**MARS VR Lab**

**CODING CHALLENGE**

-The game must be functional with clear start and exit points

The game has a proper game loop with clear start and exit points.

Goals : Get maximum teddy-dolls for goodie-points, find a gun[rocket-launcher], find a pink-pig-doll the final goodie-point to win the game. The game is lost upon loosing health till zero.

- The game must have a win and a loss condition

The game is won by finding a rocket-launcher and then finding the pink-pig-doll.

The game is lost if the health of the player reaches zero.

- Demonstrate usage of the Unity physics engine

Physics used for multiple collision and trigger detections.

The bullets are fired with force, uses physics.

Player Controller uses physics.

* Integrated a paid asset : RayFire a plugin to shatter game-objects.

- Custom animation or texturing are bonus points

- I was not able to make the existing turret animation work for shooting bullets, so had to manually create an animation for shooting for turret.

- You may use free assets from the asset store, bonus points for creating your own

- I used multiple free store assets for art.

- Used 1 paid asset mentioned above: RayFire plugin for demolition of GameObjects.

- You need to implement at least 2 design patterns, bonus points for more

(examples, Singleton, Command, Factory, etc)

* All the managers in the game are singleton pattern.
* I implemented two types of Factory pattern:
  + PropertyFactory
    - Used for the factory of Functions
  + BulletFactory
    - Used for deciding which bullet to instantiate at runtime.

Design Choices :

* I wanted to make a game which will be FPS and have rocket launcher instead of straight-shooting bullets. I chose Martian as the name obviously because I was making a game for MAR\_VR\_LABS.
* I really wanted to have some work involving RayFire plugin, was able to enjoy working on it.
* It was fun working on maximum possible requirements from a small game, like UI, Gameplay, music, Score, Level streaming, etc.

Corrections To make :

* There are many places where it felt like this could’ve been nice. Added quite some things, but had to stop at one point.
* Named the code Goodie and have puppets/toys being used instead😅.

Note :

* The script folder contains the code written by me. The rest of the folders might contain code, which is not written by me.
* 1 paid script asset used : RayFire
* Rest all assets used are Art Assets.

Overall, major time spent was choosing the art and integrating it and then discarding changes upon less satisfaction. 😁