Portfolio

About Me:
I am a Software Engineer at Intel. Prior to this, I was an Augmented Reality Engineer at Aireal contributing to the development and advancement of Geospatial AR platform.

I graduated from The University of Texas at Dallas with a Master's degree in Computer Science in December 2016. I was a Graduate Researcher at Future Immersive Virtual Environments (FIVE) Lab under Dr. Ryan McMahan where I developed Virtual Reality applications and related user studies. I presented one of my accepted papers in 2016 IEEE 3DUI Symposium. This document is a collection of my projects.

I am passionate about Augmented, Virtual & Mixed Reality, Internet of Things, Computer Vision, Robotics, Creative Coding and above all amalgamation of these technologies to design aesthetic application.

Before joining grad school, I used to work on Jaguar Land Rover's infotainment modules in TATA Consultancy Services. While I am not working I love being a chef or meet people in various meetups in the City or improve my tricking skills.

Swaroop Kumar Pal
swarooppal.wordpress.com
github.com/dv-dt-spal

Train-AR-Corgi Stanford Continuing Studies

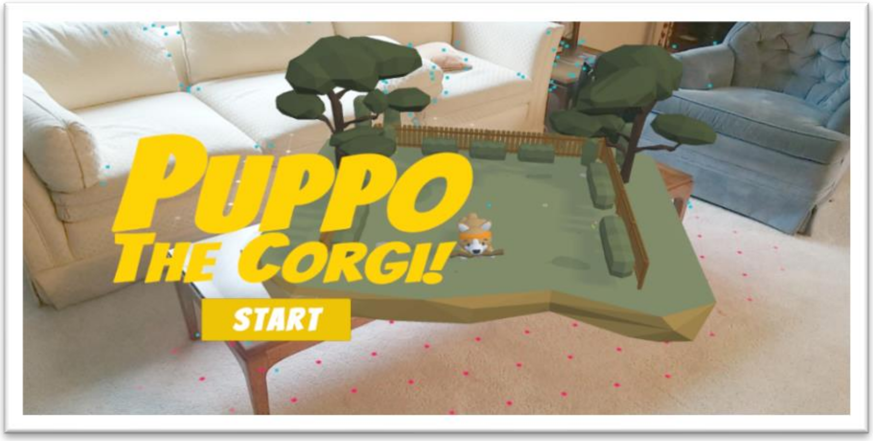


Fig – Train AR Corgi

This project aims to explore reinforcement learning as a strategy to guide an ml-agent in a mixed reality application. Mixed reality applications allow virtual content to interact with an environment as detected by sensors, such as a smartphone camera. This poses the challenge of finding algorithms that can cope with a variety of uncontrollable environments. Machine learning provides solution to adapt and overcome this challenge. In this project reinforcement learning is used to provide the framework for defining the behavior of a Non-Player-Character. The NPC can be trained by providing it with rewards when it achieves the desired goal and completes a task correctly. This is showcased by a mixed reality mobile game for Android where a user can play fetch with a dog in a virtual scene placed on a real-world environment.

Technical Details

- The application is delivered as an Android app.
- The app is created in Unity game engine with the ML Agents plugin as a central framework.
- TensorFlow & Python are used as a machine learning API for training the ML toolkit brain.
- Google's AR Core for environment scanning and interpretation.
- Combines an ARCore and ML Agents into one entity.

