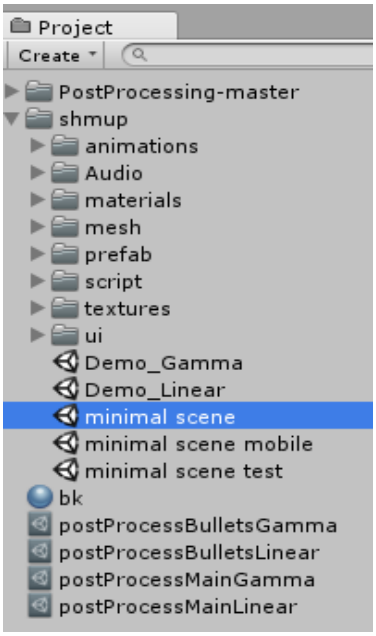
TUTORIAL:

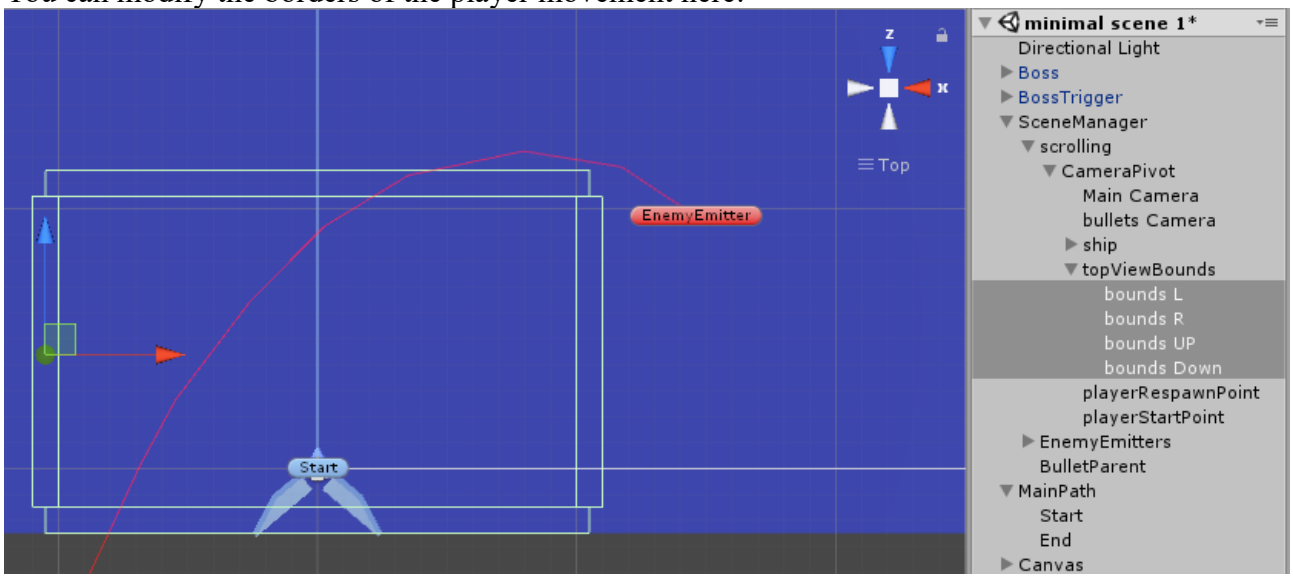
- Select “minimal scene” and press ctrl+d in order to duplicate it.



Then Double click that new scene to open it. This will be your new level.

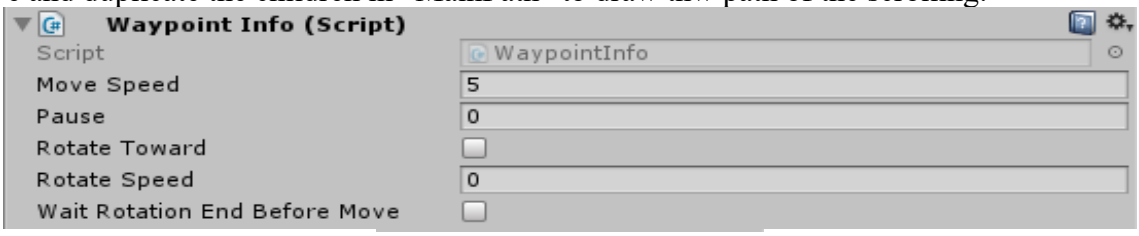
-Screen bonds-

You can modify the borders of the player movement here:



