A logo with green text

Description automatically generated

**CS4096 - AI for Games**

**PROJECT REPORT**

**AI Child Companion Game**

**Word Count (1968)**

**Video Link:** [**video**](https://ulcampus-my.sharepoint.com/:f:/g/personal/24101613_studentmail_ul_ie/Ev4nzAKbLeFCnHkA-y-gcLsB9frLVzIkMx0KdzD7RoWlGQ?e=2ormha)

**Project Source Code:** [**Source Code**](https://ulcampus-my.sharepoint.com/:f:/g/personal/24101613_studentmail_ul_ie/Em4mslGa_GJNgKsK7HzZdhMB1JVCyHU41Ow9SMhkcVhA9A?e=5zx2vs)

**Executable Build:** [**Build**](https://ulcampus-my.sharepoint.com/:f:/g/personal/24101613_studentmail_ul_ie/EgcZTIvUkk5Eid-JKYskcasBBfWuyFkoevszMVvz6VyxOw?e=ahJdff)

**Submitted by -Aryank Gupta**

**Student ID - 24101613**

**Index**

|  |  |
| --- | --- |
| **Section** | **Page Number** |
| **Title Page**   * Student Details * Word Count | **1** |
| **Introduction**   * Overview of the Game Concept * Problem Statement: Child Safety through AI | **3** |
| **Significance**   * Role of AI in the project * Tools and Technologies used | **4** |
| **Analysis**   * Game Design and AI Techniques * Stranger Controller: Behaviour and Implementation * Working and Implementation * Environment Setup and Navigation Design * Integration and Asset Utilization | **5-6** |
| **Reflection**   * AI Behaviour and Results Analysis * Challenges and Limitations * Individual Contribution and Team Integration | **7** |
| **Results and Observations** | **8** |
| **Future Scope and Improvements** | **8** |
| **Conclusion** | **8** |

**Introduction**

The “AI Child Companion Game” is the safety- concerned interactional game that apply artificial intelligence so that the typical real life related child safety scenarios can be played. It is a typical child and/ or pet playing a game within an environment. The ‘guardian,’ the dog, gets up from its sleep to bark, either at people who might be coming into the compound or any other figure of threat. In this project, my main contribution was to model the Stranger Controller, and to have the level construction done in Unity in conjunction with other team members who worked on the child and the dog controllers.

This project relates close to the issue involving critical safety of children with reference to ‘stranger danger’. Specifically, the use of artificial intelligence in the design of protective game themes has improved within the gaming business. Paw Patrol Adventures and other games combined with games aimed at education and safety or AI simulation games like Sims or Animal Crossing, show how engagement and positive reinforcement can be applied to the use of AI. Our game, however, changes the roles and goals and makes an individual expose and defend, as well as identify risks and threats.

**Overview of Game Concept**

The *AI Child Companion Game* is designed to address child safety concerns through an interactive AI-driven experience. The game places a child and their protective dog companion in a simulated environment where potential threats from strangers are mitigated through the dog’s vigilant behavior. As strangers approach, the dog reacts by barking, deterring the stranger and alerting the child. This gameplay mechanic serves both as an educational tool and a demonstration of AI-based decision-making in safety-critical scenarios.

The core gameplay revolves around three primary AI behaviours:

1. **Child AI** - The child must navigate the environment and react appropriately when a threat is detected, including following the dog when needed.
2. **Dog AI** - The dog acts as a protector by detecting when a stranger is too close to the child and alerting them through barking.
3. **Stranger AI** - The strangers patrol the park and react to the presence of the child and the dog.

**Problem Statement: Child Safety through AI**

In a world where child safety is paramount, leveraging AI to simulate protective behavior provides an engaging way to educate and raise awareness. The game integrates AI techniques to create an environment where the child is safeguarded against potential dangers, emphasizing the importance of proactive and reactive protective mechanisms.

**Significance of Stranger character**

The *Stranger Controller* is a critical element in the *AI Child Companion Game*, introducing the concept of threat and unpredictability. Its role is to simulate realistic stranger behavior, alternating between random movement and targeted approach toward the child. This behavior not only challenges the dog’s protective AI but also adds dynamic gameplay tension. Professionally, the Stranger Controller exemplifies the use of AI to simulate antagonistic behavior in a structured and reactive system, showcasing Unity’s capabilities for creating complex AI interactions.