Technology & Tools

- ARCore, Machine Learning, Unity, C#, Tensorflow, Unity ML Agents Toolkit

Silver Unicorn - Reality Virtually Hack MIT Media Lab

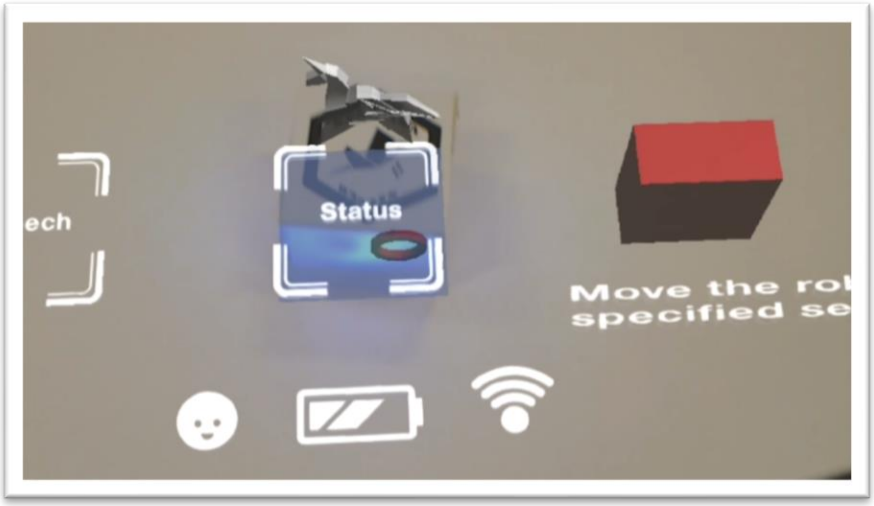


Fig – Silver Unicorn UI

Silver Unicorn was developed at Reality Virtually hackathon at MIT media lab in October 2016. This project was awarded the 1st prize in Getting things done and doing business (engineering) category. Silver Unicorn is a personal companion robot with mixed reality, tangible and voice interface. The main objective of the project was to showcase human-digital-physical-natural interface. I was responsible for developing the interactivity modules and networking components on Hololens and building and deploying the application.

Technical Details

- Application was built in Unity and deployed on to Hololens using Visual Studio.
- Application uses the Universal Windows Platform (UWP) networking packages.
- Application uses the inbuilt gaze, gesture and voice control on Hololens.
- Hololens interactivity scripts programmed in C#.
- A desktop application containing a TCP server to receive voice commands from Hololens and a Bluetooth classic client using Sphero SDK to send motor commands to the robot.
- The app makes use of Hololens coordinate system & spatial mapping to place 3D spatial UI.
- The robot consists of a marker recognized by Vuforia running on Hololens.
- The marker reveals a virtual “Silver Unicorn” which has predefined interaction with the spatial menu.
- The speech menu enables the voice interaction on Hololens which sends out data to the TCP server on keyword recognition.

Technology & Tools

- Microsoft Hololens, Mixed Reality, Unity, C#, TCP/IP, Vuforia, Sphero robot

IoTxMR - Smart Home & Mixed Reality Fusion on Hololens

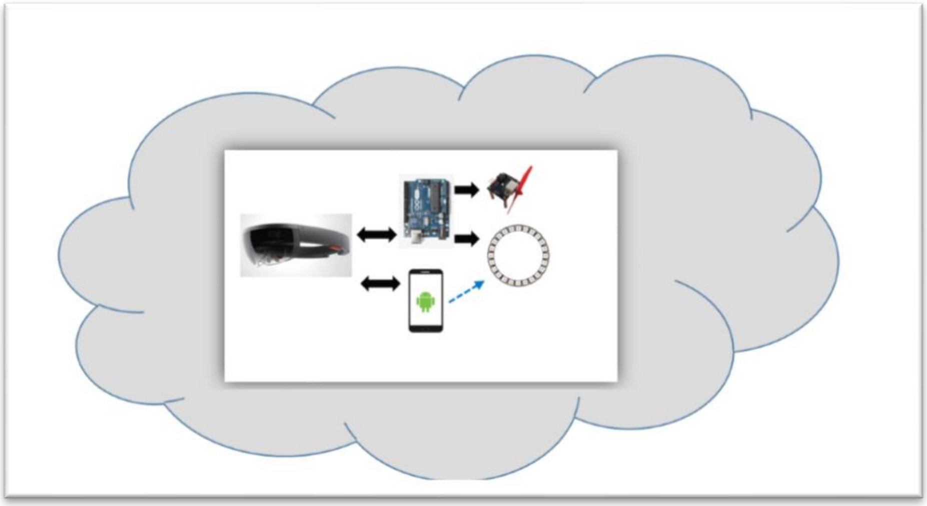


Fig – IoTxMR Architecture

IoTxMR is a proof of concept application developed at Holohacks in San Francisco conducted by Microsoft Hololens Team. The primary goal of the app is to provide a 3D spatial UI for cross-platform devices (Android Music Player app and Arduino controlled Fan and Light) and to interact with them using gaze and gesture control. The app has a unique use case which portrays devices not having a direct correlation in real-world can have one in the Mixed Reality. The application also incorporates a Zen mode where the user can have an engaging mixed reality experience, interacting with virtual objects fused in a real world environment. This was featured in Digital Trends.

