



# Ballistic Kit

DOCUMENTATION  
KRELI STUDIO

# Contents

1. System Overview	2
2. How to use	3
3. Detailed description	8

More examples and tutorials available on 

# 1. System Overview

The Ballistic Kit is a fully functional system for simulating the flight physics of projectiles and interaction during collisions with other objects. When designing and creating this system, I modeled on the mechanics of other existing games such as "World of Tanks" and "Sniper Ghost Warrior". The system has the vast majority of features that can be found in these games, which make up the trajectory of the projectile and its behavior during a collision. If you know how a projectile behaves in both of these games then the Ballistic Kit is a combination of these mechanics. Additionally interfaces, property drawers and inheritance, you can create new types of projectiles in a very simple way and modify existing behaviors. Create a bullet with AK-47, shotgun, explosive tank shell and even guided missile without any problem.

Ballistic Kit features:

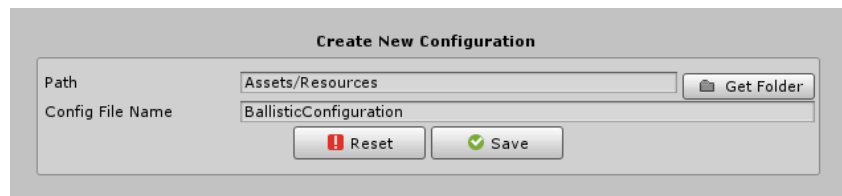
- Penetration of the object,
- Ricochet from the object,
- Blocked on the object,
- Acting gravity from "Physics.gravity",
- Air resistance/ Drag force,
- Acting wind from "Wind Zone",
- An extensive bullet hole decals system depending on caliber of weapon and material of object,
- The projectile's caliber determines which weapon can be used in,
- Many types of ammo in one weapon and the ability to change them,
- Many types of weapons in player and the ability to change them,
- Four fire modes in weapon: Single, Burst, Semi-auto, Full-auto,
- Configuration file as a Scriptable Object holding global data with a caliber of weapons, decals etc,
- Editor Window to handle and management the configuration file,
- Interfaces to simplify integration of Ballistic Kit into an already existing project,
- Technical settings such as trajectory resolution, RaycastHit sorting, Visual Debug Mode etc,
- Normalization, i.e. rotate projectile towards perpendicular to hit object to reduce thickness of armor to penetration,
- Custom inspector and property drawers for greater readability and easier handling,
- Several examples of bullets, weapons showing the possibilities,

## 2. How to use

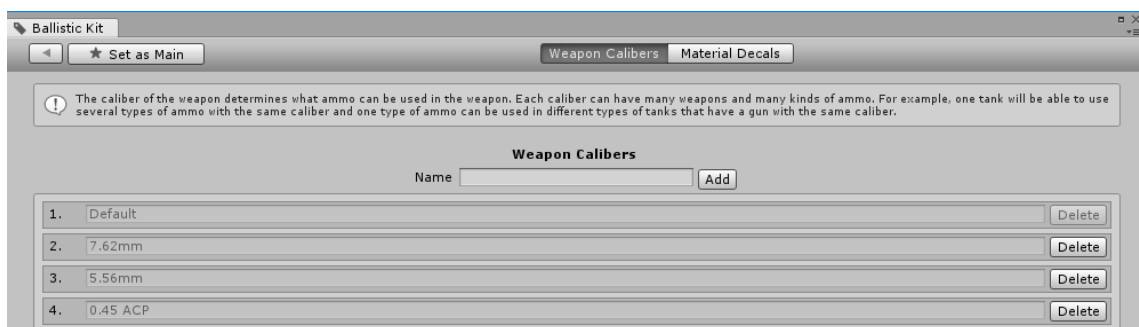
To show you how to use the Ballistic Kit, I will describe how to create a simple scene with a changer of weapons, several types of ammo and an object with structural points hidden behind an armored obstacle.

### Create and setup Configuration file

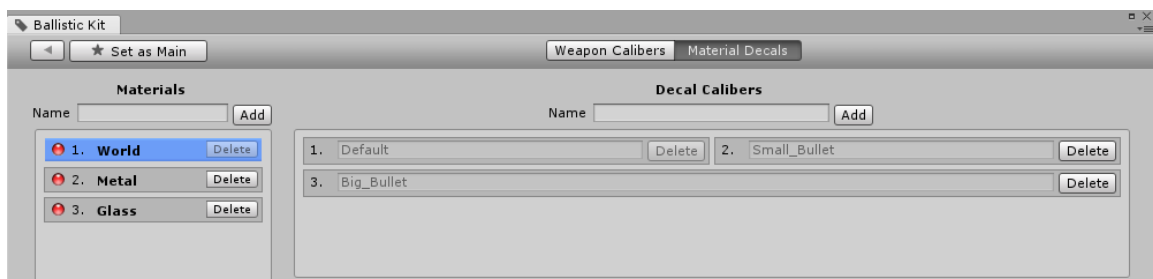
1. Open Ballistic Kit Window from Window -> Ballistic Kit and then create new config file.



2. Add few Weapon Calibers you want to use.



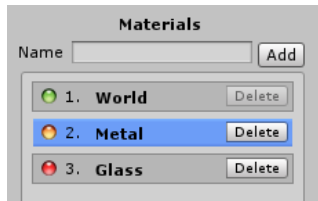
3. If you want different bullet holes on different materials and calibers you can do it in Material Decals tab. Add materials and decal calibers you want to use.