-Scrolling-

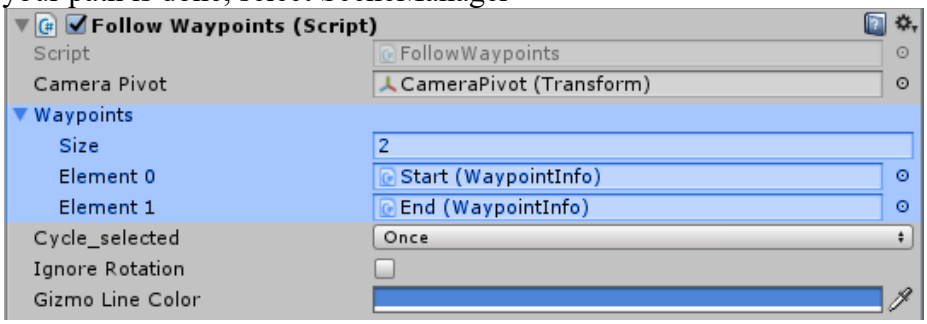
- Move and duplicate the children in “MainPath” to draw thw path of the scrolling.



At each waypoint you can modify the speed, add a pause or decide if rotate the view toward the

next waypoint.

- When your path is done, select SceneManager

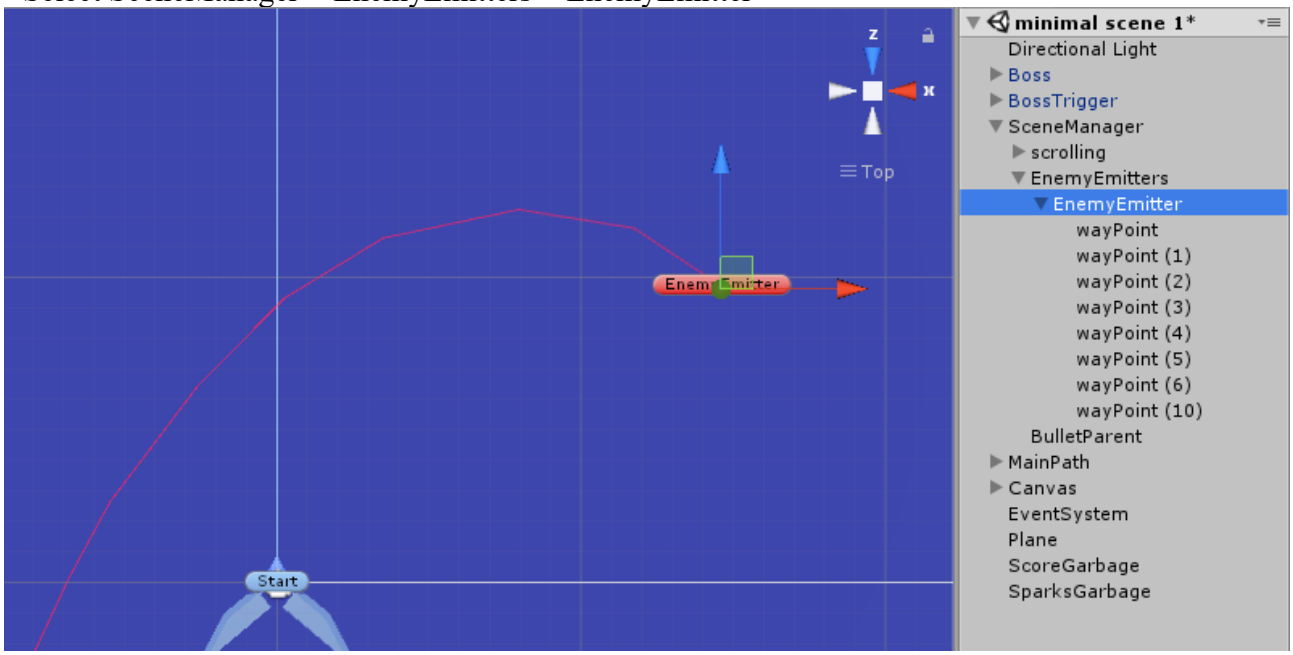


and feed the array in FollowWaypoints with the children in MainPath.
NOTE: Put them in the order from the start point to where the player will meet the Boss.

