IronPython vs Unity ML-Agents toolkit:

**IronPython -**

IronPython is an [open-source](http://www.opensource.org/licenses/apache2.0.php) implementation of the Python programming language which is tightly integrated with the .NET Framework. IronPython can use the .NET Framework and Python libraries, and other .NET languages can use Python code just as easily.

**Unity ML-Agents toolkit –**

**The Unity Machine Learning Agents Toolkit** (ML-Agents) is an open-source Unity plugin that enables games and simulations to serve as environments for training intelligent agents. Agents can be trained using reinforcement learning, imitation learning, neuroevolution, or other machine learning methods through a simple-to-use Python API. We also provide implementations (based on TensorFlow) of state-of-the-art algorithms to enable game developers and hobbyists to easily train intelligent agents for 2D, 3D and VR/AR games. These trained agents can be used for multiple purposes, including controlling NPC behaviour (in a variety of settings such as multi-agent and adversarial), automated testing of game builds and evaluating different game design decisions pre-release. The ML-Agents toolkit is mutually beneficial for both game developers and AI researchers as it provides a central platform where advances in AI can be evaluated on Unity’s rich environments and then made accessible to the wider research and game developer communities.

**Advantages of Unity ML-Agents toolkit over IronPython:**

IronPython is used to integrate python into the.NET framework, which is what we intended to use in our project for controlling the AI elements. But on further research, the Unity ML-Agents toolkit is much more useful and resourceful to our project than IronPython is.

Advantages of ML-Agents vs IronPython:

* ML-Agents is an officially unity plugin, whereas IronPython is an independent open source add-on to the .NET framework.
* ML-Agents was built for the purpose of serving as an environment for training intelligence agents, which would benefit us greatly for the project we are working on. Whereas if we were to use the IronPython plugin, we would have to design these environments from scratch.
* ML-Agents is currently in development and is being improved constantly, whereas IronPython is a finished product and doesn’t have as much more potential beyond being a python plugin for Unity.

**Conclusion:**

We initially set out to use IronPython in our project but after some research we found that there is a new integrated toolkit which is focused on training AI in an intelligence training environment with the use of python.