4. Select material from left list and add your decal prefabs to created table. Make it to all materials.

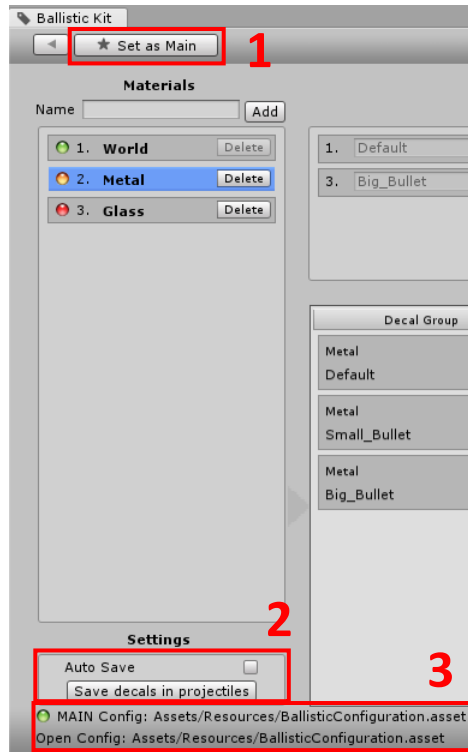
Decal Group	Blocked Decal	Ricochet Decal	Penetration Decal
Metal Default	Decal_Debug_Blockec Scale 1	Decal_Debug_Ricoche Scale 1	Decal_Debug_Penetr Scale 1
Metal Small_Bullet	None (Game Object) Scale 1	Decal_Debug_Ricoche Scale 1	Decal_Debug_Penetr Scale 1
Metal Big_Bullet	Decal_Debug_Blockec Scale 1	None (Game Object) Scale 1	None (Game Object) Scale 1

Color icon next to the material is information for you that all prefab fields are completed.



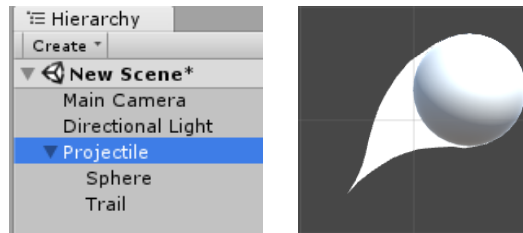
- Red - empty,
- Yellow - not completed,
- Green - completed

5. Set this configuration as Main (1) and save changes (2). At bottom of window is display what is main config and currently open (3).

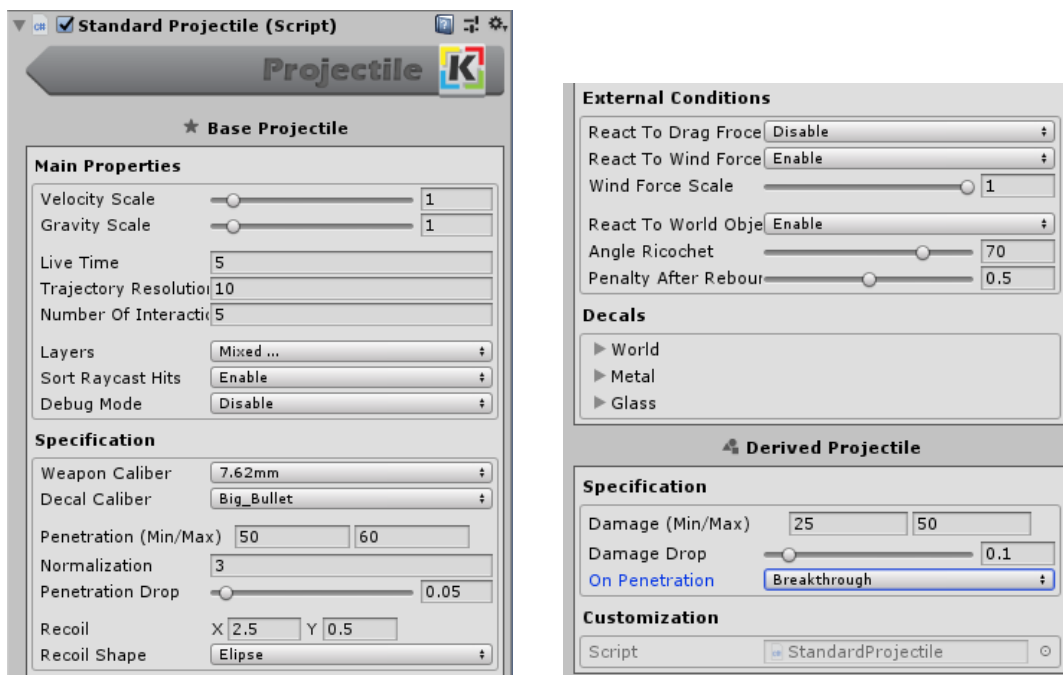


## Create and setup Projectile

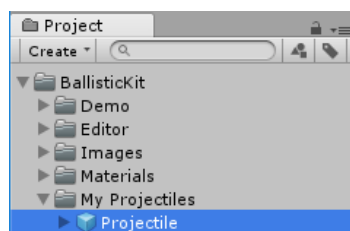
1. Create empty game object and add sphere object and trail as child.



2. Add one component from projectile components (Base, Explosive, Frag, Standard) to parent object and setup it. In new projectile, decals should automatically import from main config.



