

GDC



GDC

Windows Machine Learning for Gaming

Stuart Schaefer, Microsoft

Software Architect

Windows AI Platform

Daniel Kennett, Microsoft

Engineering Manager

Xbox Advanced Technology

Group

Yury Yuralsky, NVidia

GPU Architect



Agenda

1. Introducing the WinML API
2. Getting Started with ML
3. Using WinML with Unity
4. Roadmap for Game Developers
5. Call to action / Resources

Introducing Windows Machine Learning

- Applications of ML to make games more real are rapidly becoming more practical
 - Denoising, Superscaling
 - Character Animation
 - Scene Generation
 - NPC Interaction
 - ...
- Windows Machine Learning allows developers to utilize trained models for inference on AI capable silicon running with Windows
 - Game developers target a broad set of hardware with sophisticated in-game AI
- Perf Talk: Room 3022, West Hall @ 11:30am



ML Superscaling vs Bilinear Upscaling

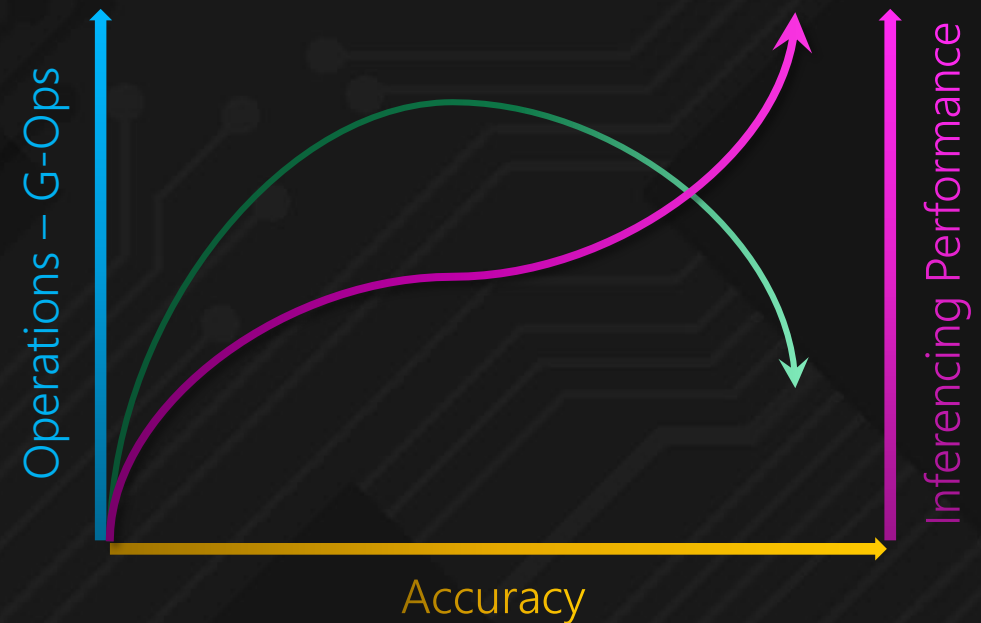
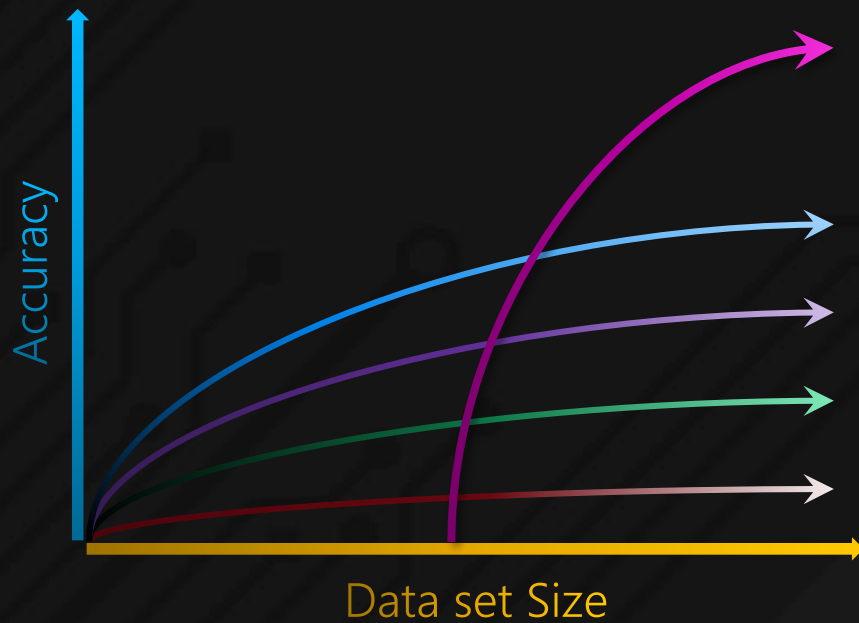
Scale Changed Everything

