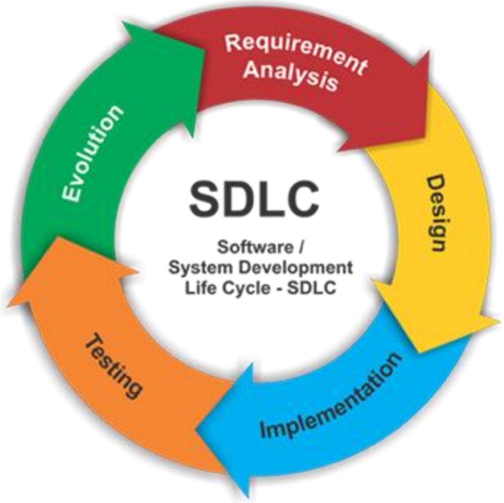
Task 1 – Analysis



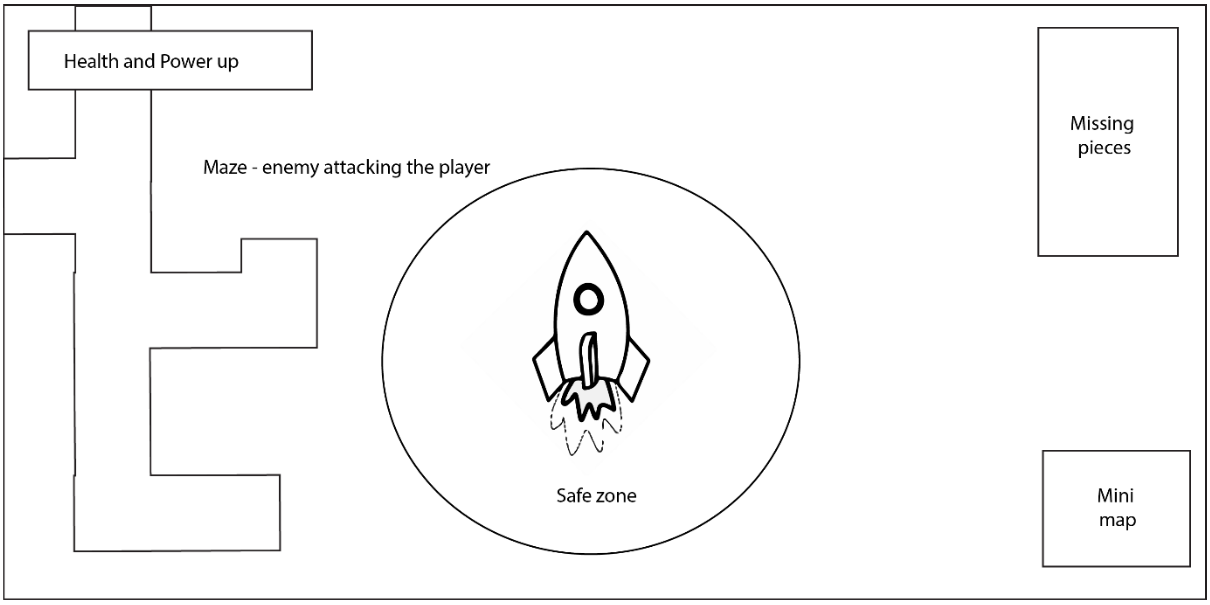
The model that was chosen for this game.

Requirement analysis

We start to think on what the game will be, as we were discussing different ideas for the game we decided that we wanted a type of hide and seek element to the game or "chase" in this case. In the end we ended up with the idea to create a character that would run around in an area that would be able to collect certain objects. But we needed to have the protagonist to be able to "chase" the player around the only problem was that we needed to use an AI (Artificial intelligence) to make this work. The first thing we did was research how we would be able to create this AI, after some research we found that we could use Unity's NavMash system, what this does is create the path for the enemy to follow. Then we needed to create a better concept for the game, we decided to create a robot that was stuck in a factory and the aim for him was to collect different parts to create a rocket ship, while being chased by another evil robot. We also decided to add a pick-up system where the player would be invincible for a limit amount of time.

Design

The idea is that when the game starts the first thing the player sees is the main menu with "Play" and "Quit" options. Button "Play" would start the game and the button "Quit" would close the game. The game would start with the player next to the spaceship that is missing some pieces. While player is next to the spaceship he would be safe and nobody would attack him. When the player leaves the save zone the enemy would start following the player till he catches him. The only why would the player survive if he is running or he collects the blue coloured power up and activates it which helps to kill the enemy and he respawns to random place and starts following the player again, unless he is in the save spot. The player would have only three times, till the game is over. To win the game, the player would have to pick up all the pieces of the spaceship scattered around the maze. The plan for the visuals is that the health and power up would stay together on the top left side. It would be useful to have a section on the right that would help the player by showing the pieces he is missing to collect. Also, mini map should be included that would help the player to be aware where he is and not to get lost.