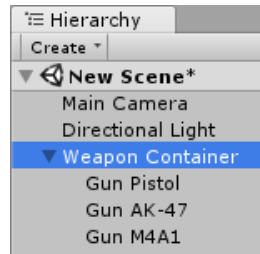
3. Save projectile object as prefab.



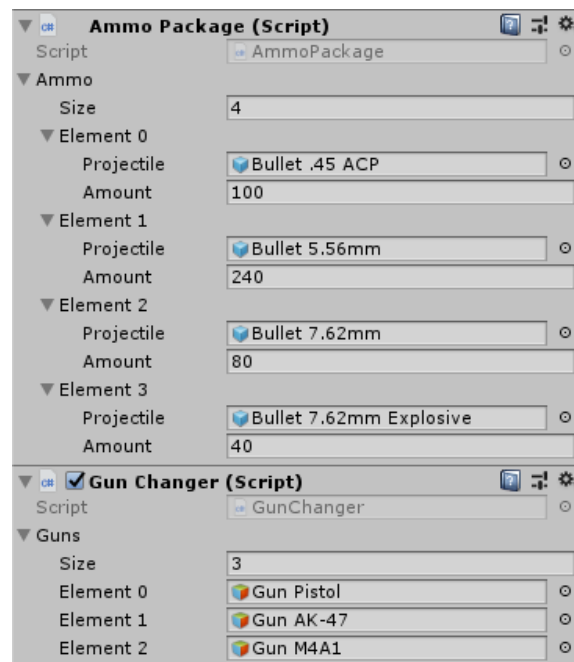
4. Create few more projectiles for different guns.

## Create and setup Guns

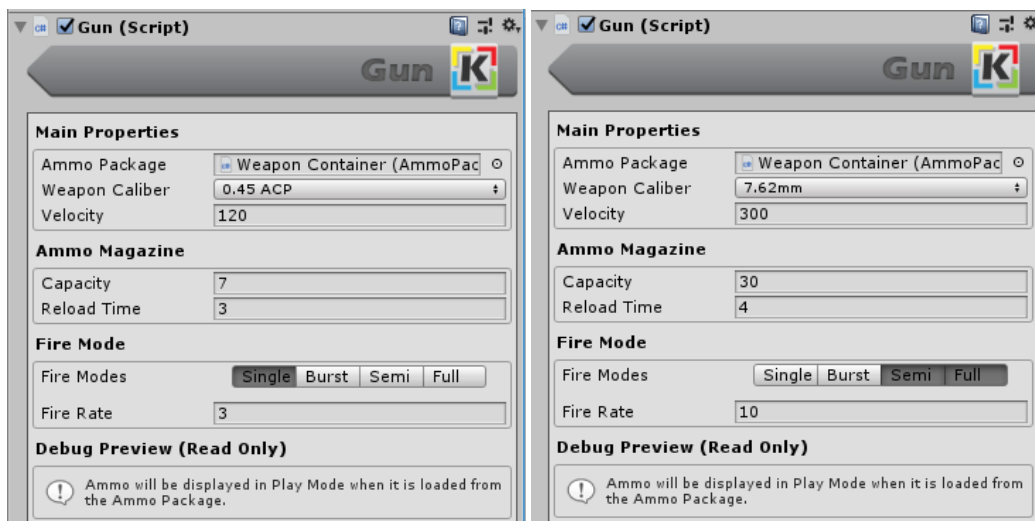
1. Create empty game objects hierarchy like below.



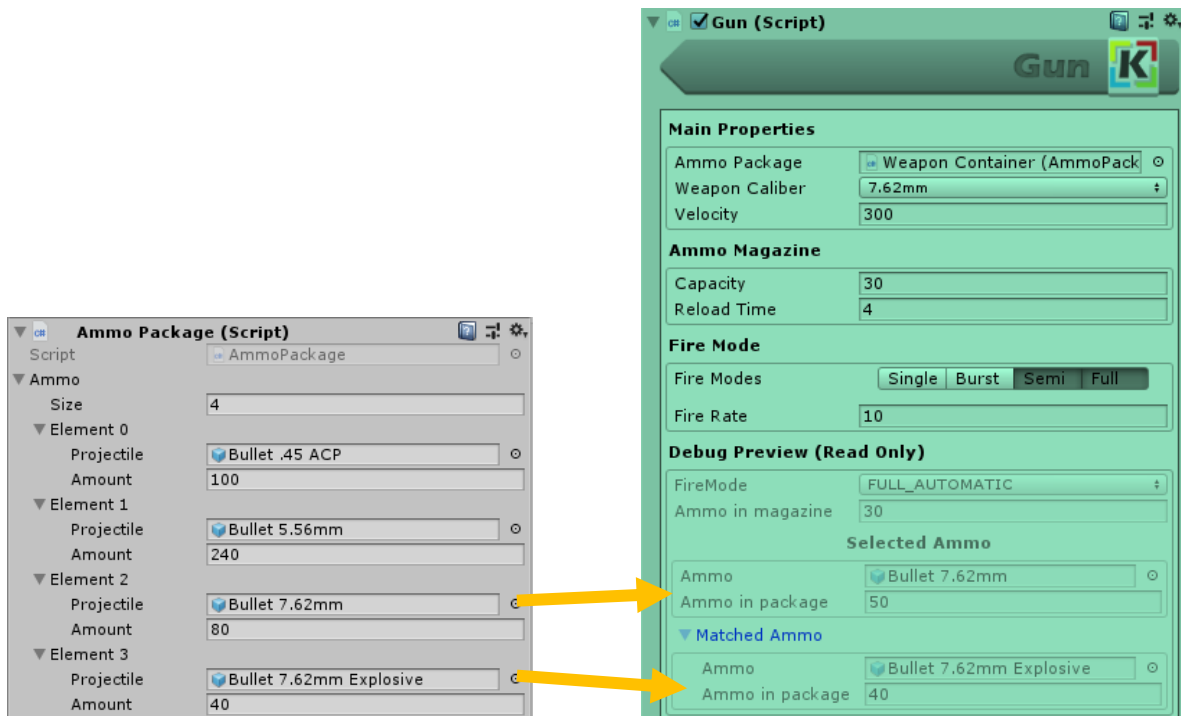
2. Add AmmoPackage, Gun Changer components to weapon container object and set projectiles prefab and gun references to them.



