Game Name: Gun Swing

Platform: Mobile IOS/Android

Store URL: https://apps.apple.com/pl/app/gun-swing/id1607866496

Unity Version: 2020.3.26f1

Contact: umurcg@reboot.ist

The task is completing development of a simpler version of the provided game. All arts and plugins required for development will be provided by the studio.

You may ask any question without hesitation.

Game Objective

The objective of the game is killing all enemies with guns before they pass player avatar.

Game Mechanics

1. Movements

Player movement will be towards in -Z direction of the world. While player's avatar move with constant speed, player will be able to change avatar's horizontal alignment with horizontal touch input delta. Horizontal alignment must be in limited range.

Enemies' movement will also be towards -Z direction. They will not change their horizontal alignment and they won't rotate during their movements. Their speed will be constant with time.

2. Enemy-Player Interaction

- Player avatar will be able to shoot enemies with it's guns.
- Enemies will have specific life amount. Bullets will decrease their lives' with each collision.
- When life of an enemy become less or equal to zero the enemy will die.
- When an enemy passes Z position of the player, it will be destroyed and variable amount of players' life will be decreased. When player's life become less or equal to zero, player will die and game will be lost.
- When all enemies is cleared from level, the game will be win.
- Player and enemy movements must be supported with animations.

3. Guns

There will be 4 different gun types in the game.

- *Pistol*: It will fire bullets in slow rate and bullet impact will be also less compared to other guns.
- *Kalashnikov*: Fire rate will be high, and impact will less.
- *Shotgun*: Fire rate will be low. Also it will shoot multiple bullets in one shot. Impact will be high.
- *Laser*: It will have power up duration, where laser energy is being collected. After that duration, the gun will blast a single ray that will kill all the enemies that are in the path of ray.

Models and effects will be provided for each guns

4. Swing Mechanic

In the game, player carry the guns with ropes that are attached to it's arms. Because of that player will not have full control on movements of guns. Player will start the level with variable number of guns. And it will be able to collect more guns during gameplay. For the sake of simplicity gun will not be upgraded like original game.

"Obi rope" plugin will be provided to attach the guns to player's arm.

Swing Mechanic Alternative

If time is limited, swing mechanic is not have to be implemented to the game. Instead, guns can be connected to arms without a rope physic mechanic. Line renderer may be used for visuality of connection.

5. Game System

The game will be played in single scene. Usage of multiple scene for each level is not valid for test case. Levels can be created with using scriptable objects and prefabs. Game logic must contain level system, win and loose logic. Also last level index of player must be saved persistently and game should continue from that level when the game is opened again.

6. User Interface

Simple user interface is required for game logic.

Game start screen with start button
Game lost screen with restart level button
Game win screen with next level button.

Unity UI elements can be used as placeholder.
Canvas must be set as Scaled with screen size.
Anchor points of UI elements must be set properly for different screen resolution support.

7. Levels

- 3 levels would be enough for the test case. The design of levels will not be evaluated for test purposes.
- Levels may be repetitive.
- Levels should not be generated with procedural way. There shouldn't be a randomness through levels.
- Levels are requested due to evaluating level system. However if prefab system will be used for levels, deriving levels from a base prefab would be best practice.

8. Additional Features

These features are bonus tasks. If you have time, and want to improve the test case, you can implement them.

Mine Traps: When player presses on a trap button a mine will fall from air. If an enemy steps on the mine it will explode within a radius range. In this range all enemies gonna be dead.

Rocket Traps: When player presses on a trap button a rocket will fall from air and explodes when it touches to ground. It will explode within a radius range. In this range all enemies gonna be dead.

9. Additional Notes

Clean and readable code is essence of the test case.

Creating clean prefabs and simple scene setup is required.

All design related variables must provided on components as serialized field.

Using known code design patterns will be plus in evaluation.

Commenting is not required but functions and variables that are not explicit should be commented shortly.

Unused functions and variables shouldn't be included in project.

Custom gizmos may be plus in necessary places but not compulsory.

10. Plugins and Assets

In project folder, there is prototype scene where all required assets included. Please use them as placeholders.

Also DoTween, Lean Touch and ObiRope plugin is included in the project folder.

Obi Rope Doc: http://obi.virtualmethodstudio.com/manual/6.2/ropesetup.html (Required)
Lean Touch Doc: https://carloswilkes.com/Documentation/LeanTouch (Required)
DoTween Doc: http://dotween.demigiant.com/documentation.php (Optional)

Using and sharing these assets out of the project without permission is forbidden!