**Enemy**

The enemy flies in circles until it notices the player. Then it starts to attack the player by flying in a circle around his head and shooting bullets every now and then. The enemy is killed by shooting at its shooting base.

They do not fully work as expected, because sometimes they intstantiate inside a platform, but apart from that they work pretty well. I am not yet fully satisfied with the prototype, but with a few minor changes it will suffice and it will be used in the game.

Creating an enemy does take more time than expected because it is not something that performs a single task, but something that has to respond on different situations which means that it has to be prepared for everything if you don’t want it to bug. This is still not completely done, so there will be more improvements that need to be made. It needs to be more intelligent and it has to get more skills.

**Player**

**movement**

The player is controlled by the WASD keys and spacebar to jump. The player its movement is fully physics based and Controlled by a PID controller. The control values are determined by trial and error. It also inherits velocity from moving objects when walking on a moving platform for example.

* Controlling a player this way works very good. It is a great advantage that it is fully physics based. It scales really well but it can take some time to get the control parameters right.

**Grappling Hook**.

The player shoots a hook that travels in a straight line. It attaches itself to an object  when it registers a hit. Then you can swing on the rope, fully dynamic and physics based.

* It works as expected. Right now it uses a rope that cannot bend visualy and cannot collide.
* The attach to other object is done without the transform.parent because Unity does weird scaling to the child if used.
* Trying to make the rope bend took way too much time. First tried it with physics made out of rigidbodies and wrote a custum spring damper script. This didnt work well because with a high springconstant/mass ratio the physics get unstable in Unity. This problem was solved by creating a custom integration algorithm so it was possible to solve the motion over some iterations between a Time.deltaTime. This gives realistic results and works pretty good. However this is only useable in realtime with few rigidbodies and not too much iterations. The algorithm can be made faster though.
* Then tried to make rope bends by raycast from the direction of the  rope. This also took too much time and didn’t work very well in 3D situations.

It is very likely to keep this prototype, but it could use some improvements, especially on the rope. Maybe cut the rope at a certain angle and have it not attach to all objects.

**GameWorld**

The Gameworld in the prototype exists out of procedurally generated platforms, with offsets to suit the low poly style. The generator creates a path of platforms, only the relative distance between the platforms is specified, the order of the different moves isn’t. The way the platforms spawn now is pretty decent, it does limit the possibilities however, and it greatly limits the possible tools. This prototype generator was finished pretty quick, but its successor is already underway.