3. Add Gun component to all guns object inside weapon container and setup it.



- Its Done. Now you can test your guns! Note that the package contains two types of projectiles with the same caliber. Weapon that uses this caliber can switch between these projectiles. You can check what projectiles can be use in gun in debug mode in play mode.



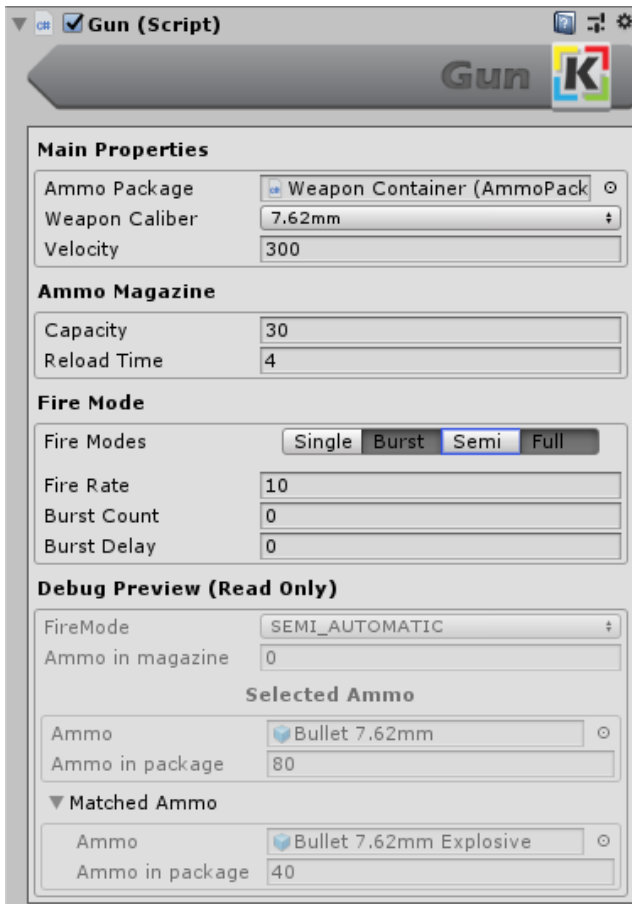
## Default Control Information

Action	Input Manager Name	Default button
Fire	„Fire1”	LMB, Left Ctrl
Reload		R
Change Ammo Type	„Mouse Wheel”	Mouse Wheel Up and Down
Change Fire Mode	„Fire2”	RMB, Left Alt
Change Weapon		1, 2, 3, 4



# 3. Detailed description

## Gun Component



Gun component is a representation of firearms, tank guns etc. Get projectiles from the Ammo Package component with the same caliber it has. Allows you to fire a projectile with the speed specified in the weapon.