**Role of AI in the project:**

AI is the backbone of the *AI Child Companion Game*, enabling dynamic interactions and decision-making among the child, dog, and strangers. The Stranger's AI utilizes several techniques to drive its behavior. It employs random movement generation within a defined radius, shifting its position at regular intervals using coroutines to determine new target positions. As the Stranger approaches the child, its behavior shifts based on proximity, transitioning from random movement to approaching the child, which is controlled dynamically using flags and coroutines.

The Stranger's behavior is scripted in real-time using C# within Unity, allowing it to adjust its actions based on interactions with the child. Additionally, collision detection is used to trigger responses from the Dog Controller when the Stranger enters or exits the child's proximity, utilizing Unity's trigger-based collision system (OnTriggerEnter and OnTriggerExit).

**Tools and Technologies used**

To achieve the intended gameplay, the game employs:

* **Unity**: The primary platform for developing and running the code, specifically using Unity's MonoBehaviour class for script functionality, like movement, collision detection, and object manipulation.
* **C#**: The programming language used for the script.
* **Coroutines**: Unity's feature for running time-dependent functions in parallel without blocking the main game thread. The code uses IEnumerator for coroutines like RandomMovementRoutine, MoveToPosition, and ApproachChildRoutine.
* **Vector3**: A Unity class for handling 3D positions and directions in space. It is used extensively in the code for random movement, approaching the child, and calculating distances.
* **Triggers and Colliders**: Unity's physics system, where the OnTriggerEnter and OnTriggerExit methods are used to detect when the stranger enters or exits a trigger zone, such as when close to the "Child" object.
* **Debugging**: The code uses Debug.Log for logging status updates during execution, useful for debugging purposes within Unity's console.
* **Asset used**: Kevin Iglesias, Human Base Meshes.

**Analysis**

**Game Design and AI Techniques**

The core gameplay revolves around three AI-driven entities: the child, the dog, and the stranger. FSMs manage their behavior, ensuring modularity and scalability. The child follows predefined paths, the dog responds dynamically to threats, and the stranger alternates between random patrolling and targeted approach.

**Example Implementation: FSM**

FSMs govern state transitions, such as the stranger moving from *patrolling* to *approaching* states upon detecting the child. This modular design simplifies debugging and facilitates adding complexity, such as avoidance behavior when detected by the dog.

**Stranger Controller :**

The Stranger Controller script defines the Stranger’s behaviour.

* Patrolling Behaviour: The Stranger moves randomly within a designated radius. Random movement ensures dynamic gameplay, making encounters unpredictable. This was implemented using a coroutine to periodically generate random target positions and move towards them using normalized vectors.
* Approach Behaviour: On detecting the child within a defined range, the Stranger shifts behaviour to approach the child. This action demonstrates a gradual escalation of threat, enhancing the dog's role in safeguarding.
* Proximity Detection: Trigger events detect when the Stranger enters or exits the Child’s proximity, notifying the Dog Controller to initiate protective actions.

**Working**

The project simulates the behavior of a Stranger character within a Unity-based environment. The Stranger moves randomly within a defined radius and transitions its movement when it detects proximity to a child object. Initially, the Stranger moves to random positions within the defined radius using coroutines to periodically shift its target position. When the Stranger gets close enough to the child, it stops the random movement and begins to approach the child by calculating a target position in front of the child, determined by a stop distance. After reaching this position, the Stranger returns to its random movement behaviours. Collision detection is integrated to detect when the Stranger enters or exits the child's proximity using Unity's trigger system. Upon detection, a Dog Controller is notified to respond accordingly. This setup provides dynamic interaction between the Stranger and the child, simulating real-time behavioural shifts in response to proximity and environmental changes.

**Implementation:**

1. **Initialization**: The Initialize method is used to set up the "Stranger" by associating it with a child object and a radius for random movement. This method starts two coroutines: one for random movement and one for approaching the child.
2. **Random Movement**: The RandomMovementRoutine coroutine handles the random movement of the stranger within a specified radius around its initial position.

