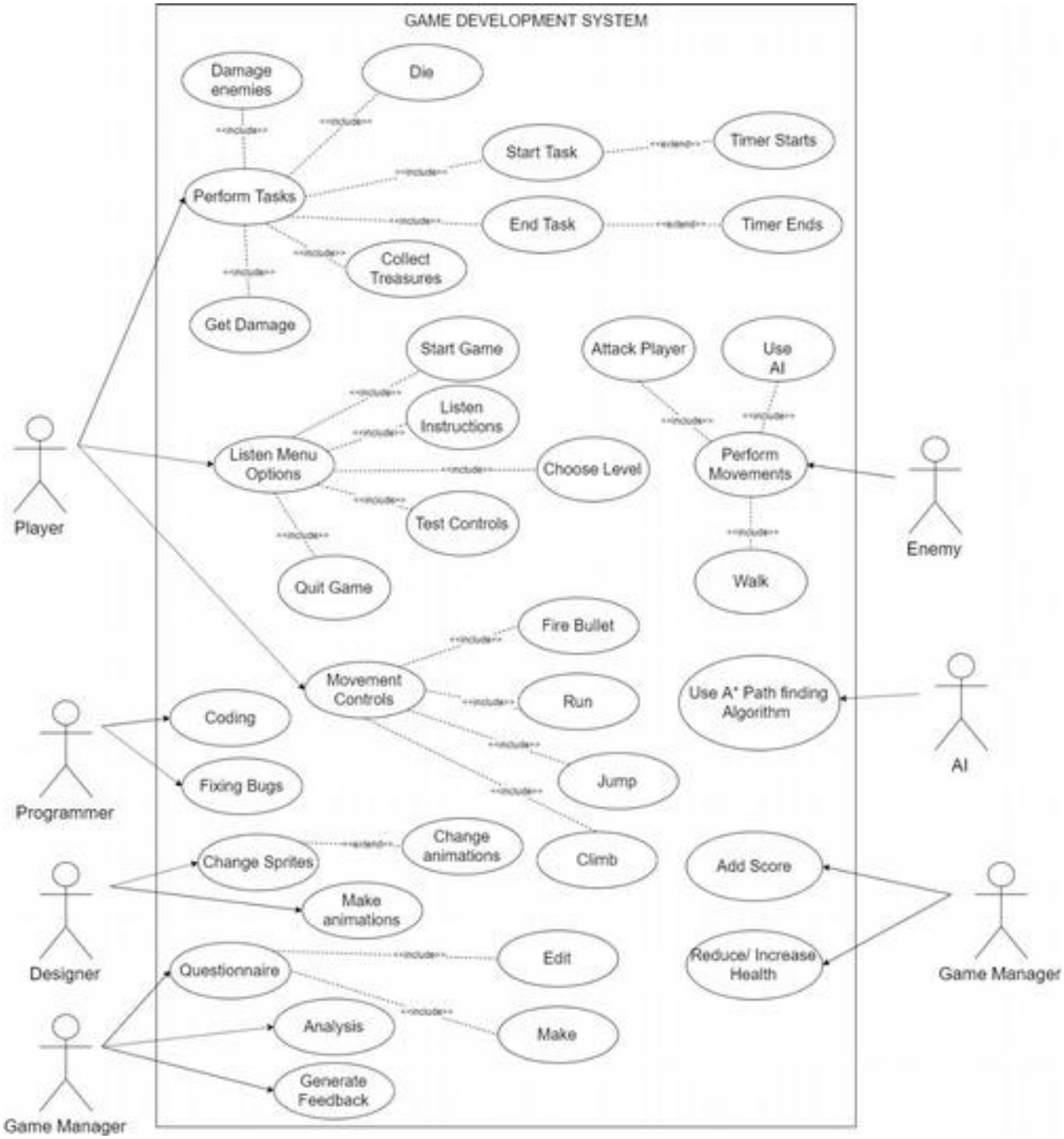
**Proposed System :**

This proposed system provides an insight into the entire process of developing a 2d indie game using unity framework. It would include the process of creating 2d pixel art assets , as well as normal assets using adobe photoshop, adobe illustrator and blender ; outputting sprite sheets which are used in the animations of the game characters and objects. Also we are designing various tiles for the game and these graphics is implemented in background and foreground. We are setting the behavior of player in such a way that he can free to move everywhere but not able cross any obstacle. It would also include a brief on shortest path-finding A* AI algorithm which would be implemented as the enemy chasing the player character to attack via the shortest path in real time. Layers of player and enemies are setting different here so that when enemy collides with player ,it should get detected. Unity physics components & concepts such as colliders, raycasting , etc would also be implemented. Unity is actually a real development platform for 2D,3D as well as AR and VR visualizations. The basic logic of running, walking, jumping and climbing of the player character would be implemented first. Then comes the game assets, puzzles and enemies. The character will be given a health and die system. Also, a score script will be written which catches the level to which the character has advanced, also being able to remember the checkpoints as it goes along. Game will be having different levels, with each level with an increasing complexity. Lots of levels which contains collecting coins , enemies attack and time constraints to required to succeed the respective level are implementing here. After the game completion , to test the quality of the game , a brief questionnaire would be prepared to get the feedback from all age groups on the basis of which statistical analysis of the game will be done which would result in the rating of the game's feel, theme and environment and would provide an area for the improvement of the same. Also, quality evaluation would be carried out to ensure a superlative, placid graphics design and gameplay.

