A NOVEL APPROACH ON THIRD PERSON OPEN WORLD GAME USING BLUEPRINTS

Hariharan S K[1], Aravinda Krishnan S[2]
[1]UG Scholar, [2]Assistant Professor,
Department of Computer Science & Engineering,
SRM Institute of Science & Technology (Deemed to be University),
Vadapalani, Chennai, Tamilnadu, India.

ABSTRACT: The proposed project will be a computer game created using unreal engine, where the player will fight against AI bots to cross a level and save the hostages. The AI bots are trained with GP algorithm, where a set of actions and moves are scripted to the bots so that it is able to perform an action for a predefined task. The game will be based on a 3D environment which will be created using 3ds max and the characters and textures will be created using Autodesk Maya and photoshop cc respectively. The upper hand in the project is that advanced programming knowledge is not necessary. The main advantage is the use of unreal engine for creating blueprints instead of coding in programming languages. Unreal engine also has an advantage of letting the user create new functions and define them inside the engine. This project improvises the environment into a 3D environment to give the user an interactive perspective.

Keywords- AI (Artificial Intelligence), GP (Genetic Programming), scripted, 3ds max, Autodesk Maya, unreal engine, blueprints. .

1. NAVIGATION CONTROLS USING A* ALGORITHM

The **Navigation control** of the NON PLAYER CONTROLLER’s (non player controller), the AI bots, is done using the Blueprints and in-built module of the game engine called “Unreal”. It uses the AI Algorithm called the A*, which is best suited for guessing the best path for traversing, when compared with the other path traversing algorithms. In this algorithm navigation path is set on the game world to mention the AI bots where they have access to move and where they cannot.

The path is searched by the agent using the A* algorithm in the navigation path, this is the modern path finding systems which at least contains four parts.

Navigation Map: Usually complied at development time. Describes all the traversable areas and their connectivity. It is cooked while compiling.

AI Path Queries: Performs idle animation while standing in the Nav mesh and when the player comes inside the boundary assigned to the AI it started to move inside the navigation path

AI Path Steering: Updates the AI’s position over time after it moves to a particular point and starts to play the idle animation again.

Obstacle Avoidance: As other dynamic objects (such as other agents) will not represented in the Navigation Graph, a separate system must exist that moderates the queried path to perform on-the-fly adjustments to the AI’s velocity to avoid nearby dynamic objects.

By using these four parts in the process of finding the path, the AI traverse takes place when the character comes near and also avoids the obstacles in the Navigation Path.

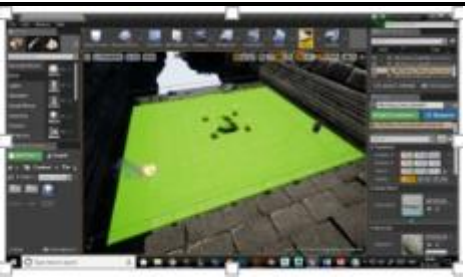

Figure 1.1 Navigation map – Paths where AI can move.

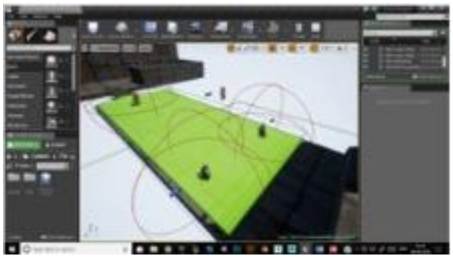

Figure 1.2 Navigation map – Boundaries of AI.

1. CHARACTER DESIGN

For the character design we did our research on how the character must be used and what the character must do (the player and Non player controller). This player controller is a third person character controlled by the one who plays the game and it is the main character of the game and will have abilities like run, jump and punch which makes the player feel the game realistic. While idle state it performs animation like searching around and this character can also see the environment and the game world through the camera attached to it in the viewport.

There are two other characters in the game which are AI bots which will not be controlled by the player, these bots will have Artificial intelligence's adaptive learning with blueprints given to them using genetic programming. These bots have been given instruction by blueprints to do the activities like interacting and chatting with the third person character, moving around the environment, triggering, attacking, etc., the bot intelligence is tested in a way that it traversing with humanly behavior to the given in the environment. As the difficulty level of the bot increases the hand-mind coordination and concentration of the player increases.

The game play is completely depends on the player who plays the game. The Third person character can communicate with the objects and AI bots in the game like and is able to give us audio, video, effects, etc.