Every few seconds, the stranger moves to a new random position within the defined area. It uses MoveToPosition to gradually move the stranger to the new position.

1. **Approaching the Child**: The ApproachChildRoutine coroutine handles the logic for the stranger approaching the child. After a random delay, the stranger starts moving towards a point in front of the child (calculated in MoveToPositionInFrontOfChild). Once the stranger reaches this position, it returns to random movement.

The approaching behaviours is controlled by the approachingChild flag to ensure the stranger doesn’t try to approach the child while moving randomly.

1. **Moving to a Specific Position**: MoveToPosition is a coroutine that smoothly moves the stranger towards a target position using vector calculations, based on a given speed. This ensures that the stranger moves continuously until it reaches the target.
2. **Moving to a Position in Front of the Child**: The MoveToPositionInFrontOfChild method calculates a position in front of the child based on a stopDistance and moves the stranger towards it.
3. **Collision Detection**: The OnTriggerEnter and OnTriggerExit methods are triggered when the stranger enters or exits a collider, specifically the proximity of the child. If the stranger enters the area around the child, a log message is generated, and if a DogController instance exists, it notifies that a stranger is nearby. Similarly, when the stranger exits the area, it notifies the Dog Controller that the stranger has left.

**Environment Setup and Navigation Design**

The environment is designed to balance gameplay complexity and navigability. Key elements include:

* **Terrain Setup:** A terrain with obstacles such as trees and benches challenges navigation while maintaining realism.
* **Obstacle Placement:** Strategic positioning ensures meaningful interactions between the child, dog, and strangers.

**Integration and Asset Utilization**

Unity Asset Store provided placeholders for the child, dog, and stranger, ensuring development efficiency. Integration focused on:

* **Asset Placement:** Ensured the logical alignment within the environment.
* **Behavioural Synchronization:** Aligned animations with AI state changes for enhanced realism.

**Reflection**

**AI Behaviour and Results Analysis**

The implemented AI successfully demonstrates:

* **Dynamic Interactions:** Strangers exhibit varied patrol and approach patterns, creating engaging gameplay.
* **Reactive Protection:** The dog’s response to threats highlights effective state management and integration between controllers.

**Challenges and Limitations**

* **Navigation Issues:** Strangers occasionally fail to navigate complex terrain effectively.
* **Performance Concerns:** High stranger counts increase CPU load, particularly on lower-end systems.
* **State Transition Realism:** Abrupt transitions between behaviors sometimes detract from immersion.

**Individual Contribution and Team Integration**

As the developer of the Stranger Controller and environment, my contributions included:

* Designing and scripting stranger behaviours.
* Configured Navmash for seamless navigation.
* Integrating assets into the Unity environment. Collaboration with team members ensured smooth integration of all components, highlighting the importance of clear communication and version control.

**Results and Observations**

The Stranger Controller successfully simulates:

1. Random Movement: Strangers traverse the environment dynamically, enhancing unpredictability.
2. Threat Escalation: Strangers approach the child only when in range, triggering appropriate responses from the Dog Controller.
3. Smooth Integration: Proximity detection integrates seamlessly with the Dog Controller, showcasing real-time interaction between game entities.

**Future Improvements**

Future enhancements for the *AI Child Companion Game* include implementing Behavior Trees for more nuanced decision-making, allowing strangers to react dynamically based on the child’s state or dog’s actions. Incorporating reinforcement learning could enable the dog to adapt its protection strategies over time, enhancing realism and engagement. Additionally, introducing dynamic environments with interactive elements like gates or hiding spots would diversify gameplay and increase replicability, creating a more immersive and strategic experience.

**Conclusion**

The AI Child Companion Game showcases the application of AI in creating an engaging and educational environment focused on child safety. My contributions, including designing the Stranger Controller with FSM, configuring the navigable environment, and integrating assets, were integral to the game’s development. The project emphasized teamwork, creative problem-solving, and refining AI scripting and Unity development skills. Moving forward, future enhancements could incorporate advanced AI techniques, dynamic gameplay elements, and performance optimization, ensuring the game remains engaging, accessible, and scalable for a wider audience.