Augmented Reality Training Environment

SME Solutions Inc.

Olivia Folsom, Alex Giacoio, Gunisha Kamra, Jordan Neff, Lieu Phung, Jack Sharrar

Client Requirements

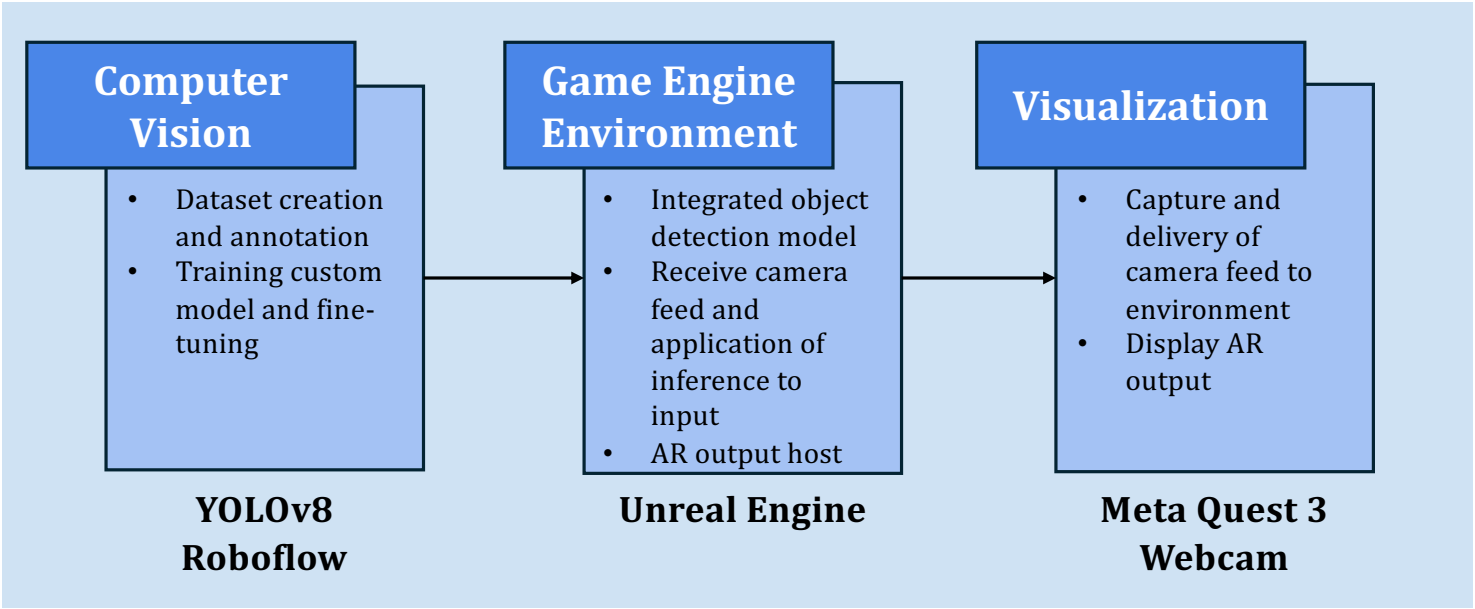
- [x] 1. The AR object detection environment shall identify at least 5 objects dynamically in real time.
- [x] 2. The AR training application shall be able to import images or wire mesh models of objects.
- [x] **3. The AR object detection environment shall identify objects in various orientations and environments.**
- [x] 4.The AR training application shall generate a data set to be used for model training.
- [x] 5. The AR object detection environment may utilize game engines such as Unity and Unreal for graphic overlays.
- [x] 6. The AR object detection environment shall be designed, developed and tested using Agile methodologies.
- [x] 7. The team shall use GitHub or Gitlab to encourage configuration management and collaboration.
- [x] 8. The AR host platform shall have a camera and be able to interface with real world (smartphone app, meta quest, or emulator).
- [x] **9. The AR object detection environment shall be designed using modular and open system methodologies.**
- [x] 10. The AR object detection environment may use, but not limited to, industry standards such as C++, C#, and python coding languages.
- [x] **11. The AR object detection environment shall outline, label and dynamically follow detected objects.**
- [x] 12. The AR object detection environment may utilize computer vision frameworks such as OpenCV, TensorFlow, or PyTorch.
- [x] 13. The AR object detection environment shall be designed to maintain a real-time AR experience thus the model must support at least 30 fps.
- [x] 14. The AR object detection environment shall include a logging functionality.
- [x] 15. The AR object detection environment shall include functionality to enable/disable specific objects from being detected.