- Now take mines, obstacles and turrets form prefab > enemy folders and put the among the path to challenge the player.

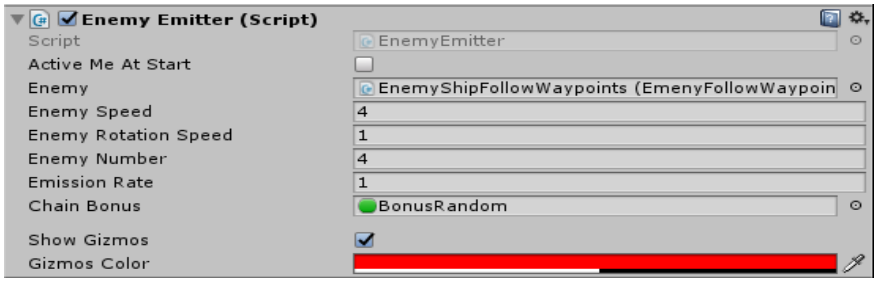
-Chain ships-

- Select SceneManager > EnemyEmitters > EnemyEmitter



This is the basic element to control the emission of a chain-ship.

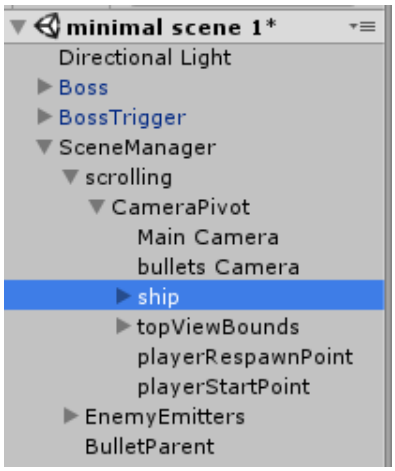
- Add, Remove and Move the wayPoint child in it to draw the path that the ships will follow. (When Move, remembered to keep the Y position allays on ZERO, or the ship will move on a plane different from the one of the bullets)
- In the inspector you can setup the chain-ship emission.



- Click and drag a ship prefab that MUST have in it the scripts “Enemy Follow Waypoints” and “Enemy” (example: The “EnemyShipFollowWaipoints” prefab), in the Enemy slot; this will be the ship generated.

- Setup enemy speed, rotation speed, enemy number and emission rate as you like

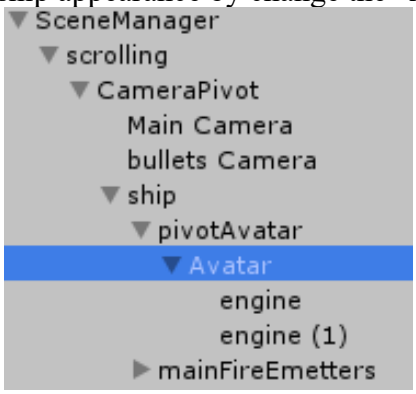
  - If you want that the chain give a bonus, or other prefab, if the player manage to destroy all the ships in the chain, feed the “chain bonus” slot with the prefab that you like.
- Now select “EnemyEmitters” again and in “Enemy Emitter Manager” script setup the “Delay” = when you want that the chain ship is ship sequence is activated.
- In order to have multiple chain ship:
  - Select and duplicate (ctrl+d) EnemyEmitter
  - setup the new EnemyEmitter
  - click and drag that EnemyEmitter in the array in EnemyEmitters > EnemyEmitterManager and setup the delay.
    NOTE: the delay refer to the last chain activated, so if a chain delay at 2second and the next a 0 seconds, it mean that both will be will be trigger simultaneously