Data, Compute and Deep Learning

- Cognitive skills such as speech and image recognition as good as humans
- Collection and labelling of large scale data sets
- Broadly available compute power including GPUs

Ready for the Consumer

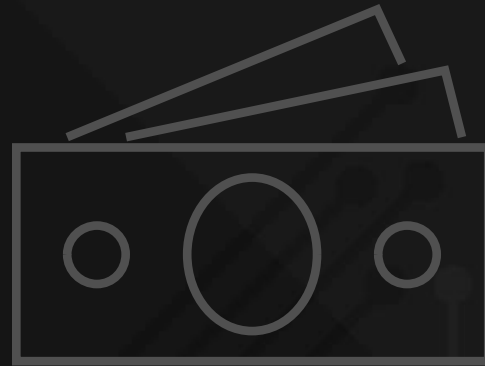
- Accessibility to tools, frameworks and data
- Reuse of across multiple domains
- Hardware advances combine with model and algorithmic improvement



AI on Windows



Low latency,
real-time results



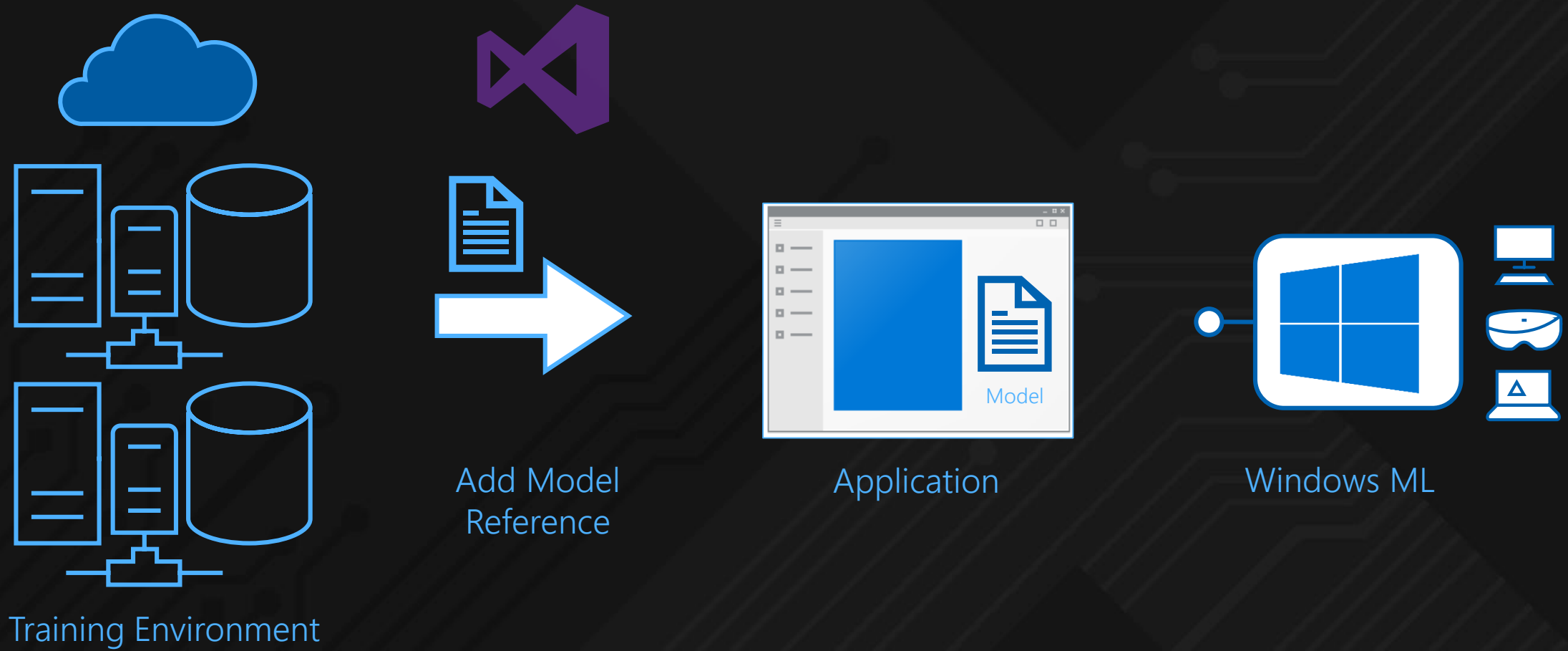
Reduced
operational costs



Flexibility

Windows ML Workflow

