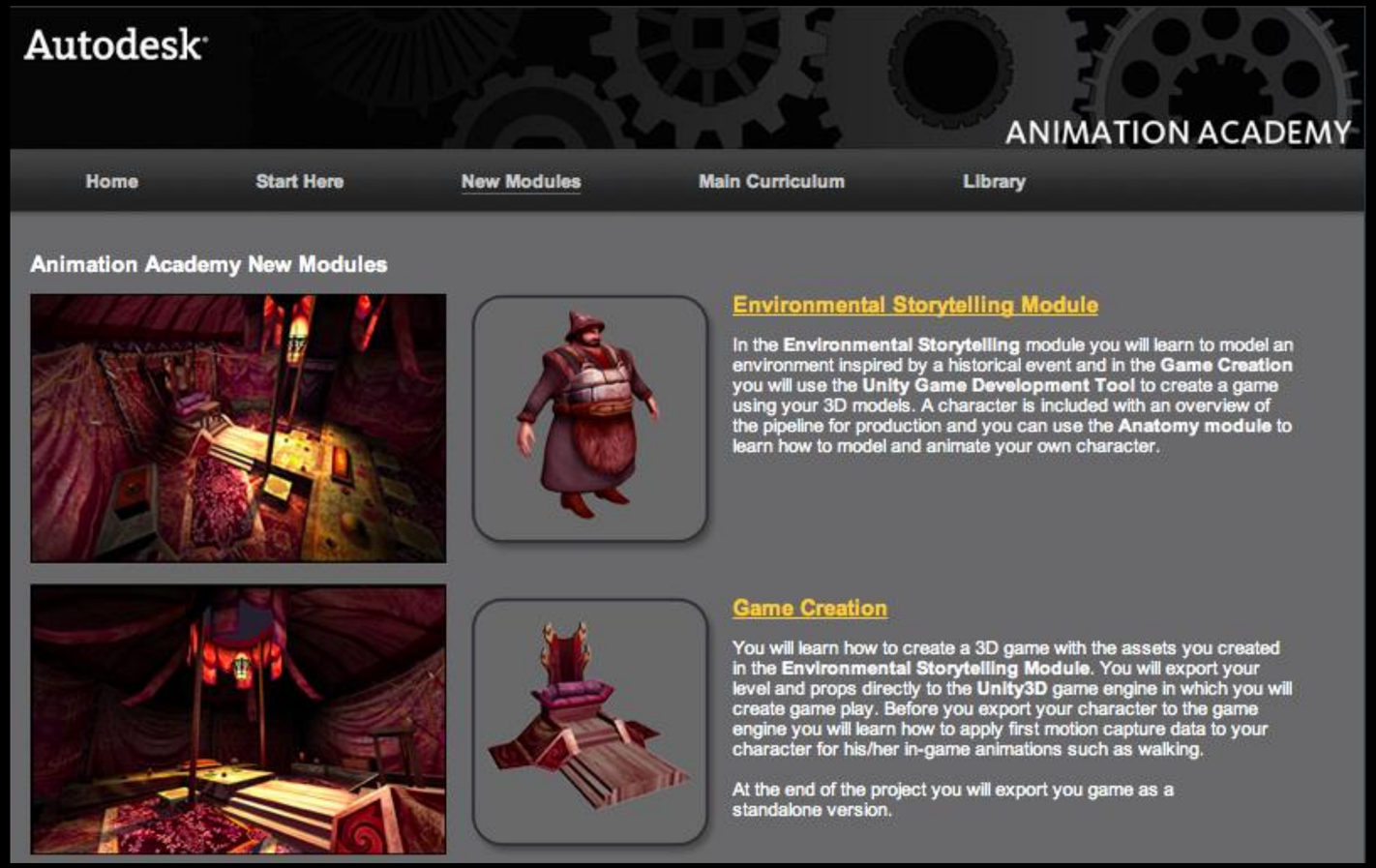


Unity & Autodesk



http://students.autodesk.com

Unity & Autodesk



Unity & Autodesk

Autodesk

ANIMATION ACADEMY

Home | Start Here | New Modules | Main Curriculum | Library

Game Creation | Video Tutorials

Overview of the Project

You will learn how to create a 3D game with the 3D assets you have created in the previous class. You will export your level and props directly to the Unity3D game engine in which you will create the game play. Before you export your character to the game engine you will learn how to apply first motion capture data to your character for in-game animations such as walking. Then you will import your character in the game engine and apply instructions to your character to be able to walk around and pick up object to collect points. On the end of the project you will export you game as a standalone version.