**Ammo Package** – reference to the component containing the available ammunition.

**Weapon Caliber** – determines what ammunition matches to this gun.

**Velocity** – initial speed of projectile when firing.

**Capacity** – how many ammo is in one full magazine.

**Reload Time** – duration of reloading.

**Fire Modes** – select available fire modes in this gun.

**Fire Rate** – how many projectiles can be shot in one second.

**Burst Count** – how many projectiles will be shot in one burst.

**Burst Delay** – break between series in seconds.

**Fire Mode** – current selected fire mode.

**Ammo in magazine** – current amount of ammo in magazine.

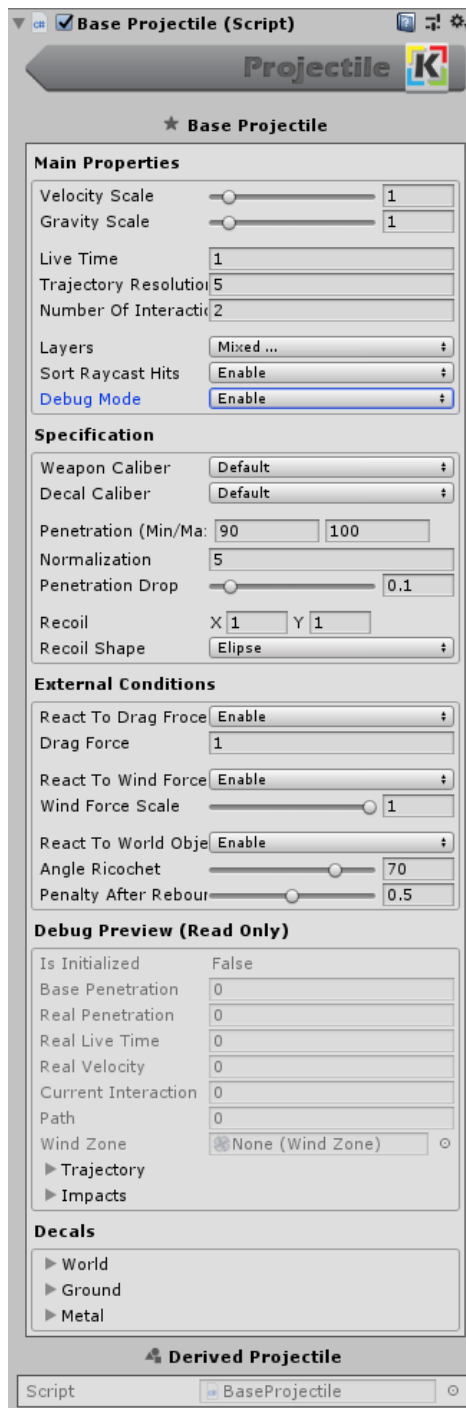
**Selected Ammo -> Ammo** – prefab of current selected ammo.

**Selected Ammo -> Ammo in package** – amount of current selected ammo in package.

**Matched Ammo -> Ammo** – prefab of available ammo to select.

**Matched Ammo -> Ammo in package** – amount of this ammo in package.

## Base Projectile Component



Base Projectile component is a representation of a ballistic missile, a bullet from a gun etc. This script includes all the mechanisms responsible for the flight trajectory, armor penetration capability and response to external conditions. Base Projectile does not do anything after the armor has been penetrated because it has no response implemented. Derived components such as Explosive Projectile, Standard Projectile etc inherit the behavior of the Base Projectile and add the behaviors of impact.

**Velocity Scale** – scales the initial velocity given by gun.

**Gravity Scale** – scales gravity acting on a projectile.

**Live Time** – time in seconds after which projectile will be destroyed.

**Trajectory Resolution** – number of trajectory points of projectile flight. The larger the number, the more accurate the trajectory is, but too much can drop performance and may cause errors in collision detection.

**Number Of Interactions** – number of possible projectile interactions (penetration, ricochet, etc.). Projectile will be destroyed after last interaction.

**Layers** – Layer mask with which the projectile interacts.

**Sort Raycast Hit** – in case the projectile hit several objects (in one frame) the hits are sorted depending on the distance to the projectile so as to determine which object was hit first. Sorting definitely improves the detection of multiple collisions in case the projectile is very fast and the objects being hit are very close to each other.

**Debug Mode** – displays flight trajectories, raycast, impacts etc. and print logs of behaviours projectile to the console.

**Weapon Caliber** – determines what gun matches to this ammunition.

**Decal Caliber** – determines what decals group will be used for this projectile.

**Penetration** – range of projectile penetration. Angle of impact on object affects penetration.

**Normalization** – An effect that causes the angle of projectile decrease on impact.

**Penetration Drop** – Percentage of initial penetration decrease over the 100m distance.

**Recoil** – recoil in X and Y axis.

**Recoil Shape** – rectangular or elliptical shape of recoil.

**React To Drag Force** – in trajectory of projectile flight air resistance will be included.

**Drag Force** – force of air resistance affect on projectile.

**React To Wind Force** – in trajectory of projectile flight directional wind (Wind Zone with directional settings) will be included.

**Wind Force Scale** – scale of wind force affect on projectile.

**React To World Object** – when projectile collide with an object that has no IBallisticInteraction interface (world object), the projectile is blocked and destroyed. If the option is enabled then the projectile can ricochet.

**Angle Ricochet** – above this angle projectile ricochet from world object.

**Penalty After Rebound** – how much potential projectile lost after ricochet (penetration, damage, physics force etc).

**Is Initialized** – Has projectile been properly initialized.

**Base Penetration** – Initial penetration drawn from the min / max range.

**Real Penetration** – current value of penetration at trajectory point . Base penetration with drop penalty.

**Real Live Time** – current life time of projectile is a result of initial live time reduced after collisions.

**Real Velocity** – current velocity of projectile is a result of initial velocity reduced after collisions.

**Current Interaction** – number of interactions that had a projectile.

**Path** – percentage specifying the trajectory traversed by the projectile (from 0 to 1).

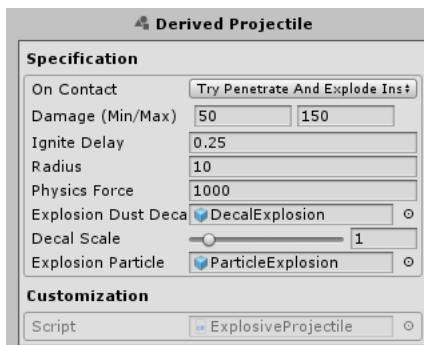
**Wind Zone** – wind force from the Wind Zone object acting on the projectile.

**Trajectory** – vector3 array with trajectory points

**Impacts** – references to transform hit objects.

**Decals** – list of decals group attached to projectile.

## Explosive Projectile Component



An explosive is a derivative of Base Projectile that explodes on contact with the target. During an explosion, the script checks if an object that can deal damage is behind an obstacle. If something is between the origin point of explosion and the nearest point of collider, then it means that the object is behind an obstacle and explosion damage is not given.