1. **Load** – Loads model into Windows ML runtime
2. **Bind** – Wires up inputs and outputs to model
3. **Eval** – Evaluates the model and produces results



Windows ML

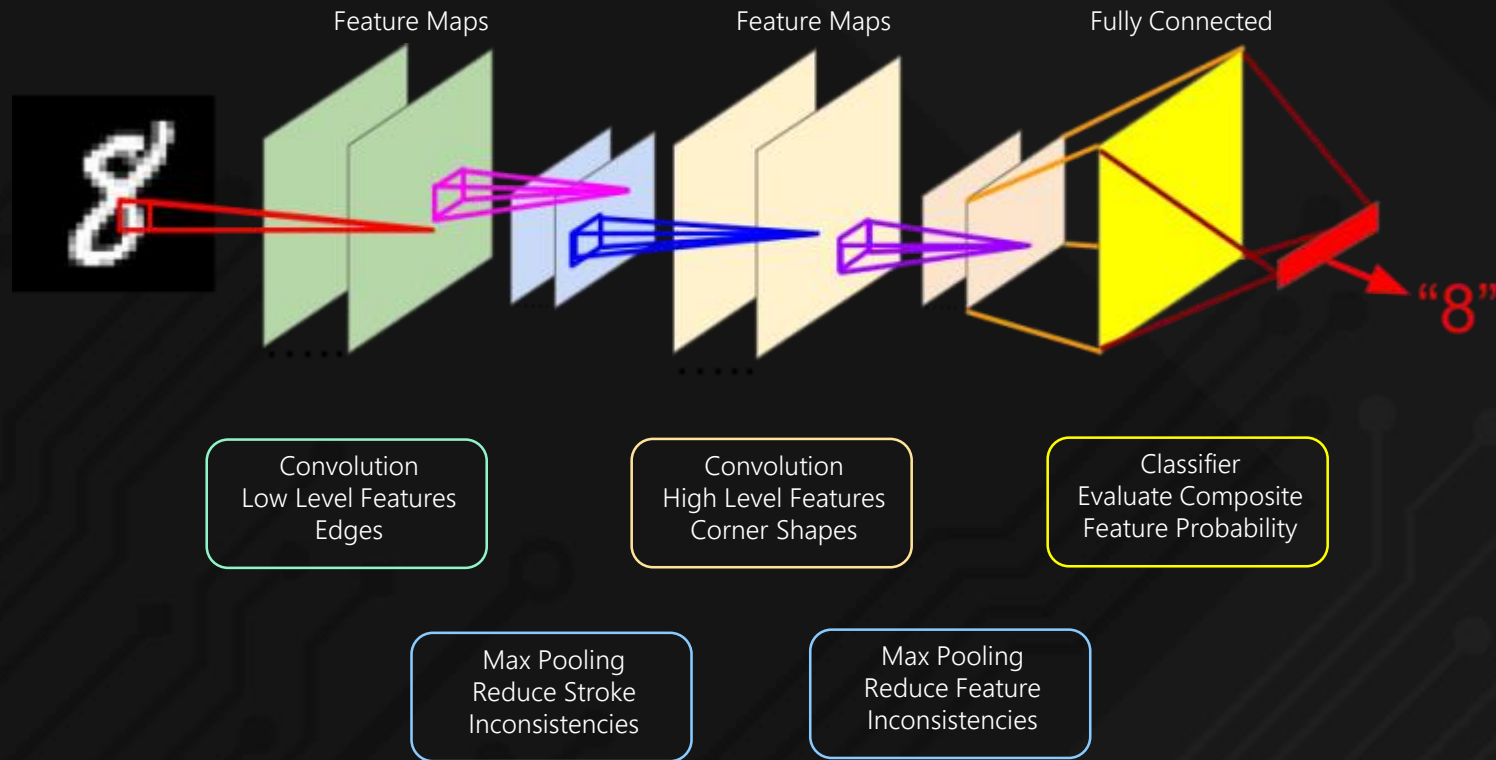
- WinML API
 - Available to Win32 & UWP applications
 - Converts images to Tensor Resources
 - Available on all Windows editions in 2018
- Inference Engine
 - Model & Device resource management
 - Loads and compiles operator kernels
 - Execute dataflow graph
- Device Layer
 - CPU instruction optimizations up to AVX-512
 - DirectML generates DX12 Compute shaders