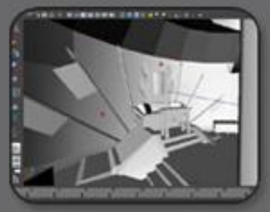
Download All Videos

Movie 01 - Play

This movie files shows the overview and the final result of the project and what you will be learning to achieve the final result.

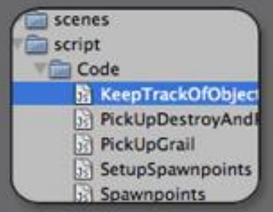
Movie 07 - Play

This movie file explains the steps to make the 3D character move in the 3D level.



Movie 02 - Play

This movie file explain the process of exporting the 3D models (the character, level and props) to file format that can be later on imported in the game engine.



Movie 08 - Play

This movie file explains how to create the game play for Attila the Hun.

Unity & Autodesk

Dinner with Attila
Student Example

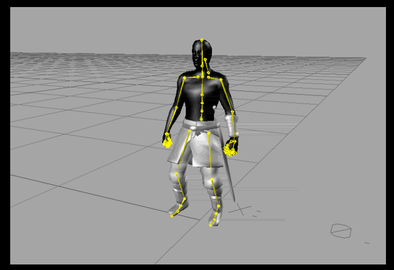
Unity & Autodesk

Dinner with Attila: Dev Stats

- 3 students
- Maya, 3ds Max, Mudbox, MotionBuilder
- Unity
- 1 month
- No programming background
- Mocap by Mixamo & Measurand