Project Overview

Task: Develop an open and modular Augmented Reality object detection environment that identifies objects and provides object information in real time

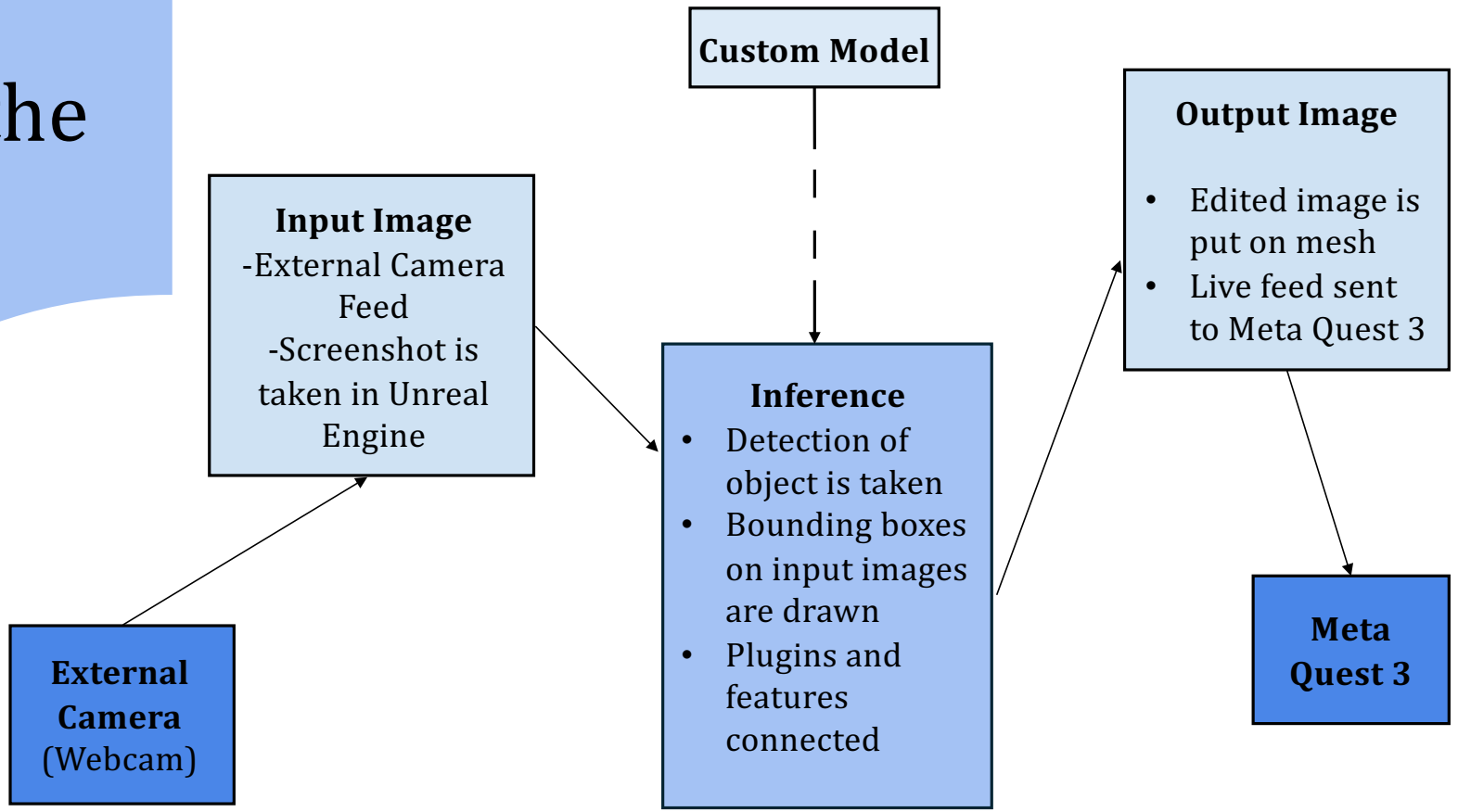
Purpose: Create an industry standard machine learning application, which is adaptable and customizable

Goal: Deliver a functional product that meets client needs and requirements



Research → Development → Final Integration + Documentation

Flow of Data through the System



Final Deliverables

Documentation

- Custom Training
- Initialization
- Unreal Engine C++ Project Structure
- Image Sequence Playback System in Unreal Engine
- Camera Input in Unreal Engine
- Training Application Proposal

Setup Guides

- Total Setup (YOLO & Unreal Engine)
- Roboflow Dataset
- Camera Input In Unreal Engine

Full System on GitHub

End Product

- Real time object detection at 30 fps
- AR environment with display of detected objects
- Modular system that allows for plugin, hardware or algorithmic changes
- Robust system of support via documentation and set-up guides.