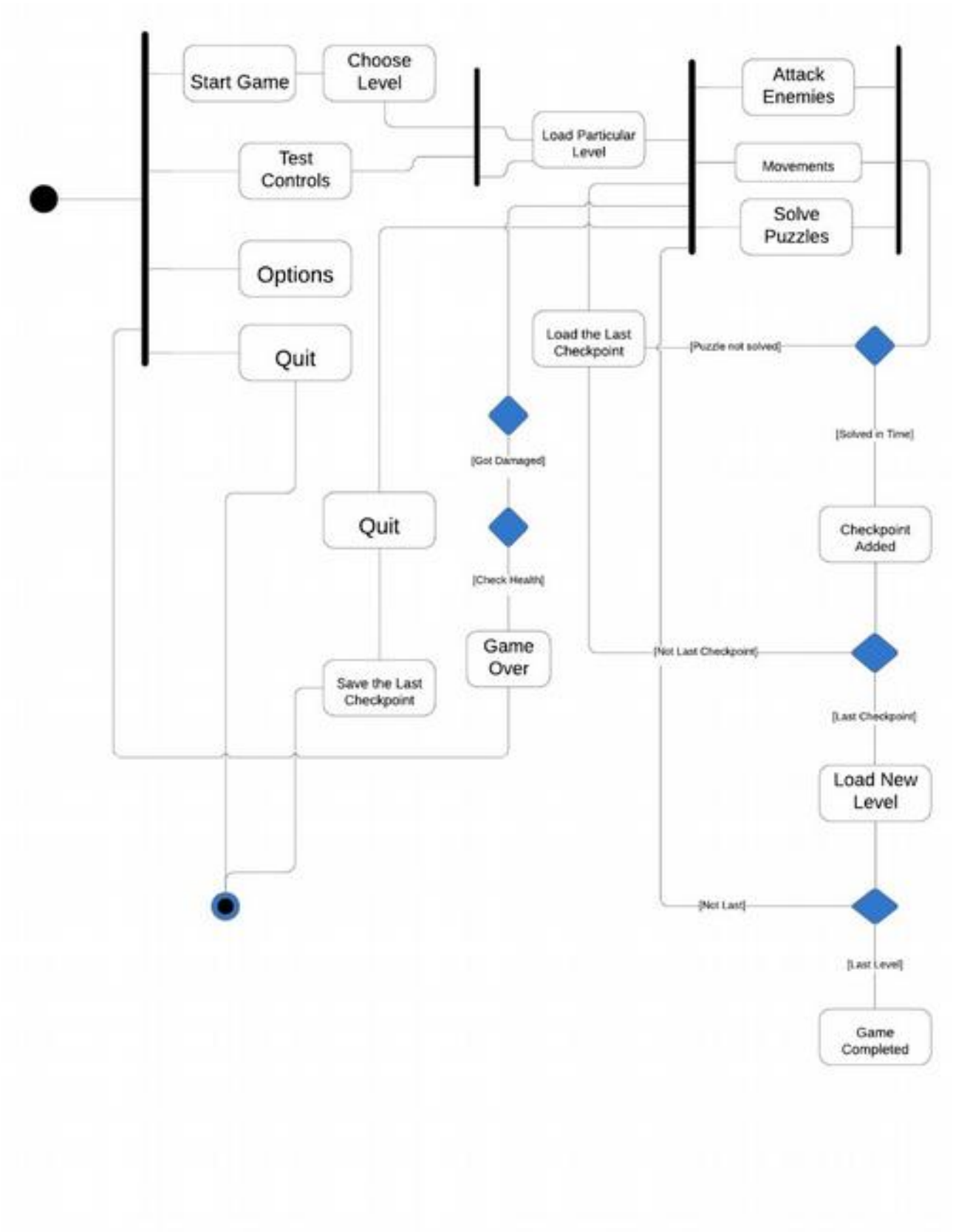
**Use Case Diagram :**



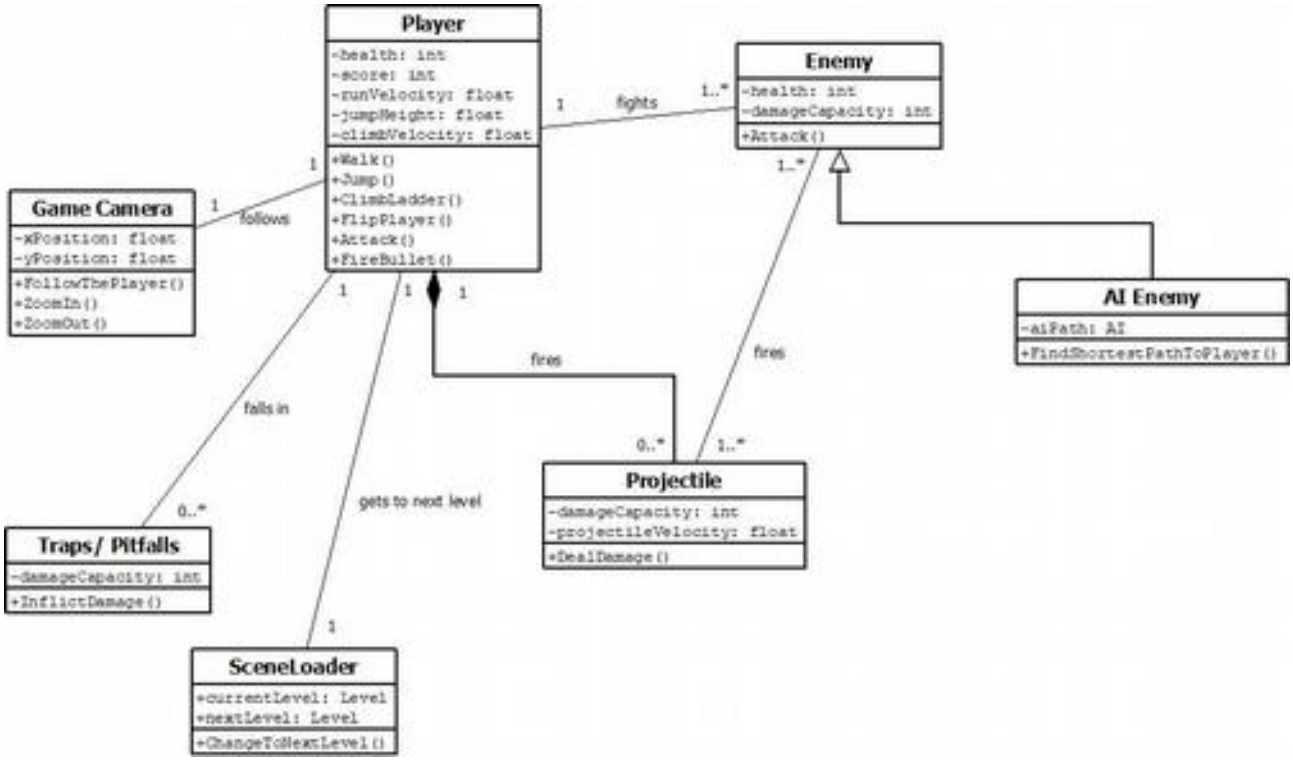
**Use Case Diagram Description:**

Here our main factors was player and enemies but along with them there are programmers ,designers and game manager who develops the game, calculating scores and decides game complexity. Role of player describes he can perform various tasks . First he starts the task(entering into game) and timer gets set. After that, he can damage enemies by shooting bullets towards them and also he can take damage when he gets shot by bullets from enemies. Also he can collect treasures to increase the score. After that if he wants he can end the game and timer will get stop at that point. If his health reduces to zero, he dies . In game's menu options ,he has various choices of selecting game's levels ,also he can see instructions for game and edit the test controls over there. He has various move controls of fire bullets,running, jumping and climbing. Programmer's role is that he can code various levels here and also he can fix bugs and improve the efficiency of game. Designer can make various animations and can change sprites and animations to make the game more effective. Making and editing Questionnaires by taking the opinions from group of peoples is done by game manager. He can also does analysis of game and generate feedback for to know more about game's performance. On other side we have enemies who can perform operations like walking towards players and attack players by using AI which gives shortest path towards player. AI using A* algorithm for detecting shortest path from enemy to player. Game manager at backend calculating score by using time constraints and treasures collected by player. He can increase health of player if player takes booster and if player gets shot by enemies bullet then he reduces health of player. These is how use case diagram works.

