WW2 Crew Weapons Pack 1

German and U.S. Light Mortars and Heavy Machine Guns

Intro - While there are many assets available for game developers to create WW2 battles, there are very few asset kits that deal with heavy weapons and their crews. Perhaps this is because of the increased complexity in creating these assets. Crew weapons involve many different game objects that need to coordinate their efforts to create a realistic and pleasing outcome.

It is the intention of this game kit to fill the need for the strategy game developer who desires a broader arsenal of weapons related to World War 2. That said, this also puts more work on the developer to ensure the assets perform well and interact with the player in the desired way.

I hope that this kit will be a one-stop shop for all things related to these specific heavy weapons. I have included all sound effects, particle systems, animations, and meshes with the hopes that the developer can simply code the desired behaviors and then have game assets that work well with their projects.

These assets are sized for use with other popular low-poly WW2 kits. The animations from those kits can be used with these soldiers (and vise-versa) so I did not want to re-create those same animations for this kit. Rather I wanted to expand on these other kits with more animations and meshes to broaden what is available for indie game developers.

It is quite possible that I have left out animations or other objects related to these weapons that would really help create a fantastic game. If you find in using these assets that there are a lot of pieces missing please let me know in the comments or email me directly at Bassmanpray@gmail.com.

Thank you,

Andrew Pray

Instructions -

Basic Fundamentals -

All Crew weapons have two states - Idle and Firing. There are no states for moving or aiming. It was my intention to leave this up to you, the game developer. (More on this later.) There are also no crew member deaths or options for the crews to leave or mount the gun positions. This will also be up to the game developer if these features are to be added. The animations

included in this kit are all the animations that are required for the idle state or the firing state. There are a couple of animations for the machine guns for charging and one animation for the loader of the big FlaK 38 to reload the magazine.

Aiming Principles -

In the real U.S. Army the total traverse angle allowed for a generic machine gun on a tripod is 30 degrees total. The field manual for the M2HB says that the total traverse is +/- 22 degrees. My game assets allow for a **total traverse of +/- 30 degrees** to allow more flexibility for the game developer. Move beyond these limits and the animation breaks down.

Deployment Considerations -

Heavy Machine guns and Mortars are simply too heavy for assaults and are employed primarily in a defensive role. While airborne troops did airdrops with a 60mm mortar they still were not considered a primary attack weapon. All crew weapons are for fire support and defense. Setting up the fire position for these weapons was very carefully considered in the real world since, once they were set up, they were not easily moved around or even shifted to new fire zones. Moving a crew weapon involves a lot of time that most game-battle-scenarios do not have.

It is most likely that your battle scenarios will have a “defence” and an “offence” side. These weapons will most likely be used on the “defence” side and start the battle already in-position and ready to fire.

Aiming and Firing your Crew Weapons -

Each “Plain” crew weapon is ready to rock with a publicly-addressable function, usually called “Fire()” . There is usually also a “CeaseFire()” and other sundry functions all publicly-addressable. These functions will turn the firing of the weapon on or off.

As for aiming, the Machine guns are aimed by rotating the barrel to line up with the target. (this script is up to you). The Mortars are aimed by rotating the MortarBarrelEnd towards the target and adjusting the initial force for the round (for the distance).

The Machine gunners hands (and the ammo belt) should match the rotation of the barrel as long as it rotates within the limits discussed above. You may opt to have an animator component move the barrel around the fire-zone while the weapons sprays a volley of fire. Or a combination of script/animator to mimic a gunner moving the gun about to spray a large area with bullets. Toggling the fire/ceasefire functions with animator events can give a realistic controlled burst.

Check the public functions in the crew weapon control script to see what options are available for that weapon.

It is up to you to create the fire control scripts for these weapons. You may consider using a Fire-control officer to orchestrate several mortars to fire on the same target. Again, target acquisition and aiming is up to you.

Mortars -

You are welcome to drag a **Demo** mortar crew (or any demo crew for that matter) into your game environment and watch what it does once you press play. You will be treated with a glorious series of animations, rounds launching and explosions erupting. No target is needed for this to happen. These assets will simply start firing when in play mode. Obviously, you will want to control these assets a bit more for your game BUT I wanted to show you the logic I had for creating them so you can understand how they tick.