Working with Models - ONNX

- Windows ML uses ONNX models natively
 - Microsoft teamed with Facebook and Amazon to establish the Open Neural Network Exchange (ONNX)
 - Numerous industry partners including Nvidia, AMD, Qualcomm, Intel and others
- Train in common frameworks and convert to ONNX using WinMLTools
 - Caffe2, CNTK, CoreML, SciKit, SparkML, XGBoost, LibSVM
 - TensorFlow just released to open source - <https://github.com/onnx/tensorflow-onnx>
 - Keras and others coming soon
- Create ONNX models on Azure
 - Azure Machine Learning
 - Coming soon: Azure Custom Vision Service

Image Recognition - Convolutional Network



Explore layers of spatial features using common image filters

- Convolution - trained filter kernel to extract features
- Pooling – downsample to denoise and create spatial invariance

Use “fully connected” classifier to evaluate identified features

<https://www.analyticbridge.datasciencecentral.com/profiles/blogs/overview-and-simple-trial-of-convolutional-neural-network-with>

Windows ML - Interact with Model

- Load Model

```
ComPtr<IWinMLRuntime> spRuntime = WinMLCreateRuntime();
```

```
ComPtr<IWinMLModel> spModel;
```

```
spRuntime->LoadModel(GetModelPath(), &spModel);
```

- View Graph Metadata

```
spModel->GetDescription(&pDescription);
```

```
spModel->EnumerateMetadata(count, &key, &value);
```

```
spModel->EnumerateModelInputs(count, &pDescriptor);
```

```
spModel->EnumerateModelOutputs(count, &pDescriptor);
```



The screenshot displays a 'MODEL PROPERTIES' window with two main sections: 'MODEL PROPERTIES' and 'GRAPHS'. The 'MODEL PROPERTIES' section shows 'Format: ONNX v3' and 'Producer: onnxmltools 0.1.0.0000'. The 'GRAPHS' section shows 'Name: coreml_MNIST' with a 'View' button. Below the name, it lists 'Inputs: float[0,1,28,28] : image' and 'Outputs: : prediction' and 'int64[1] : classLabel'.

MODEL PROPERTIES	
Format	ONNX v3
Producer	onnxmltools 0.1.0.0000

GRAPHS	
Name	coreml_MNIST View
Inputs	float[0,1,28,28] : image
Outputs	: prediction
	int64[1] : classLabel

Windows ML - Bind Resources

- Setup the D3D Device to use for Inferencing

```
ComPtr<IWinMLEvaluationContext> spContext;
```

```
ComPtr<ID3D12Device> spDevice; // ← Any D3D device which is Compute capable!!!!
```

```
spRuntime->CreateEvaluationContext(spDevice.Get(), &spContext);
```

- Connect input and output data

```
WINML_BINDING_DESC bindDescriptor;
```

```
bindDescriptor.BindType = WINML_BINDING_TYPE::WINML_BINDING_TENSOR;
```

```
bindDescriptor.DataType = WINML_TENSOR_DATA_TYPE::WINML_TENSOR_FLOAT;
```

```
bindDescriptor.NumDimensions = 4;
```

```
INT64 shape[4] = { 1, 3, 25, 25 };
```

```
bindDescriptor.pShape = reinterpret_cast<INT64*>(&shape);
```

```
spContext->BindValue(&bindDescriptor);
```

- Current support for Tensors as UAV or CPU resource
- Enables asynchronous binding of multiple sets of inputs and outputs

Windows ML – Evaluate and Get the Answer!!!