Technical Details

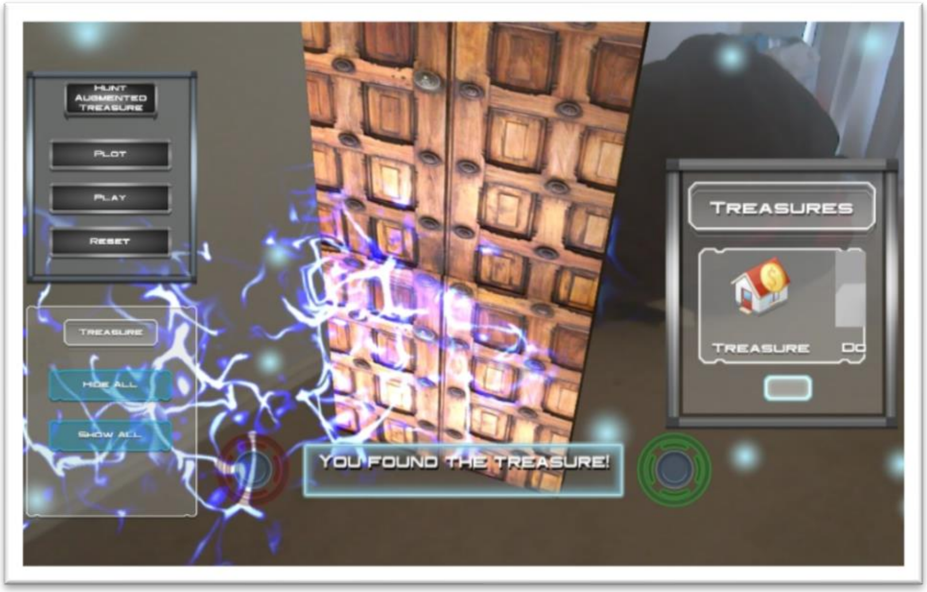
- Application was built in Unity and deployed on to Hololens using Visual Studio.
- Application uses the Universal Windows Platform (UWP) networking packages.
- Application uses the inbuilt gaze and gesture control on Hololens.
- Hololens interactivity scripts programmed in C#.
- Android Music Player app and Arduino controlled fan & light implement a TCP server in Java and Arduino sketch respectively along with desired functionalities.
- All devices were connected to a single access point.
- The app makes use of Hololens coordinate system to place 3D spatial UI and the Zen portal.
- On any UI interaction, the app sends an appropriate TCP request to the concerned server.
- The Zen mode's state is toggled based on the user's position relative to the portal.

Technology & Tools

- Microsoft Hololens, Mixed Reality, Unity, C#, TCP/IP, Arduino, Android.

Note: A Magic Leap version of the project was recently developed at ATT SF Hackathon which also included Social Mixed reality features using AWS.

Google Project Tango - Hunt Augmented Treasure



YOU FOUND THE TREASURE!

Fig – Project Tango Hunt Augmented Treasure GUI

This is an Augmented Reality Game on Project Tango using Unity 3D game engine. Project Tango is a platform that uses computer vision to give devices the ability to understand their position relative to the world around them. It has three main features (1) Motion Tracking (2) Area learning (3) Depth Perception. The device augments virtual 3D objects onto the current camera captured frame and tracks the location of the virtual object using Tango's motion tracking APIs. The application also provide a user interface to select/show/hide objects. The project involves concepts of augmenting 3D objects on to 2D camera screen enabling augmented reality and rendering the object based on the device pose.

The game allows one of the players to place the "treasures" (virtual objects) in various places in an indoor location. These treasures could be hidden from the view after placing them. The game shows clues and a trail for solving the treasure hunt. A different player plays the game searching for clues, solving them and locating the virtual objects. The game gets over when the player finds the treasure or quits the application.

Contributions

- Ideated the game story and design.
- Self learnt Project Tango Development process and familiarized with motion tracking APIs.
- Designed the GUI and interactivity module for the project. Implemented the scripts in C#.

Technology & Tools

- Augmented Reality, Unity 3D, C#, Project Tango, Android, Motion Tracking.

