Fun and Challenges
Stable Diffusion, ONNX, WASI-NN

Stable Diffusion v1.5

- Began with the demo - Stable Diffusion with C# | onnxruntime
  - Models at - runwayml/stable-diffusion-v1-5 at onnx (huggingface.co)
- Stable Diffusion is a pipeline of multiple models
  - Prompt Encoder
    - Tokenizer -> Text Encoder
  - UNet – called multiple times
  - Image Decoder
  - Safety Checker
- Demo at BUILD used latest platform optimizations

Model Loading

- Runtime Version / OpSets
  - graph-encoding is not version sensitive
  - Error contains only invalid-encoding
  - No API to query or specify runtime version
- Single model vs. Model plus weights
  - Implementation specific -> assume [model] or [model, weights]
- Session Options
  - Many runtimes have configuration to control optimization, threading, device preference, …
  - UNet can use DirectML specific free dimension optimizations
  - Session configuration and initializers also unsupported
- Custom Operators
  - No means to load custom operators to support the graph
    - OrtExtensions library used by the tokenizer
  - Could be provided as WASM only?

Compute

- Execution Target
    - Which GPU? My system has three – Integrated and 2 Nvidia Titan V
    - ONNX on Windows supports both CUDA and DirectML
    - Normally add an ordered list of Execution Providers as device preference
    - Forced to hardcode implementation specific behavior
- Execution Context
    - Often used to hold reference to CUDA stream/DirectML device, not just gpuid
    - Maintain buffer references and synchronization between stages and external processing
- Run Options
    - Session configuration overrides

Tensors

- Named Tensors
- Shared Buffers / GPU Buffers
- Model Chaining
  - In the pipeline it is preferred to directly connect the outputs of one model to the inputs of the next with no memory copy
  - get-output provides a tensor-data where set-input takes a tensor
  - Adds latency to create tensor on each run