Implementation

After all the requirements are prepared the way the game will be implemented step by step like so. While Darren Camilleri creates the 3D models, the first raw prototype of the enemy will be programmed to test the reliability of the navmesh AI agents provided by Unity. After this is tested the player will be added to allow implementation of the interaction between player and enemy done by Daniel Cassar. After the 3D models are done Joana Nummelin will then texture the models to give the colours required to match the theme chosen. While that is being done Darren Camilleri will start working on the random spawning positions of items, after which it will be implemented in the game by Daniel Cassar to further increase the interaction between the player and the enemy by allowing the enemy to randomly spawn in different spawns. The same system will be applied to spawn the power-ups and the missing pieces of the Rocket Ship which the player will have to collect to win the game. While this is in the process Joana Nummelin will finish designing the UI elements needed to be implemented in the game for instance the hearts that will represent the health of the player. The music for the environment will be researched and tested by Luke Parnis further refinement will be done by Daniel Cassar when the audio is implemented in the final game to make sure the transition from one song to another is fluid. After Joana Nummelin completes the design of the UI she will then move on to program the mini map feature for the game. The graphics for the UI interface and the mini-map are implemented in the game by Daniel Cassar to add some of the final functionalities. As the final step the scenes will be setup for the game as to be able to test the game fully without interruptions.

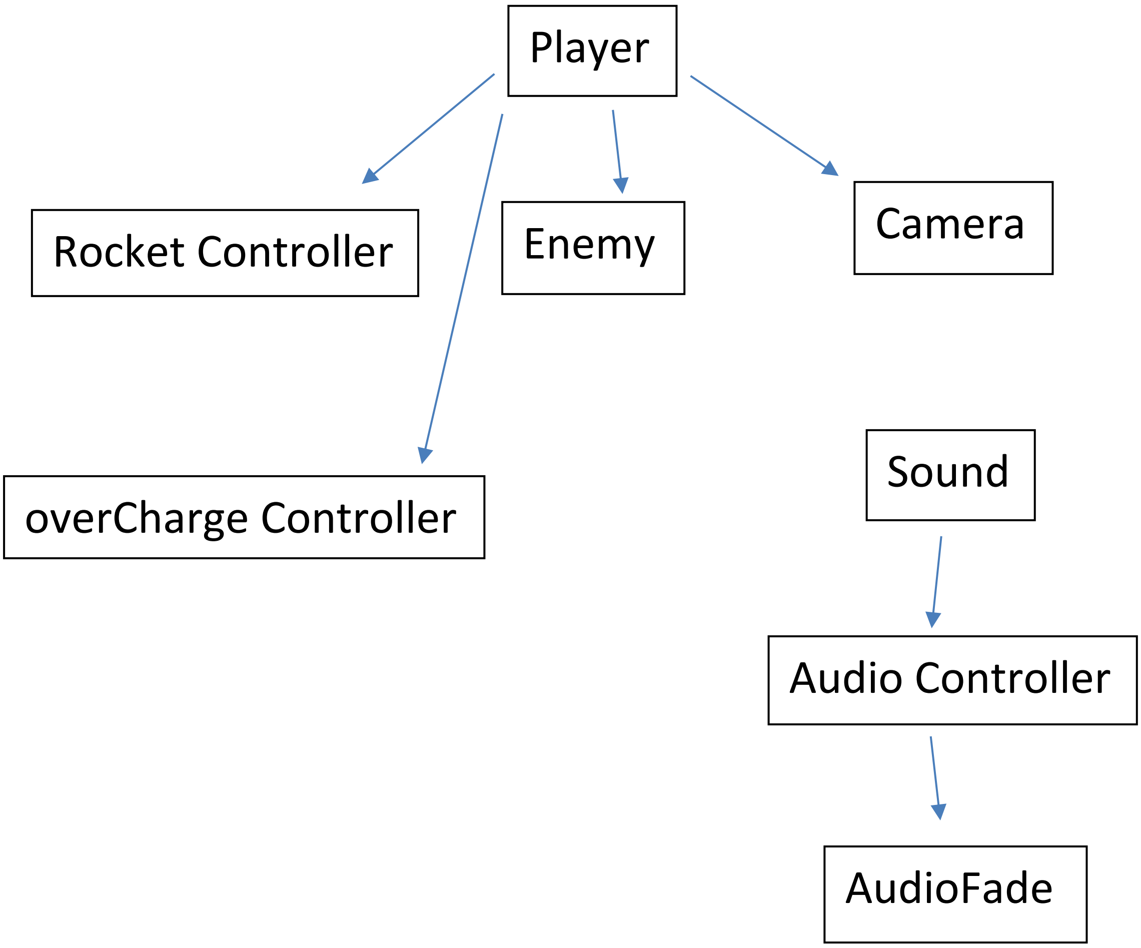
Testing

After the first prototype of the game is finished, it will be tested for any bugs that might be present. After which, each feature will be thoroughly tested separately to make sure nothing is missing.

Evolution

After each test run the game will be altered as needed to create a smoother experience in the game. This might offer the possibility to add more features to the game and continue to increase the size of the game.

Visually outline and discuss the major components and code structure.



The main features of the game will revolve around the Player controller as every interaction that will happen in the game will finally end up effecting the player. The rocket controller will use only a counter present in the player so as to be able to see which piece was collected. The camera will use the player as reference to be able to create a smooth follow on the player. The overcharge controller will use the player to be able to trigger the "GODMODE". While the enemy will use the player in order to set a destination to be able to navigate through the level and chase the player. The sound will be triggered by the player as he goes in and out of the safe zone and the switching from one song to another will be handled by an array in the audio controller which will run Coroutines to allow for the fade in and fade out effect.