Head Based Rendering Virtual Reality User Study

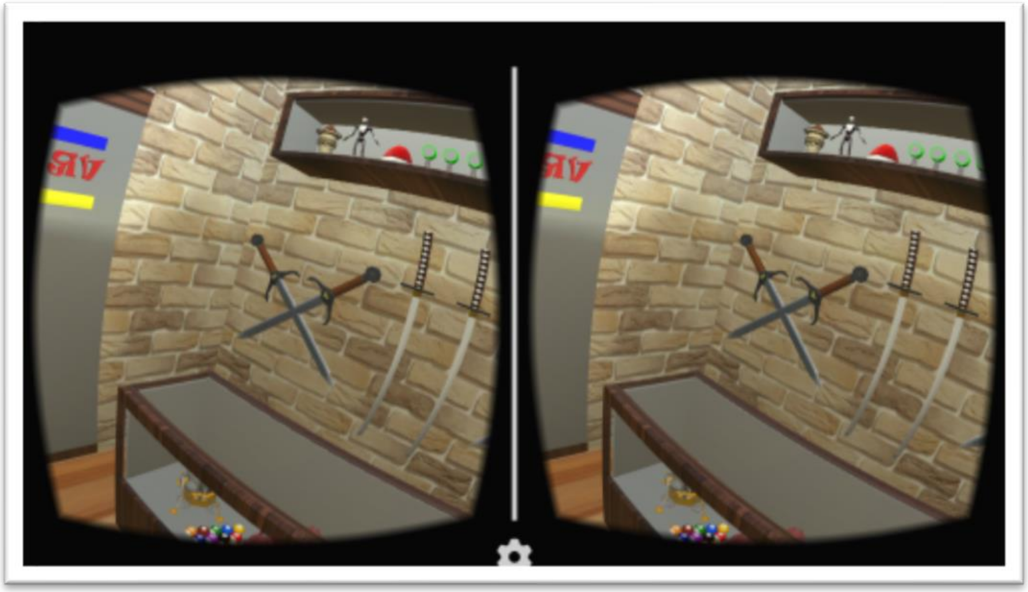


Fig – Pawn Shop Virtual Environment in Unity 3D

This project was developed for a Virtual Reality user study in Future Immersive Virtual Environment (FIVE) Lab at UT Dallas. This purpose of this study was to investigate and compare the effects of complete head tracking (6 DOF), rotational head tracking (3 DOF in HPR), and translational head tracking (3 DOF in XYZ).

The project consisted of a pawn shop as the virtual environment. The pawn shop had objects scattered all around in the virtual space. The user task was to select different set of objects using 3 different head tracking. The application provided visual and audio cues for selecting an object.

The user study was conducted in a Motion Capture Vestibule with pre-installed and calibrated Vicon system. Oculus Rift DK1 was used as the Head Mounted Display, Nintendo Wii remotes were used for interaction in the virtual environment.

Contributions

- Modelled pawn shop in Maya.
- Designed the Virtual environment in Unity and added features for audio visual cues.
- Designed the interactivity module for the project and implemented the scripts in C#.
- Imparted training to under graduate students to conduct user study.
- Supervised the user study and performed data analysis.

Technology & Tools

- Virtual Reality, Unity 3D, C#, Maya, Oculus Rift DK1, Nintendo Wii remotes.

Positional Tracking in Samsung Gear VR

Fig – FIVE Lab Virtual Environment in Android with actual dimension

This project was developed as an independent study in Future Immersive Virtual Environment (FIVE) Lab at UT Dallas. The project was an attempt to enable positional tracking on Samsung Gear VR, which uses an Android device for processing. The Head Mounted Display has an inbuilt rotational tracking. Incorporating positional tracking on it, enables it as a mobile Virtual/Augmented Reality device.

The project had a virtual environment which had exactly the same dimension as the FIVE Lab. The environment and its interactive components were developed in Unity. The output was exported into an Android app. The Android app was developed as a framework for Unity and OpenCV. Optical Flow concepts in Computer Vision were used to determine the positional tracking every frame. The Android device's rear camera was used in a background capture mode. The output from the vision processing mapped the actual movement of the user in the virtual space.

Contributions

- Modelled virtual world which had exact same dimension as FIVE lab in Maya.
- Designed the interactivity module for the project and implemented the scripts in C#.
- Integrated Unity and OpenCV on Android. OpenCV layer was implemented using NDK.
- Enabled the background camera capture to capture frames to be analyzed by Optical Flow algorithms in OpenCV.