Demo Mortar Operation - All mortar crews have a similar way of operating. The mortar crews are divided into 4 parts:

1 - Mortar Squad Leader
2 - Mortar Loaders and Ammo bearers
3 - The Mortar
4 - Ammo boxes and other meshes to help fill out the look of the crew

In the mortar demo prefabs I have set up the code so the mortar Squad Leader controls the other assets. The Squad Leader makes the decision to fire the mortar and communicates with the rest of the team to fire. The animation for the Squad leader has events that prompt the squad leader to make a decision to fire or not. Once the Squad Leader signals the rest of the team to fire all crew members switch from their “idle” animation to the “fire” animation. These animations are timed to look like the crew is working together. The Mortar Loader signals the actual mortar to fire via an event placed in his animation. Once the mortar is fired a mortar round is instantiated at the muzzle end of the mortar and flies towards the target. Once this round hits anything with a collider it tests to see if the object it collided with has the “terrain” tag. If it does then the mortar instantiates the explosion prefab with the crater and then the mortar round is destroyed. Otherwise the other explosion prefab is used without the crater.

Aiming the Mortar - I have not included any animation to aim the mortar. This is due to the complexity of having such actions and the difficulty involved in coding all of those behaviours - all for something that would have very little impact for the player. Here are my suggestions for how to aim the mortar:

Mortar Aiming Option 1 - simply adjust the azimuth (Y - rotation in world space) of the mortar barrel end. Since the mortars are pointed up at an extreme angle it is difficult to say just what it

is aimed at in the first place. Nothing changes for the crew or the model. Only the mortar barrel end is changed. This would be ideal for most aiming situations. That said, It will be more apparent if the model is firing at something over 60 degrees away from its forward direction that anything is amiss. If the mortar needs to change its forward direction by more than 50-80 degrees then try option 2.

Mortar Aiming Option 2 - Grab the whole mortar transform and rotate it, crew and all. You can try to do this slowly over time and hope the player doesn't notice or make it happen all at once so the mortar aimpoint is updated and the player gets that the mortar is pointing in the desired direction. I don't think the player would gasp at the sight of the mortar rotating quickly to the new direction, especially if it didn't affect the firing rate. Keep the momentum of the gameplay going without interrupting things to rotate a mortar.

Adjusting the distance the mortar fires is done by adjusting the force given to the mortar round when it fires. There is no elevation rotation. That said you COULD adjust the rotation of the mortar's barrel end and the player would probably not notice.

To RECAP - the mortar crew has two modes of operation, Idle and Firing. Once the squad leader gives the command to fire all the crew members switch to their firing animations. Pretty simple but it looks complex only because the animations are interwoven with each other.

Mortar impacts and craters

I created a shallow crater mesh to indicate where the mortar round impacted. You may or may not want to use this. The crater mesh only looks good if the impact is at ground level and the GROUND is LEVEL. If it is on a hillside or some other sloped terrain feature the crater looks silly since the edges would be visible. The mortar rounds I created check to see if the object the mortar hit is tagged with the "Terrain" tag. If it is then they will create the crater. You may want to create your own mortar rounds and tags to decide what will make a crater and what will not.

Machine-gun Operation

The machine gun crews operate in a similar way to the mortar crews BUT the main driver for the machine guns is the actual machine-gun barrel. The barrel is rotated via animation or script to point towards the enemy and the machine gun operator animation is updated to make the hand positions match up with the hand-grips on the gun (by using a blend tree). This gives the illusion that the gunner is moving the gun around. You can see in the script how this is done for more detail. The ammo belt is operated in a similar fashion. As the machine gun is rotated the ammo belt is flexed to make it match up with the gun's rotation.

Once the gun fires it turns on all 3 particle effects and toggles the loader's animation so the loader "feeds" the ammo to the gun. The gun also toggles the script for the texture on the ammo to make it animate - thus the rounds appear to move to the barrel when the gun is firing.

Incorporating These assets into your Battle Scene =

If you select and drag a PLAIN Heavy weapon crew into your scene you will have a group of soldiers manning a weapon and they will remain in their idle state until their public function FIRE(); is called. On the machine guns this function is the Machinegun_Control script usually attached to the root object in the prefab. On mortars this script is the Squad_Leader_Mortar script attached to the Squad Leader Character. So call this function to fire the weapon. Enjoy!