- Evaluate model (inference)

```
spRuntime->EvaluateModel(&spContext);
```

- Process the results – map of classes to probabilities

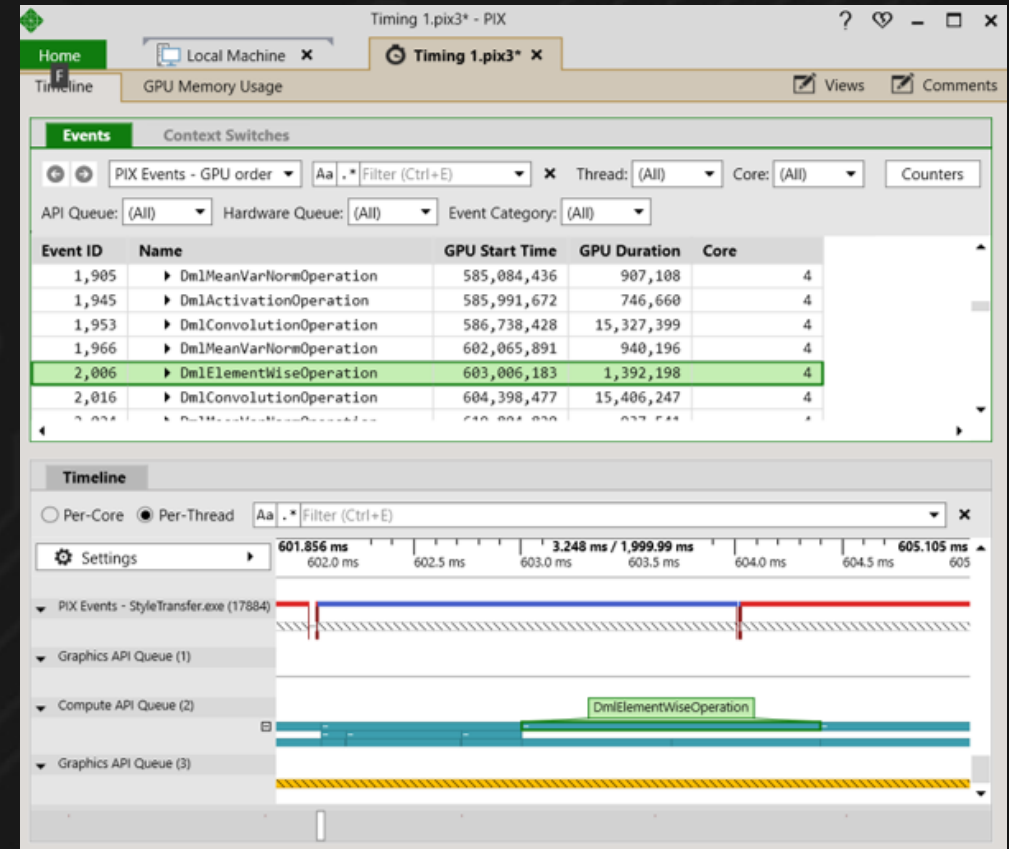
```
int maxIndex = 0;
float maxProbability = 0.0;
for (int i = 0; i < binding.shape ; i++)
{
    if (binding.prediction[i] > maxProbability)
    {
        maxIndex = i;
        maxProbability = binding.prediction[i];
    }
}
```

Demo – MNIST on Windows ML



Tools

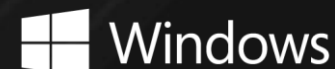
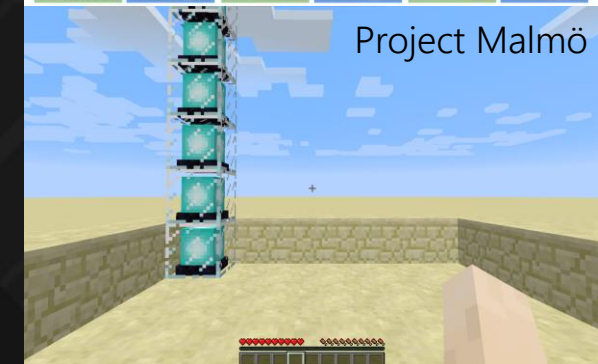
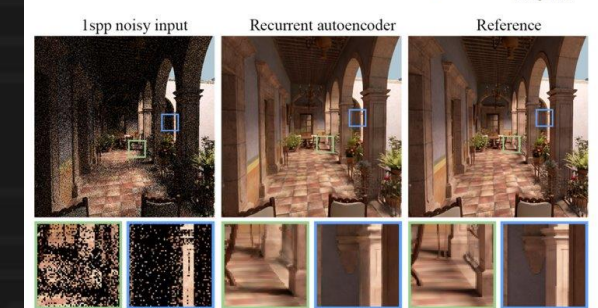
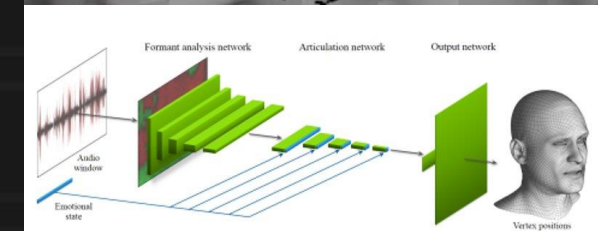
- Converters - WinMLTools
 - <https://pypi.org/project/winmltools/>
- Project integration with Visual Studio
 - MLGen – in the Windows SDK
- Visualization - Netron
 - <https://github.com/lutzroeder/Netron>
- Debugging - PIX
 - ML events in the Compute queue
 - Operator and model timing