-The Player Ship-

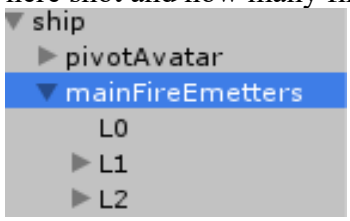
appearance

You can change the player ship appearance by change the “Avatar” with your 3d model



main fire

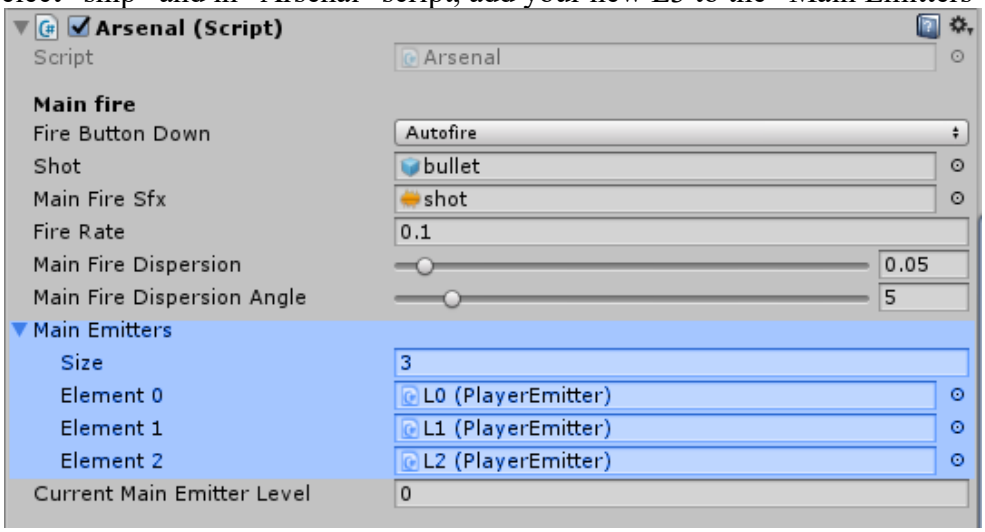
In order to change from where shot and how many fire emitter the player have,



Move and duplicate the transforms elements in each “Player Emitter” script

L0 = the basic fire power; L1 and L2 = the incremental fire power.

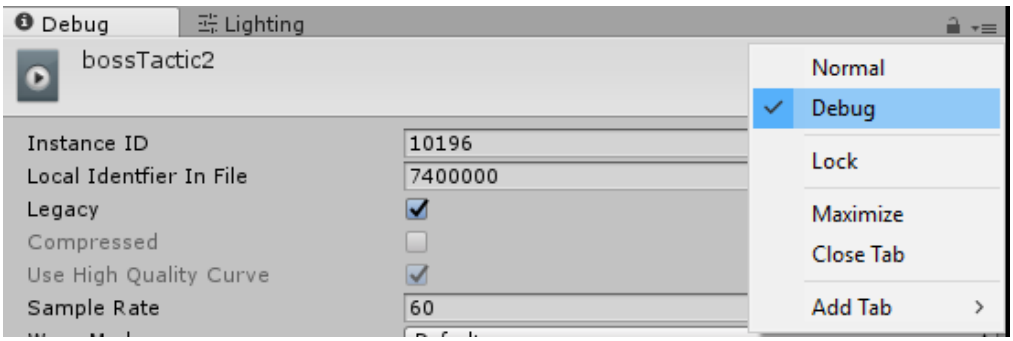
If you want more level of fire, duplicate (ctrl+d) L2 and modify the new L3, then select “ship” and in “Arsenal” script, add your new L3 to the “Main Emitters” array