**On Contact** – determines whether projectile should explode at any contact with anything or try to penetrate the enemy's armor and explode inside the object.

**Damage** – range of projectile explosion damage.

**Ignite Delay** – distance in which the projectile will explode after a collision. This allows for example to break through the wall and explode 2m behind it.

**Radius** – explosion range.

**Physics Force** – add force to objects with rigidbody located in explosion radius.

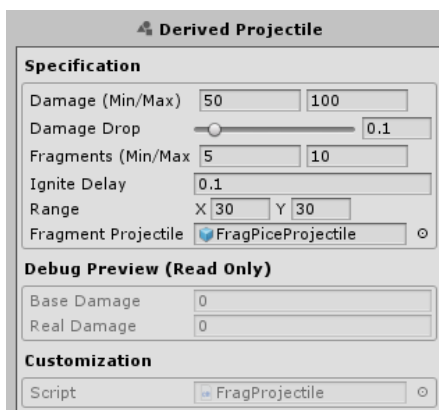
**Explosion Dust Decal** – a blast/dust decal created after explosion if the projectile explodes outside the object.

**Decal Scale** – scale of dust decal.

**Explosion Particle** – particle system of explosion.

**Customization** – section where variables of class derivatives will be displayed.

## Frag Projectile Component



**Damage** – range of projectile damage.

**Damage Drop** – Percentage of initial damage decrease over the 100m distance.

**Fragments** – range of fragments amount to create.

**Ignite Delay** – distance in which the projectile will fragmentation after a collision. This allows for example to break through the wall and start fragmentation 2m behind it.

**Range** – range of the angles in which the fragments will be directed in X and Y axis.

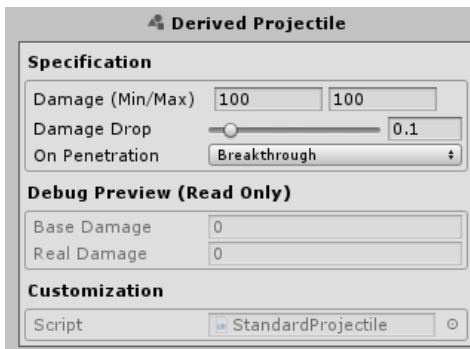
**Fragment Projectile** – Prefab with projectile component which will be created as fragment.

**Base Damage** – Initial damage drawn from the min / max range.

**Real Damage** – current value of damage at trajectory point . Base damage with drop penalty.

**Customization** – section where variables of class derivatives will be displayed.

## Standard Projectile Component



A standard projectile simulating rifle bullets. It penetrates armor and deals damage.

**Damage** – range of projectile damage.

**Damage Drop** – Percentage of initial damage decrease over the 100m distance.

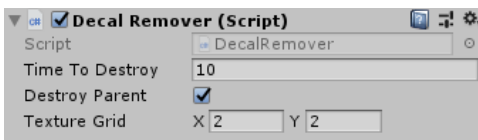
**On Penetration** - determines whether projectile after successful penetration is to be destroyed in the object, or pierce the object and fly further along the trajectory.

**Base Damage** – Initial damage drawn from the min / max range.

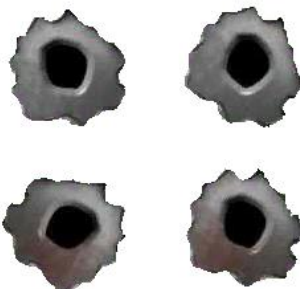
**Real Damage** – current value of damage at trajectory point . Base damage with drop penalty.

**Customization** – section where variables of class derivatives will be displayed.

## Decal Remover Component



Script responsible for the behavior of bullet hole decals.

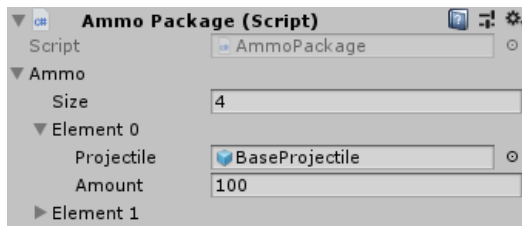


**Time To Destroy** - time after which the object will be destroyed.

**Destroy Parent** - whether parent object in which the component is located should be destroyed.

**Texture Grid** – split decal texture and randomly choose one of piece to get diversity.

## Ammo Package Component

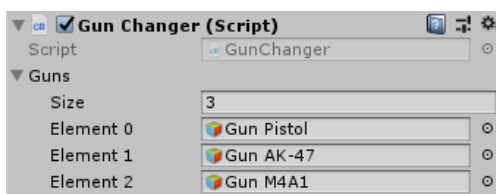


Collection of all player ammo to which Gun components reference and get ammo. It is a minimalist script that simplify integration with an existing project.