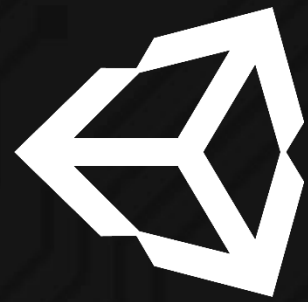


Machine Learning and Game Development

- So. Excite.
- ML can be used in so many ways in games
 - AI Bots (testing, competitive, etc)
 - Services (matchmaking, inappropriate content detection, etc)
 - Art & Content pipelines
 - Runtime visual FX, animation (post-fx, motion/locomotion, etc)
 - Telemetry/BI analysis
 - And so much more....
- Windows ML lets you iterate and share quickly, on any compute capable silicon.
- Speaking of quick iteration....

*Images on right are illustrative examples of ML only; not running Windows ML and with exception of Project Malmö, none are Microsoft projects.





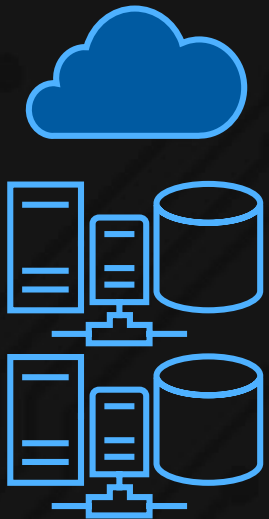
unity ML-Agents Support



Unity ML-Agents on Windows ML

- **Unity Machine Learning Agents** (ML-Agents) is an open-source Unity plugin that enables games and simulations to serve as environments for training intelligent agents
- We added Windows ML support to the list of Brains allowing Unity to execute ONNX models via Windows ML
 - Enables game developers and researchers to run their models on any AI capable HW

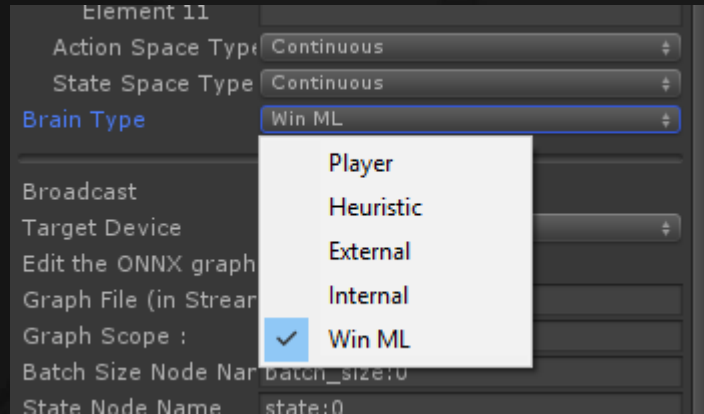
How it works....