**Activity Diagram :**



**Class Diagram :**




**Flow of modules:**

With the help of adobe photoshop, different frames of pixel art characters' performing different actions will be created and merged together in a file called as a sprite sheet. It is a normal .png or .jpeg file, with different frames. Also, original music will be made with the help of online audio tools and/or mic. These are the asset creations that will be used in unity, the game engine. The asset logic will be written in C# programming language integrated with the unity framework. The sprite sheet created will also be given a meaning by splitting it up and creating an animation by using the unity's features. The animations will be attached to the scripts and objects thus outputting a game. Unity's 2D physics engine helps with the physics concepts like gravity, translation, rotation, etc. Other than that, there are different kinds of colliders, trigger events that could be used for helping with the coding logic. Some maths functions will be used as well, and instantiation of the objects can also be done dynamically. Player can solve puzzles, attack enemies, score and also instantiate projectile objects i.e. bullets which are of the damage dealing type. The AI script will be then written which will help the AI enemies find the shortest path to the player. After the game's logic has been implemented, different levels with different complexities could be designed by using Tile Map feature. Finally, the game can be deployed in Xbox, PC or android. After that, testing is done and the game will be ready for the release.

**Module 1 : Implementing the basic Player and Enemy Movements.**

Different scripts are to be written for different kinds of objects. For eg, a script named "PlayerMovement.cs" should be written in order to control the basic player movements. We can

run, jump, climb, fire, etc. The player can instantiate a projectile. As soon as its instantiated, a “Projectile.cs” will start running on that projectile object. Apart from the player, the “Enemy.cs” script has to be written and given to the enemy object. Even enemy can fire projectiles. The “Camera.cs” could be written to control the camera movements which follows the player or Cinemachine could be used which is a component of Unity. Script or component, the output will be the same. There would be other objects and their respective scripts too. Basically, this module focuses on the basic movements and the logic associated with it.

**Module 2 : Implementing the A* AI Shortest path finding algorithm on the enemy.**

Some enemies would be given AIpath.cs script which is nothing but AI. With the help of this script, they can find the shortest path to our player and can attack the player.

**Module 3 : Designing of various puzzles**

Various puzzles in the game such as taking the keys and unlocking the doors would be designed in this phase. This would form the theme of the game. With the core base logic designed, with minor changes different puzzles could be made.

**Module 4 : Designing of the respective sprites**

To implement the coding of the designed puzzles, first their respective pixel arts should be made. Also, the logic that would have been also implemented for player, enemy, etc objects, those pixel arts would be designed by using adobe photoshop and applying the same in the unity framework.
Module 5 : Implementing the puzzles
Designed puzzles will be given the logic and will be put together with the animation sprites. This will include writing additional scripts, one for each of the puzzle deciding the sequence of puzzles and which puzzle to be put in which of the levels. All those decisions and coding will be included in this module.

**Module 6 : Implementing Health, Score, Die system**

The Player would lose health when it falls into the pitfalls or traps or when attacked by an enemy. It could even die and kill other enemies, with everyone having their own damaging capacity. Also, the score could be affected by collection of some treasures such as coins, diamonds, etc. The score would also be affected by the killing of the enemies. Likewise, the logic of health, score and die system as well as their respective animations and particle effects with the help of Particle System component of the unity game engine will be the scope of this particular module.

**Module 7 : Making original music track and sounds**

Online audio tools will be used for creating an original music track to give the game the feel that its made for. Entertainment and a sense of accomplishment, happiness is the target of this game and animations and sounds plays a very vital role in all of this. Also, different sounds such as the key picking sound, door unlocking sound would be made by using a mic. The future scope of this could also include recording a narrator in order to guide the player in the adventure.

**Module 8 : Fixing any bugs**

The other additional coding includes the fixing of the bugs. Even the basic movements like walk, jump, etc has a lot of bugs and that would be the target for this module.

**Module 9 : Preparing the Questionnaire and getting feedback**

To test the game, different age group people will be given this game to play and an overall feedback of the same will be taken from them by given them a set of questions. These questions would be prepared in this module and after getting the feedback, analysis of the same will be done. That is the scope of this module.

**Module 10 : Any improvements**

Based on the feedback, the game will be tried to improve. This is an ongoing process and with each feedback, the game will become more and more robust with logic and animations, thus making an aesthetic 2D game for entertainment.