**Projectile** – Prefab with projectile component.

**Amount** – amount of this ammo.

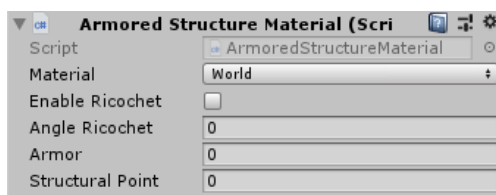
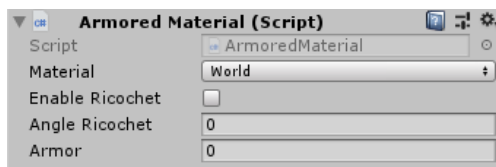
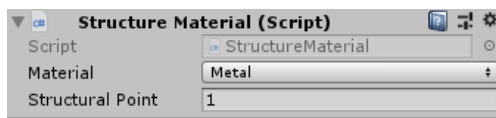
## Gun Changer Component



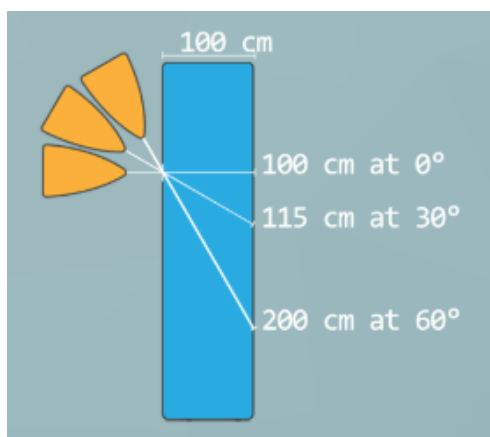
Component that allows you to change weapons contained in array. It is a minimalist script that simplify integration with an existing project.

**Elements** – references to guns game objects.

## Ballistic Material Component



Examples of materials showing how to create, add to the existing scripts support interaction with projectiles. Material without armor on hit by projectile is always penetrated and receives damage. If structural points are less than or equal to zero, the object will be destroyed. If projectile hits an object with armor, it calculates change to penetration and if armor was penetrated then projectile passes over and if object has structural points then object receives damage. As in game "World of Tanks", angle at which projectile hits object is of very important in calculating the penetration.



**Material** - determines what decals group will be use for this object.

**Structural Point** - structural points, health points that are reduct by damage of projectile.

**Enable Ricochet** - if selected projectiles can ricochet from object.

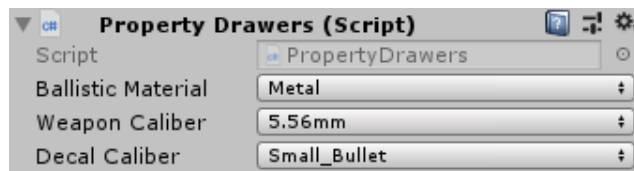
**Angle Ricochet** - above this angle projectile ricochet from object.

**Armor** - armor with which penetration power of projectile is compared.

## Property Drawers

```
public class PropertyDrawers : MonoBehaviour {
    [BallisticMaterial]
    public string ballisticMaterial;
    [WeaponCaliber]
    public string weaponCaliber;
    [DecalCaliber]
    public string decalCaliber;
}
```

Property drawers for Ballistic Material, Weapon Caliber and Decal Caliber will help you write your own scripts using the Ballistic Kit.



## Interfaces

```
public interface IBallisticInteraction {
    HitInfo Hit(BaseProjectile _projectile, float _angle);
}
```

IBallisticInteraction added to the any script will support for hits by projectiles. Just overwrite the Hit method and write action to be executed on Impact. As arguments, the projectile sends the own projectile script and the hit angle.

```
public interface IStructureInteraction {
    void Damage(float _amount);
    void Repair(float _amount);
}
```

IStructureInteraction added to the any script will support for dealing damage by projectile when it penetrate object. Object can be repair too.

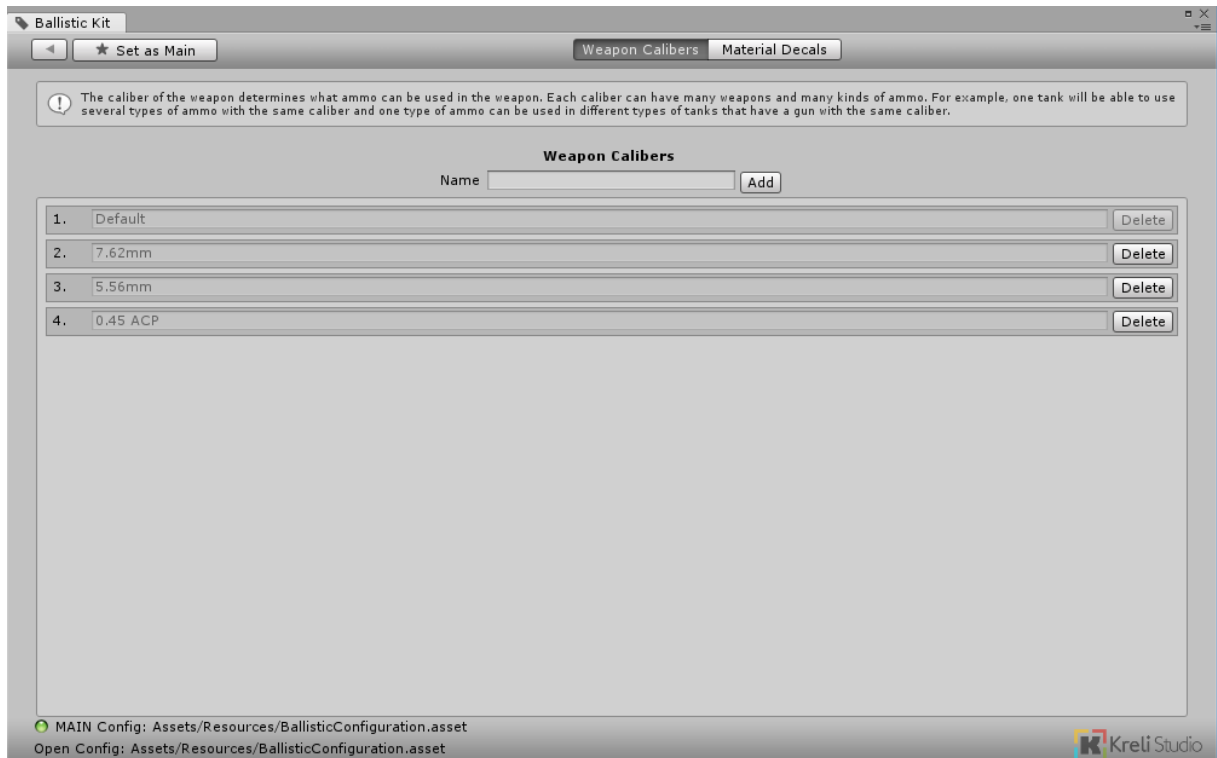
## Configuration File and Ballistic Kit Window



