**PROJECT**

**Proposal**

Road Rash is the game that was a part of my childhood. The game based around a reckless motorcyclist who races against other around the city and can do anything, similar to open-world gameplay.

However, the game never focuses on the citizen AI since they will just keep on getting run over by the user without fighting back. I want to implement a citizen AI where it can choose to throw rock to the user upon running by, by predicting user’s location at X time.

**Issue**

There are 2 possible approaches.

First solution is to implement a citizen AI similar to Agent Marksmanship project. When the main racer is at a fixed distance to the citizen, the citizen will recognise this and throw rocks towards the racer

Second solution is to implement

**Plan**

Simplistically, there will be a road created for the user to race on. The citizen will wait at a location. When distance from user to citizen is X, the citizen will throw rocks at the user at a fixed velocity. User will need to choose, based on self-speed and the rock’s speed, to slow down or speed up to avoid the rocks. If not, user will lose a small amount of health.

def fire(self, target\_pos):

enemy\_pos = target\_pos

if self.world.hunter.aim is True:

bullet\_speed =1000 if self.mode in ['Rock'] else 10

target\_pos = self.aim()

if self.mode == 'Rock':

for i in range(1):

self.world.add(RockBullet(self.init\_pos, enemy\_pos))

class RockBullet(Bullet):

def \_\_init\_\_(self, firing\_pos, target\_pos):

Bullet.\_\_init\_\_(self, firing\_pos, target\_pos +

Vector2D(randrange(-100, 100), randrange(-100, 100)))

self.radius = 5

self.velocity = 1000

self.velocity = randrange(18, 22)

A close up of a logo

Description automatically generated