Technology & Tools

- Virtual Reality, Augmented Reality, Unity 3D, C#, Gear VR, Android, OpenCV, NDK.

Interactive Projection Mapping using Kinect

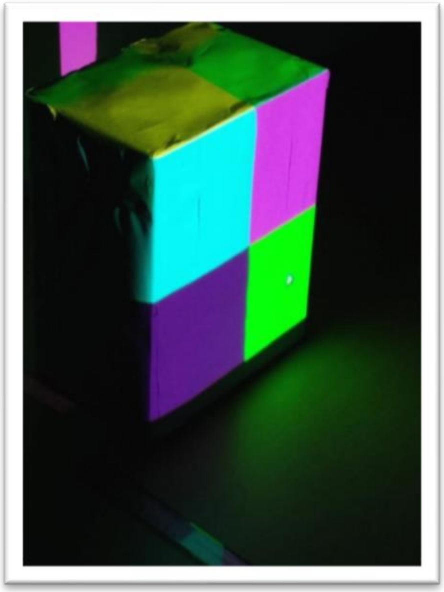


Fig – Neutral Box projected with texture generated in Unity 3D

This project was also developed as an independent study in Future Immersive Virtual Environment (FIVE) Lab at UT Dallas. The primary goal of this project was to project dynamically generated textures onto a neutral box and change the texture using predefined gestures.

The environment which consisted of the neutral box and a projector, were replicated in Unity as a 3D box (with dimension same as the neutral box) and a virtual camera respectively. It was assumed that the box was being held by the user in one hand. The user and the box were tracked using Microsoft Kinect. The gestures were added based on user's hand position.

Since the entire setup was matched to Unity's setup. Whenever the user moved, the virtual box moved in Unity taking input from Kinect. The virtual camera's output was fed into the projector's input, which was then projected onto the box. The gestures were also recorded using Kinect and the events were added to Unity C# scripts to trigger a texture change.

Contributions

- Designed the interactivity module for the project and implemented the gesture scripts in C#.
- Integrated Kinect Wrapper Package in Unity3D.
- Prototyped the entire setup – Environment setup, Projector Calibration.

Technology & Tools

- Projection Mapping, Unity 3D, C#, Microsoft Kinect, Creative Coding, Arts and Technology.

Mutual Exclusion and Broadcast Service in a Distributed System

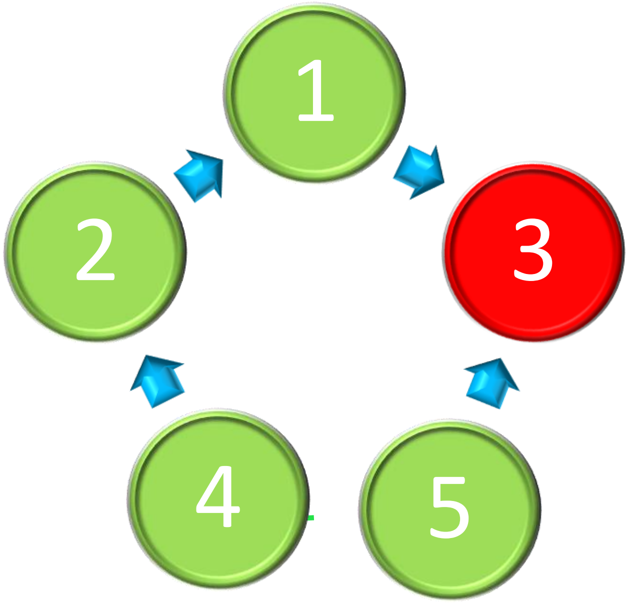


Fig – Raymond Mutual Exclusion Tree. Node 3 contains the token.

The project consists of a distributed system in which nodes are arranged in a certain topology. A spanning tree is built using a distributed algorithm. Once the spanning tree construction completes, each node knows which subset of its neighbors are also its tree neighbors. The spanning tree is used to implement a broadcast service that allows any node to send a message to all nodes in the system. The broadcast service informs the source node of the completion of the broadcast operation.

Using the above broadcast system Raymond's Mutual Exclusion algorithm for a distributed system is implemented. The algorithm uses the spanning tree of the computer network, and the number of messages exchanged per critical section depends on the topology of this tree. In this project each node generates critical section requests periodically, which are executed based on the availability of token at the node.