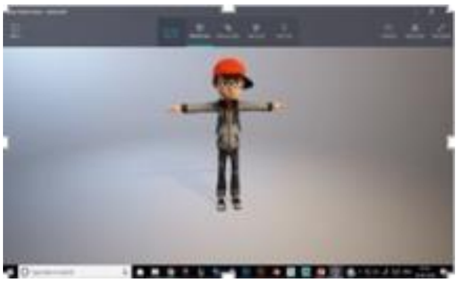

Figure 2.1 Bob (Main character)

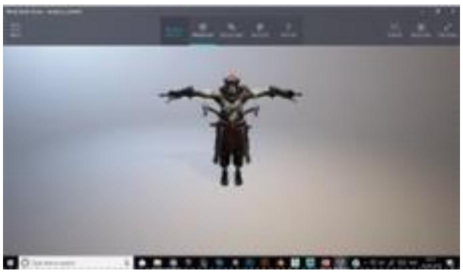

Figure 2.2 AI enemy bot

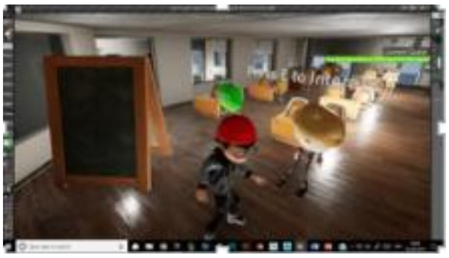

Figure 2.3 AI friendly bots

2. ENVIRONMENT DESIGN

To design the environment first we need to take reference that how the environment looks like, what are the objects to be added and the paths to travel in the game, we need from a prototype to the environment then we develop the original environment. The environment is the place where the player , the friendly bot and the enemy bot traverse from one place to other or stay static in a place doing their animation. The environment is separated into two places where one place is a school environment and other place is a fighting dungen. These two places are connected in the game through a portal where the player enters from one door to go to the other place and vice versa. The environment has static and also movable objects which can be used as lifts to move up or front.

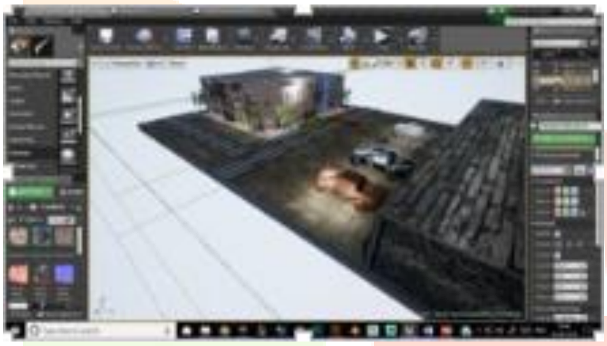

Figure 3.1 The school environment

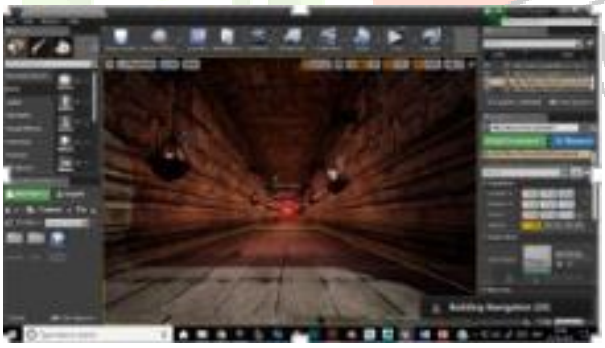

Figure 3.2 The Dungen

3.1 NAV MAP BAKE SYSTEM

The Nav Map Bake system is the process where it can be used for rendering real time lighting effects and to bake a mesh of any solid objects. Instead of changing lighting and intensities , it’s better to set lighting and bake it so that it stays constant in the entire scene when the light source is deleted and also to give the path for AI bots to roam. This Nav Bake system plays a important role in providing NON PLAYER CONTROLLER’s not to collide with solid objects.