-BOSS-

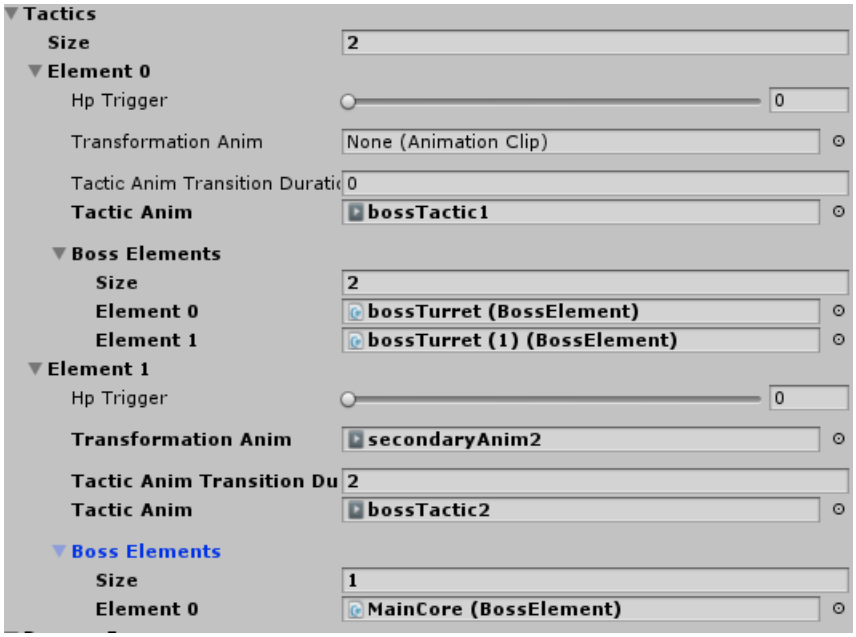
Bosses are the most unique elements of a shmup game, here some guideline using the “Boss” prefab as reference.

- Remember to have a BossTrigger prefab put in the pathway of the player ship in order to activate the Boss.
- All Boss animations MUST be “legacy”



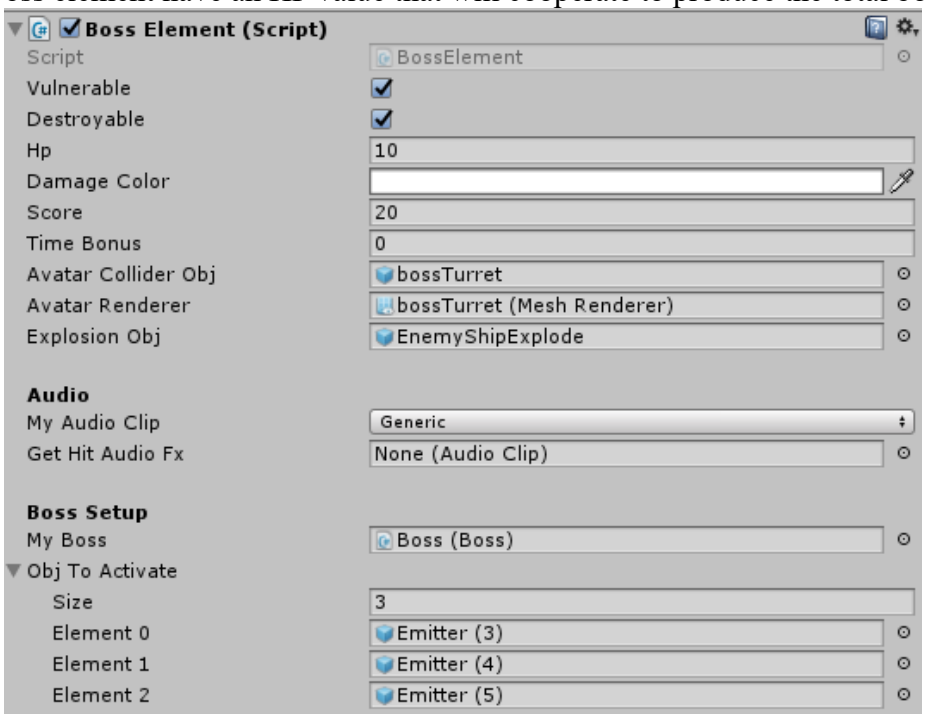
- to move around the boss keep still the Boss, and move the “pivot” instead using the animation component in the Boss
- to move the parts of the boss and play the die animation, use the animation component in the pivot instead