Unity & Autodesk: STEAM

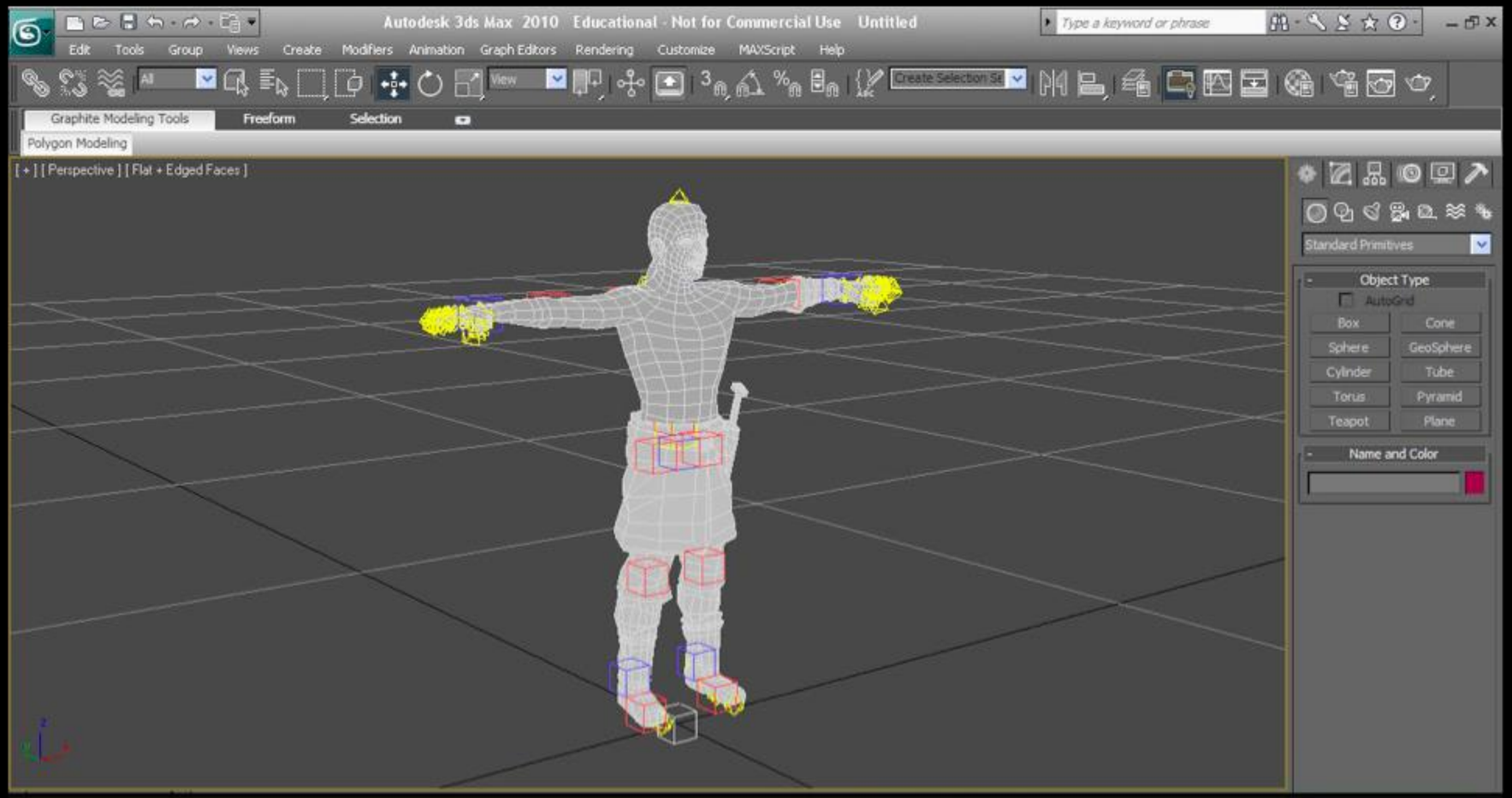
- 3ds Max
- Maya
- MotionBuilder



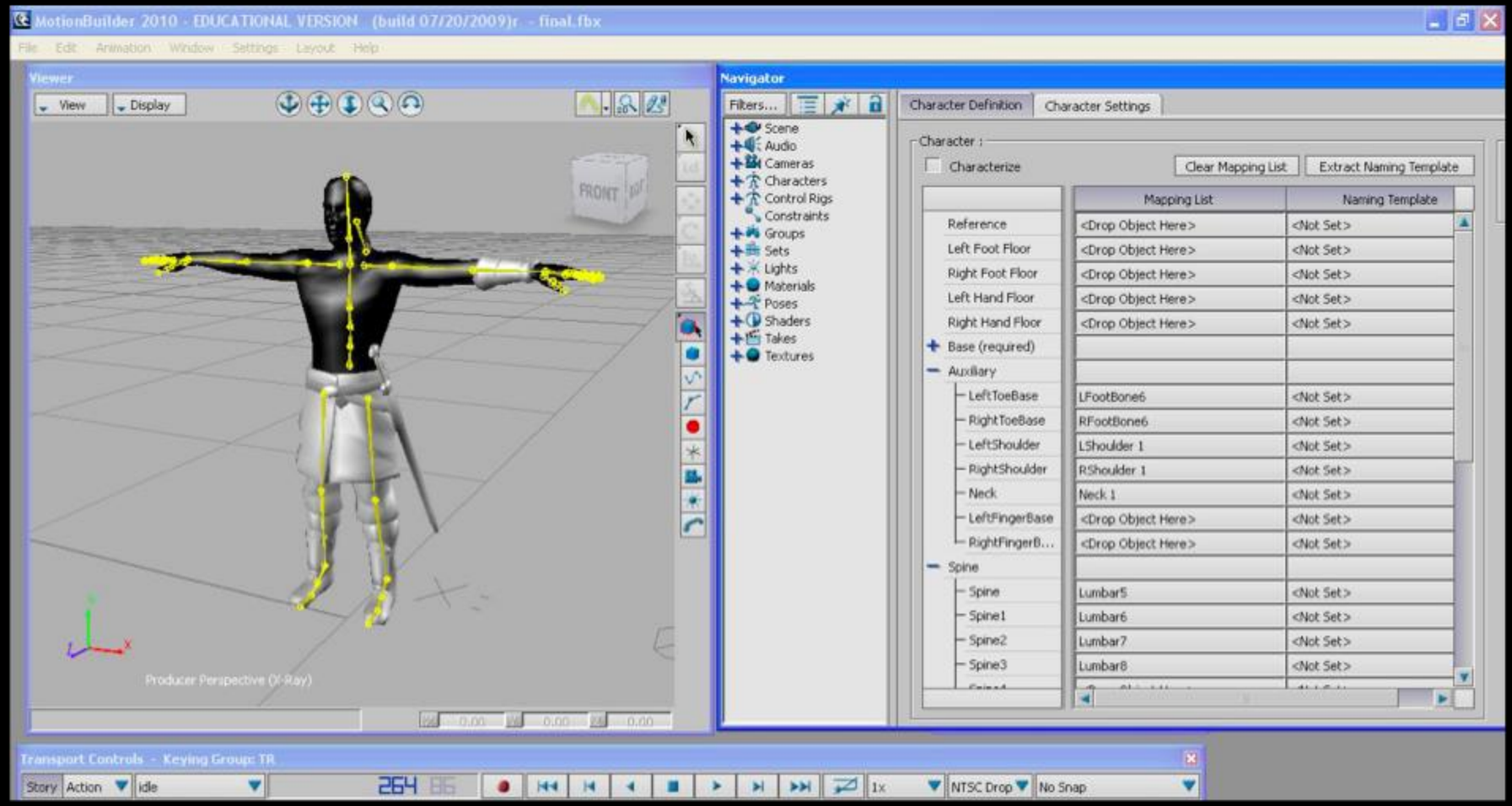
Unity & Autodesk: STEAM

Unity & Autodesk: STEAM

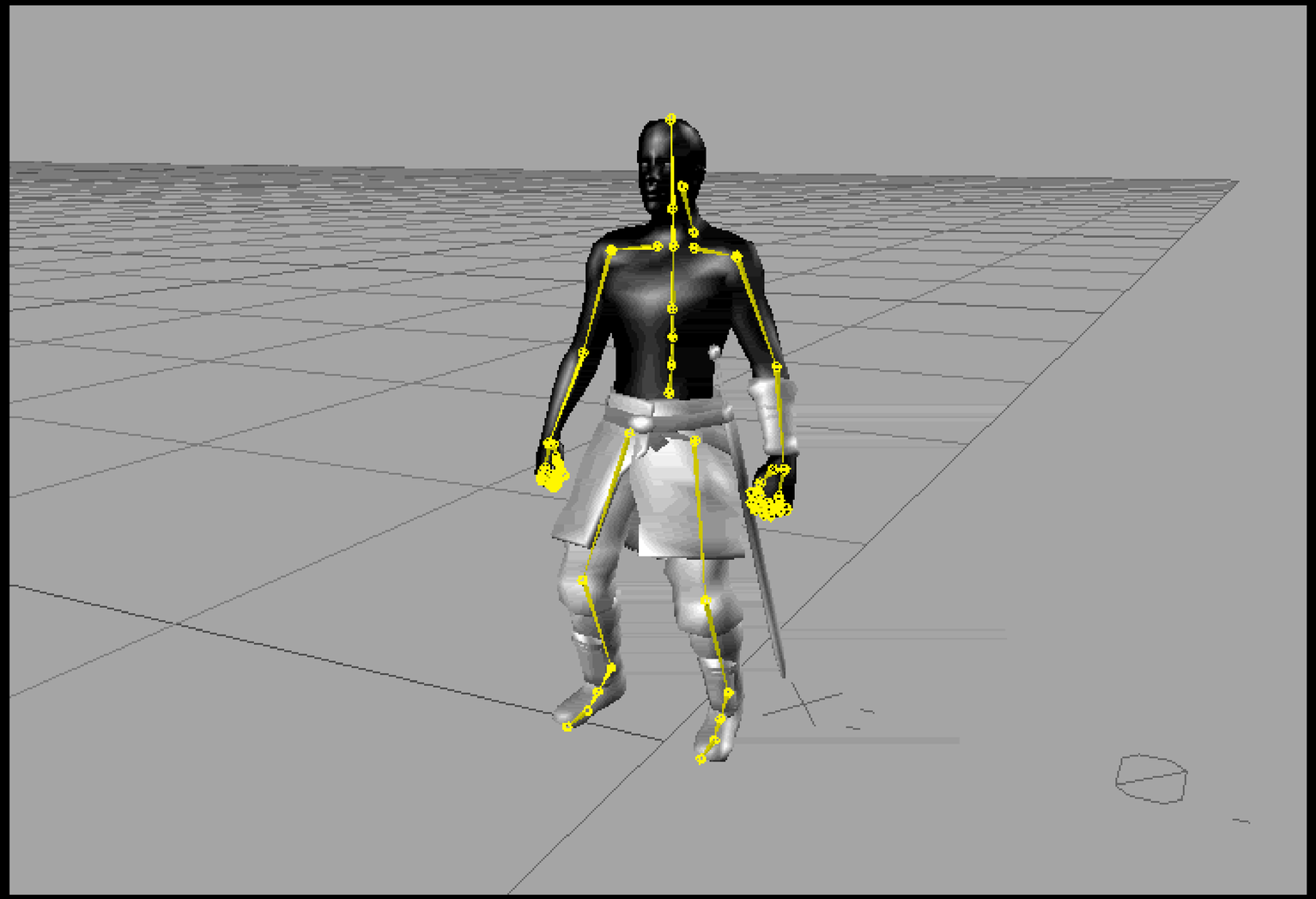
