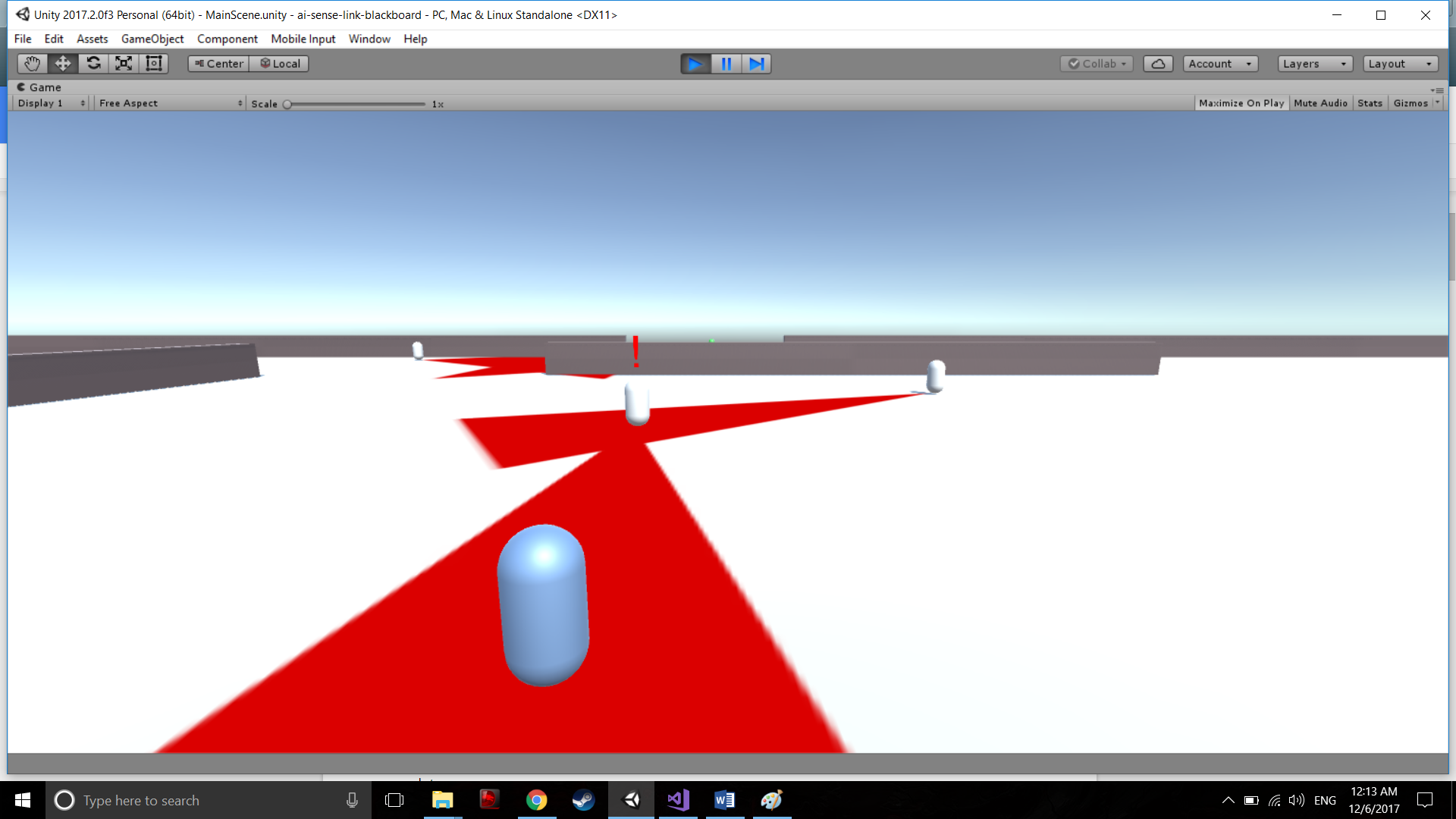
Artificial Intelligence

**Blackboard & Sensory System for AI**

NITISH VICTOR, ANUSHKA NAIR

horizontal line

# 



# 

# 

# Introduction

An AI system where “sense links” are used and senses such as sight is  
traced using view cones and raycast and a simple hearing system. The idea is to pair this  
with the Blackboard system as the basis for the AI architecture for each individual.  
A basic sense link system will be implemented which will store relation to all other  
similar individuals in the space. It will also store sensory data such as raycast and hearing. This data will act as the blackboard system to help the decision experts decide the next  
action to perform, for example: if the individual entity hears a sound within the radius, a  
decision expert will look at the data and send actions to perform based on the available  
data.  
Each Individual entity will have:

* Visual system: A Raycast and a View Cone
* Hearing System: Sound recognition within a certain radius
* Path Finding System: Based on available data, entity will move towards the  
  Position
* Attack System: Deals damage to a hostile entity if within range
* Personal Behavior System: Actions based on the personality of the individual

# 

# 

# 

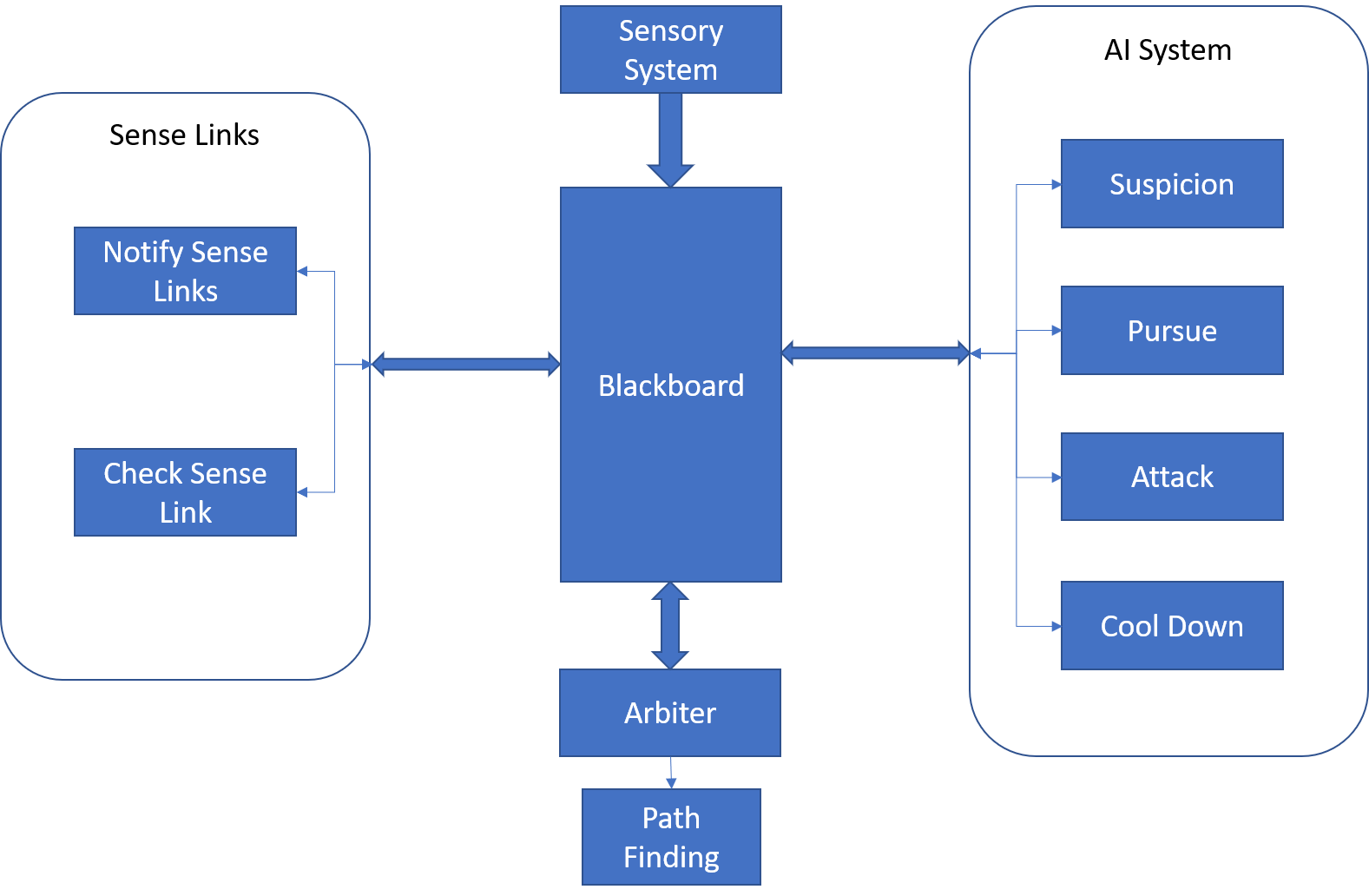
# 

# 

# 

# 

# Technical Implementation



*Figure 1.* Architecture

## Architecture

The system consists of a blackboard which is accessed by the AI system, Sense Links, Sensory System and the Arbiter (Check *Figure 1*). All subsystems have read/write access to the blackboard.

### Blackboard

The blackboard is implemented as a dictionary with a generic value type to allow storage of varied data types within one dictionary. This also allows us to easily add or remove data from the blackboard when required. Since the blackboard is accessed from multiple threads, all the threads spawned from the main thread accesses the blackboard only after placing a lock on it.

### Sensory System

The sensory system consists of a view cone and a simulated hearing system. The view cone registers if the player has entered within the enemy’s field of vision and updates the blackboard accordingly. Similarly, the hearing system also updates the blackboard with relevant data.

### AI System

The AI subsystem runs on a separate thread. The system in essence updates the blackboard with state changes and logic outputs based on the available data. Suspicion value is ramped up based on the time the player is within the view cone, and once a threshold is crossed, the pursue system takes over. If line of sight to the player is lost, where the enemy will search for the player until the cool down timer reaches zero. If the enemy is within attack range, the state of the enemy will be set to *Attack*.

### Sense Link

Every enemy is has a shared sense link object with each other. This sense link allows the enemies to share rudimentary information of the current state. Hence, when one enemy is in pursuit of the player or is attacking the player, the sense link notifies enemies within range of the location of the player.

### Arbiter

The Arbiter is the place where the actual state transition takes place. The logic to decide which state to transition to is defined in the arbiter. The decision is taken based off the data present in the blackboard. As all the subsystems continuously update the blackboard, the arbiter takes the current state of blackboard and performs any logic based off of it.