First screen that display after open the Ballistic Kit window allows you to create a new configuration file and to open an already existing file. At bottom window is information on which configuration file is currently set as the main (used by projectile, ballistic material and gun components) and currently open configuration file. The colored icon next to main config informs:

- Red - no file is set,
- Yellow - open config file is different than main config,
- Green - main config is open





In this tab you can create new weapon calibers that are set in projectiles and guns components. One projectile can be used in many guns with the same weapon caliber. One gun can use many projectiles with the same weapon caliber.



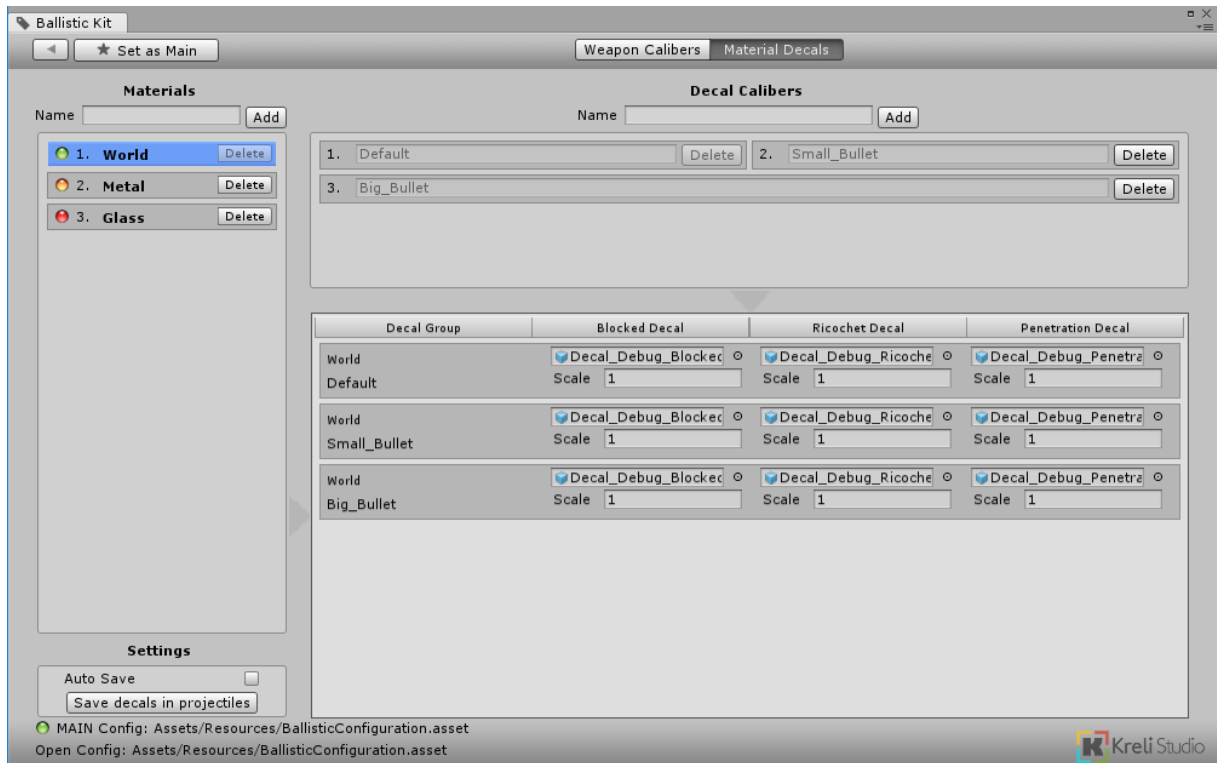
is back button when is pressed then config file close and back to first screen.



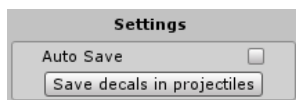
after pressed button currently open config file is set as main config.



switching between tab.



List on the left contains materials. List at the top contains decal calibers. The middle table contains a combination of both tables. To view and edit prefabs of decals groups, click on the material from list on the left. Then you can set prefabs and scales for the appropriate decals. This mechanism allows you to create different bullet hole decals depending on caliber of the projectile and the material with which it had a collision.



After any changes you have to save new decals groups to projectile prefabs in project. To do this press save button or check auto save toggle (but in big project with many resources and projectiles can little slow down Editor). I recommend to use save button.