Unity & Autodesk: STEAM



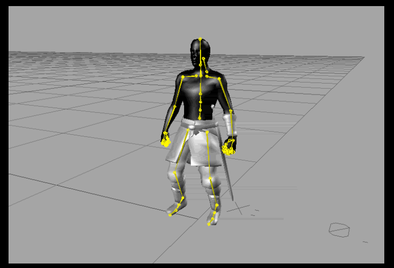
Unity & Autodesk: STEAM

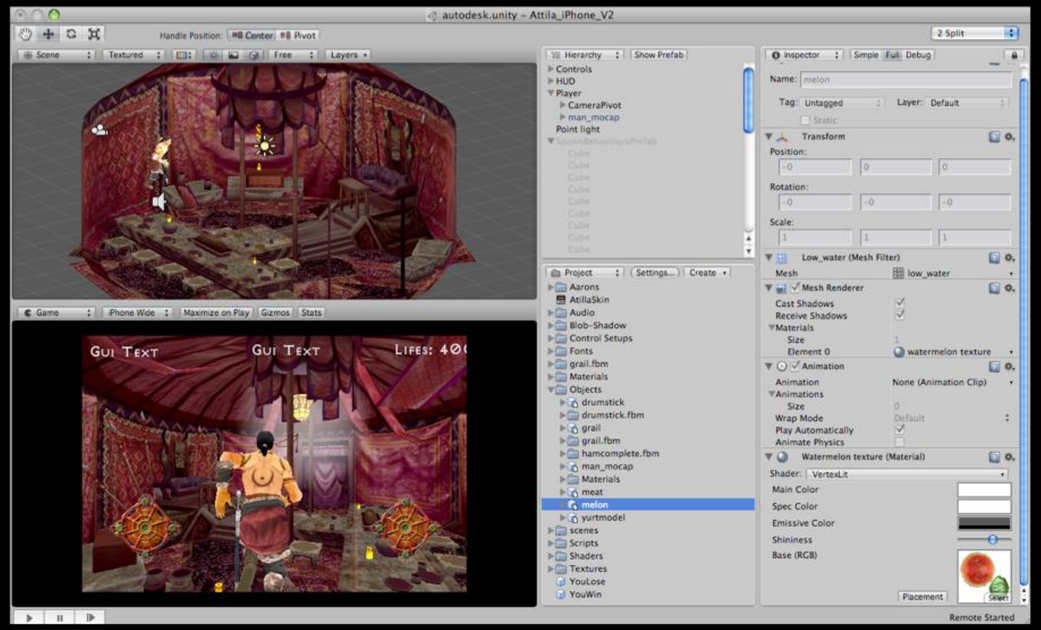


Unity & Autodesk: STEAM



Unity & Autodesk: STEAM





Unity & Autodesk: STEAM

```
var h : float= Input.GetAxis("Horizontal");
var v : float= Input.GetAxis("Vertical");

animator.SetFloat("Speed", h*h+v*v);
animator.SetFloat("Direction", h, DirectionDampTime, Time.deltaTime);
```

Hand’s on example: teaching STE(A)M

Hand’s on example: teaching STE(A)M

Q&A