- Most likely you want that the boss change his behavior when wounded. In order to archive this, create different animations and feed them in the tactics array



Also in order to activate various enemy weapons feed the “boss Elements”

- Each boss element have an HP value that will cooperate to produce the total boss HP



It will have also “Obj to activate” when it is turn On (most likely emitters to shot the player)

SCRIPTS:

- **Enemies:**
  - **boss**:
    - **Boss** = Inherit from Enemy.
      - Hp is automatically setup to be equal to the sum of all "BossElement", so use them to modulate how much the boss is strong.
      - SecondaryAnim is the animation component used for the animation that not are movements of the whole boss.
      - IntroAnimation = how the boss show himself when the battle start (Play once). It will be invulnerable until this animation end.
      - DieAnimation = the animation to show when boss HP == 0 (Play once).
      - NextTacticTrigger = What must happen to make the boss change tactic
      - Tactis = how the boss move and what boss element activate in each tactic. With this you can change the behavior of the boss when it is damaged.
        - If NextTacticTrigger == percentile HP, the next tactic will be triggered by "hpTrigger", otherwise it will be triggered when all the bossElements in the current tactic array will be destroyed.
        - TranformationAnim = you can show a transition animation to play once, when the boss pass from a tactic to the next (example: boss get angry and transform himself)
        - TacticAnim = the new scheme of movement of the boss (play loop)
      - DamageFxs = allow to turn on some particle effect when the boss HP is under a "hpTrigger" percentile amount.
      - MovableElements = this array store all the element of the boss that are animated in order to restore them to original position when the game stage is reset.
    - **BossElement** = Inherit from Enemy.
      - Hp = this will be add to the HP of the "Boss"
      - MyBoss = the "Boss" script that coordinate all the bossElements of this stage boss.
      - ObjToActivate = All the gameObject to turn On when the "Boss" tactic turn on this "BossElement"
    - **BossTrigger** = Collider trigger that start the Boss behavior when the player enter in it.
  - **enemyWaypoint**:
    - **EmenyFollowWaypoints** = Follow the waypoint path give by EnemyEmitter
    - **EnemyEmitter** = Instantiate a chain of enemy ships that follow the same path. If player destroy all of them, the last ship give a bonus gameObject.
    - **EnemyEmitterManager** = Activate the EnemyEmitters after a delay. Delay in express in seconds and refer to previous emitter activated, so 0 mean at the same time of the previous and 0.5 after half second from the previous.
  - **AutoAiming** = Turn a turret forwards the player.
  - **CheckIfVisible** = Enable the gameObjects on screen and disable these out of screen.
  - **Enemy** = Define how much HP have an enemy and what is his score. ExplosionObj will instantiate a gameObject of your choice when the enemy is destroyed (like a particle explosion). It Contains: `PulseColor(Color color, float maxIntensity), TakeDamage(float damage), DestroyMe() and RestoreMe()`
  - **EnemyAvatar** = Get the damege from his collider and pass it to Enemy script.
  - **Mine** = Inherit from Enemy. It could be triggered by timer or proximity. Shot many bullets when explode.
  - **Rotation** = Parametric rotation animation.