# 

# Challenges Faced

* Since the AI subsystem and Sense Link subsystem run on separate threads, the data in the blackboard was getting corrupted. This was causing frequent crashes early on in the development.
* Even after placing locks on the blackboard within the threads, the system is tedious to debug as working with multiple threads sometimes causes the IDE to hang or crash. As a result, some code paths are not completely tested.
* Even with the blackboard system, the overall system still hinges on the implementation of the Arbiter. Since the Arbiter is using an FSM, it is subjected to the same pitfalls. Hence, adding any more states to the Arbiter becomes a tedious process.
* *Any other challenges?*

# Individual Research

* By Nitish:
  + The Blackboard System
  + AI Sense Links
  + Sensory system → View cones
* By Anushka
  + The Blackboard System
  + Sensory System → Hearing
  + Personalities of the Enemies

# Post Mortem of the Project

* The proposed project was implemented successfully but there is a need for a slightly better laid out architecture in the Arbiter to make it more extensible.
* The FSM restricts the extensibility provided by the blackboard system.
* The multithreading makes the flow of information from the blackboard to the sense links slightly difficult, but was the most efficient way of executing the system without crashing unity.
* The types of enemies are identified as Archers and Brawlers, where the Archers have narrower view cones and are passive in nature, whereas the Brawlers have wider view cones and are aggressive in nature, thus making them faster pursuers.
* The overall AI can be improved by giving the enemies memory which would help remember the last chased position of the player and patrol in the newly established location rather than traveling back to the starting position.

# Future Work

* For stealth games and any game which requires slightly complex NPC behavior, this sensory system can be used to implement the behavior and also expand on it in an extensible manner.
* It can also be altered to use in RTS games where individual agent behavior might be important.
* The different enemy behaviors can be useful for creating a more challenging gameplay in terms of enemy AI.
* The simple behaviors can be replaced with a behavior tree to make adding and defining multiple enemy behaviors easier and more manageable.
* Dialogues can be implemented which will let the player know when and where the enemy AI is located and what are the future actions.