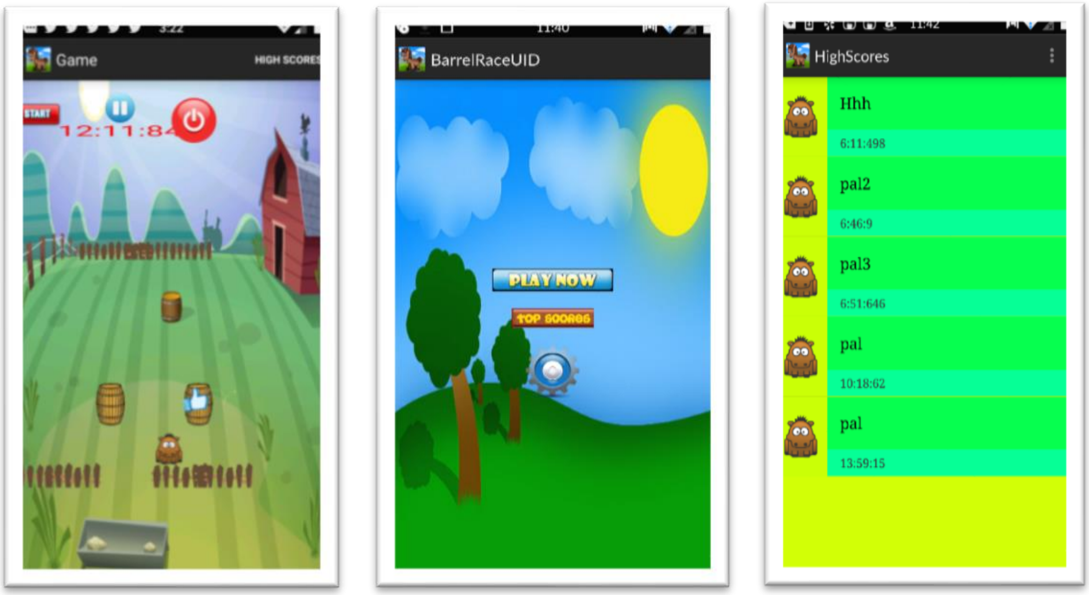
Contributions

- Designed and implemented the entire application in C++.
- Developed a utility System Layer in C++ consisting of POSIX Semaphore, Threads and Mutex.

Technology & Tools

- Computer Networks, Distributed Computing, C++, Sockets, TCP/IP, Multithreading.

Barrel Race Android



Hhh
6:11:498
pal2
6:46:9
pal3
6:51:646
pal
10:18:62
pal
13:59:15

Fig – Barrel Race Android Screenshot

The Barrel Race is a rodeo event in which the rider starts at a gate and must ride completely around three barrels. That is, the player must circle each of the barrels. The objective is to get the fastest time without knocking over any of the barrels. In the game the "horse" starts out at the gate, and maneuvers around the three barrels, as shown in the screenshot, and back through the gate.

This game uses the accelerometer to compute the position of the horse. The game uses Android "Drawable" features. The project was developed in android studio.

Contributions

- Developed, designed and implemented the game in Android studio.
- Implemented custom adapter for scores list view.
- Gave special importance to app aesthetics.

Technology & Tools

- Android, Accelerometer, Android Studio.

Smart Intelligent Home

Fig – Smart Home Setup

This project was completed in 2011 in senior year of bachelor's degree in Electrical and Electronics engineering. The project's objective was to build a smart home prototype model with internet control and gesture control for electrical appliances and motors attached to the doors. The prototype also had various sensor interfacing.

An Atmega 648 was used for developing the embedded modules for actuating the relays, which in turn actuated the electrical appliances. An Arduino UNO with Ethernet shield was used to host a web server to provide the internet control. Also, OpenCV running on a desktop was used for gesture control to actuate the motors controlling the door of the prototype.

Contributions

- Prepared timelines, cost estimation and assigned roles to the 5 members of the Team.
- Implemented a Smart Home model with electrical appliances and doors being controlled through face recognition using OpenCV library.
- Designed the CAD model using Google Sketch up.
- Programmed Arduino, Atmega 640 microcontroller for various sensor (LDR, temperature, proximity), relay actuation and motor control.
- Effected System Integration of the Model.

Technology & Tools

- Smart Home, Micro-controller programming, Atmega, Arduino, Sensor interfacing.