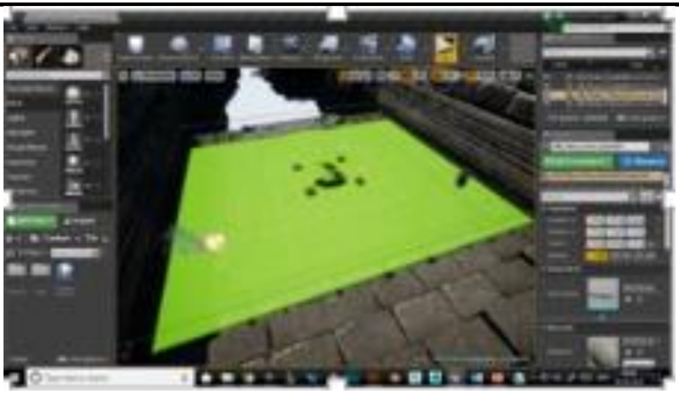

Figure 3.1.1 Nav Mesh After Baking

4. PARTICLE AND AUDIO SYSTEM

Particle systems are those which are used to emit radiant effects in certain places. Such as light sparkles, fire, water waves, wind flowing, etc. These particle systems can be added in the games to enhance the feel and performance of the game and make it more realistic for the player while playing the game.

Figure 4.1 Fire particles

With these particles if we add audio based on their type it will bring them alive in the game. Like adding audio to a water wave in sea can make it more alive like a sea water we see n hear in real life. So Audio systems are used often in the game to bring life to the game. Adding thrilling sounds based on the level gives the player to enjoy the game and get involved in it.

5. THE GAME ARCHITECTURE

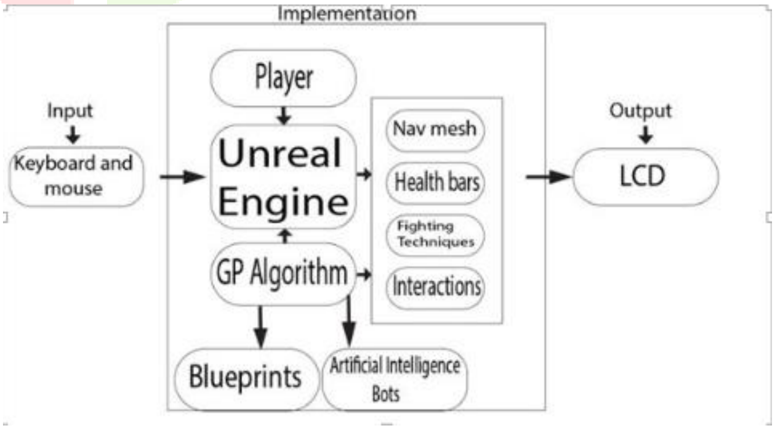

Figure 5.1 The Game Architecture

5.1 AI METHODS

Artificial Intelligence, is the main concept implemented to the bots to actually works like humans. By implementing AI techniques it adapts the actions and controls we implemented to it using genetic programming in the method of blueprints. Here the AI enemy bots when they see a human coming inside the boundary given , senses it and and runs toward him to perform the action given to it. Even when an enemy is behind , some bots will sense and some bots will not sense the character based on the technique it is implemented, and the final enemy boss is also able to perform fighting animation like punches.

5.1.1 CHARACTER AND ENEMY BOT STATES

Attack : They perform an attack like punching.

Idle : They perform the animation like searching or taunting while in idle state.

Movements : From slow movements to fast movements like walk or run.

5.2 SKELTON MESH AND ANIMATOR

The skeleton mesh is attached to the characters so that the character is able to move as a human. The rigging part is done so as to move all joints and bones separately. The animator is the tool which is used to make animations for the characters, with the skeleton mesh we create for it.

6. CONCLUSION

Thus the proposed system is more advanced than the existing system in different aspects. If some more advanced techniques and functions are used , then this can be developed into an advanced open world action game, where the player will have lots of quests to complete or if he wants he can just roam freely. The main advantage is that coding method can be easy , as we use blueprints inside the game engine to a faster access.

7. REFERENCES

1. Giovanna Martínez Arellan, Richard Cant, David Woods, "Creating AI Characters for Fighting Games Using Genetic Programming", published in IEEE Transactions on Computational Intelligence and AI in games published on Dec 4, 2017.

2.Shyong K. Lam, John Riedl, "Expressing My Inner Gnome: Appearance and Behavior in Virtual Worlds", published by IEEE Computer Society, on July 2011.

3. Chao Peng, "Introductory Game Development Course: A Mix of Programming and Art", published in Computational Science and Computational Intelligence (CSCI), 2015 International Conference on Dec 2015.

4. Timothy Davison, Jörg Denzinger, "The huddle: Combining AI techniques to coordinate a player's game characters", published in Computational Intelligence and Games (CIG), 2012 IEEE Conference on Dec 2012.