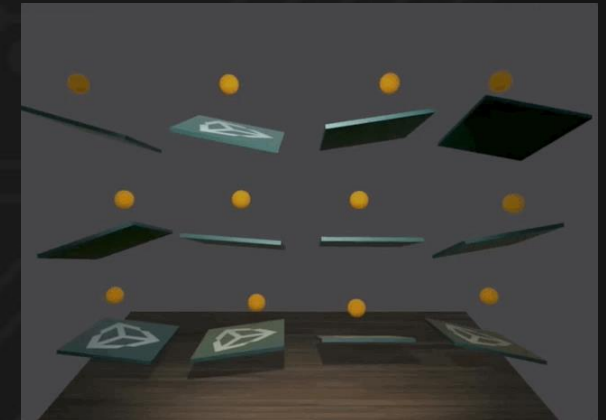
TRAIN



CONVERT



UPDATE

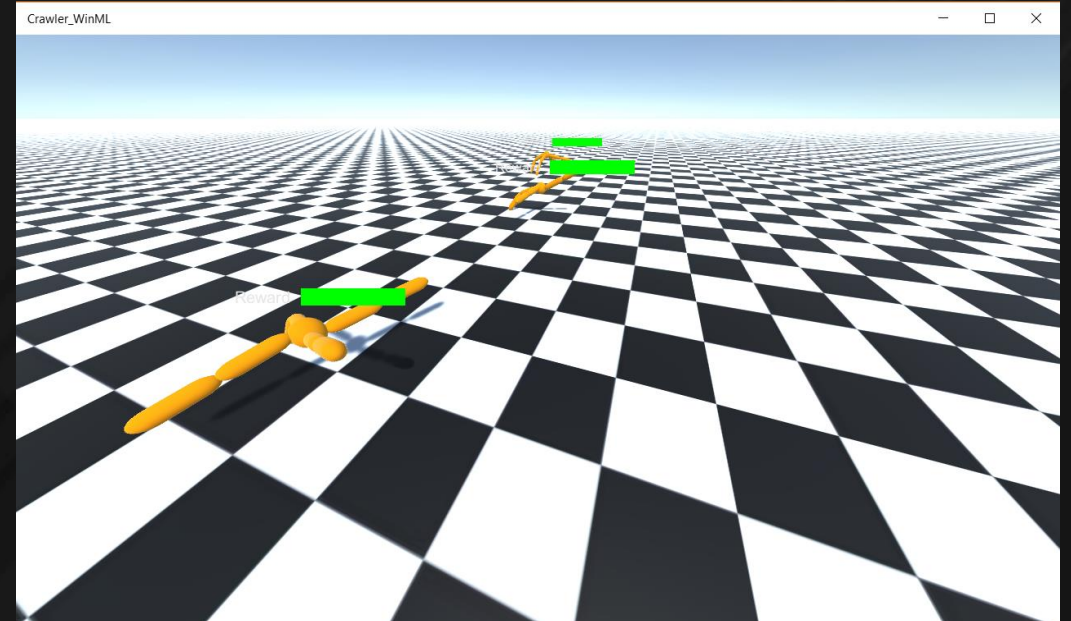


RUN

How it works, with details

- Acquired latest Windows Insider Build & the 17110 SDK
- We converted Unity's pre-trained models into ONNX via [TensorFlow->ONNX converter](#).
- Exposed Load, Bind, Eval model/calls as a Brain in Unity.
 - Used these [Windows ML docs](#)
- Selected WinML as brain type, imported converted ONNX model, then ran.

It runs! Unity ML-Agents 0.2 Release



Things to know...

- Windows ML / ML-Agents integration targeting release later this year
- Has been a great exercise in stress testing Windows ML and ONNX. ;-)
- Special Thanks to Unity!

Roadmap and Resources



Roadmap for Windows ML

- NOW: We've released a Preview of Windows ML
 - Current support for Computer Vision, Classical ML scenarios
 - Available now to developers using Insider builds
 - Broadly available in upcoming release of Windows 10
- NEXT: We know game devs want more
 - Custom operators
 - Low-precision/Quantization
 - Already added FP16 to Shader Model 6.2
 - D3D Textures, Audio, ...
 - Hardware accelerators – Come to our next talk!!!

Resources

- Check out our blog for updates: blogs.msdn.microsoft.com/directx/
 - Links for Getting Started
 - ONNX Converters
 - Tools and tips
 - Stay up to date with the latest happenings in DirectX at DirectXTech.com

Questions?



Come talk to us if you...

- Have a scenario you'd like to explore
 - Want to run a model on the client
 - Are concerned about ML performance in your game
 - Have questions about ML @ Microsoft
-
- We'll be in the ID@Xbox booth from 1:30 – 3:30 today



© 2018 Microsoft Corporation.

All rights reserved. Microsoft, Xbox, Windows, and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries. The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation.

MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.