- **Gui:**
  - **mobile:**
    - **TouchFire** = Virtual UI Fire button for mobile
    - **TouchPad** = Virtual UI swipe area for mobile
  - **FloatingScore** = Show an animated floating score when an enemy is destroyed.
- **Misc:**
  - **Bullet** = Inherit from Weapon.
    - Harm decide if the bullet will damage the player or the enemies.
    - Damage = how much hp stole the hit
    - Speed array, speedDuration array and acceleration allow to modulate the velocity to the bullet, so it could move with a more complex behavior that constant velocity.
    - Aimed, aimingDelay and maneuverability make move the bullet toward the player. If Aimed is false, the bullet will move forward the orientation of his emitter.
  - **Emitter** = This is the script that shot the bullets. It allow to setup what bullet shot, how many, the length of the blast and to modulate the precision using "shotDispersion" and "shotDispersionAngle"
  - **EmitterGroup** = Allow to pass the same setup to all the emitter in his myEmeitter array.
  - **LaserBeam** = Inherit from ReusableWeapon. Generate a laser as LineRenderer + Raycast as weapon for player and for enemies. Public float warningFxDuration and public GameObject warningFx allow to show an effect before to enemy shot in order to warning the player before the shot.
  - **OutScreenTrigger** = Destroy the bullet when it exit from the screen
  - **PermamentElementOnMap** = Turrets, desctructable Obstacles and every other element that can be damaged or destroyed, need this to be restored when the lever restart.
  - **SceneManager** = Manage the win/gameOver screens and the reset and restart of the scene. It also update the score UI.
  - **Spark** = Recycle the spark particles.
  - **TimeManager** = manage the countdown.
  - **Weapon** = This is the main class of all weapons. It set if the weapon harm the player or the enemy and how much damage it does.
- **Player:**
  - **Arsenal** = All the player ship weapons are managed here.
    - <u>Main fire:</u>
      - FireButtonDown:
        - Fire Forever = Ship shot always without need to press the fire button
        - Autofire = Ship shot keep firing until fire button is keep pressed
        - Charge Shot = if you press and release fire, ship will fire one shot, instead if you keep the fire button pressed, it will charge a bar and, when released it will shot a special bullet
      - shot = the basic bullet prefab fired by the player
      - fireRate, mainFireDispersion and mainFireDispersionAngle set the frequency and precision of the shots
      - mainEmitter array store from where the bullet or bullets will be shot
      - currentMainEmitterLeves = the current mainEmitter element to use (using mainEmitter and currentMainEmitterLeves and bonus you can increase the number of bullet fired by a single shot)
    - <u>Charged shot</u>: this matter only if FireButtonDown == chargeShot
      - miniShot = what shot if the charge bar is < 100% and >50%
        You can leave this empty if you want.

        - Charging_shot = what bullet fire if the bar is full. It is an array, so you can decide to have multiple bars to fill with incremental power
    - Secondary fire: (what happen when you press the secondary fire button)
        - Secondary weapons consume energy. You can set a number greater than zero in "Auto Refill Energy Rate" to continuously refill the energy or don't (so the player could recharge only if grab the right bonus.
        - Secondary weapon array store and setup.
            - By default there are 4 weapons:
                - Missile = auto-homing missile
                - Laser Beam = continuous laser
                - Force Field = a defensive sphere around the ship
                - Spears = a melee attack
            - If you want create your own weapon remember this:
                - for bullets and other stuff that you shot:
                    - set "Shot Model" == "Shot one new each time"
                    - leave Reusable Weapon empty and feed "emitter" and "bullet"
                - for on/off weapon (like laser):
                    - set it as "Enable the same"
                    - set fire rate to 0
                    - leave bullet empty and feed the Reusable Weapon
- **Bonus** = Generate a random bonus with weighted odds. It could give lives, energy for the secondary weapon, upgrade the main fire level and select the secondary weapon in use.
- **ForceField** = Inherit from ReusableWeapon. It set on/off the secondary player weapon force field.
- **Melee** = Inherit from ReusableWeapon. It trigger a MeeleAttack() as secondary weapon of the player ship.
- **PlayerController** = Manage the various input supported and set the player ship speed and tilt. If InputType == VirtualButtons, it take the inputs from MoveZone and FireZone scripts
- **PlayerEmitter** = Store the positions of the main fire emitters of the player
- **PlayerShip** = Decide, and show in the UI, how much lives the player have. It also setup the respawn behavior (position, speed and invulnerability duration).
- **ReusableWeapon** = Inherit from Weapon. It is used by weapons that can be turn on/off, like lasers and force field.

◆ **Waypoint:**

- **FollowWaypoints** = Movable turrets and the scroll screen use this. You need only feed the pubblic WaypointInfo[] waypoints array to generate the path, and decide if play it once (for scene scroll screen) or pingPong and restartFromZero for turrets movements.
- **WaypointInfo** = Indicate speed, pause and rotation to change when reach this waypoint