

GAMETEMPLATE

EBAL  
STUDIOS



**MINES  
EVADER**

## Contents

• Introduction and Features	2
• Folders	4
• Getting Started	5
• Updates and Notes	6
• Demo Scenes	9
• Package Components (Index)	11
• FAQ	19
• Terms of Use	20

# Introduction

Mines Evader is a Unity engine template, in which the goal of the player is to detonate as many mines as possible; by coming close to them then escaping away before they explode. Its behaviours are physics based, uses a minimum number of scripts. And is compatible with both mobile and desktop applications.

This package can be modified quickly to create your own simple game, or to learn from it and use some of its components in your own game.

We originally created Mines Evader when we were testing out the mines feature for our shoot 'em up creator package, but then had so much fun playing with it that we thought of sharing it for free as a standalone product, and as a promotion for our main package.

Our shoot' em up creator, Shmup Baby can be found here:

<https://assetstore.unity.com/packages/templates/systems/shmup-baby-128937>

## Features

- Physics based proximity mines system.
- Detailed mines controls
- Mobile and desktop controls support.
- Uses Unity event system.
- Camera tracks player and auto clamps game region.
- Bullet burst based on particle system.
- Warning flash color.
- Easily customizable background, player and enemies.
- Royalty free audio to use in your commercial game.
- 4 Ready to use and modify demo scenes.
- Artwork included.
- Animated effects and explosions.
- Minimal demo and scripts added for you to see if you are very new to Unity

# Folders

- **Art**  
Contains the animations, materials, textures, sprites and meshes. Basically all the source files for the art used in this pack
- **Audio**  
Holds the audio used in this package, one soundtrack and six sound effects.
- **Documentation**
- **Prefabs**  
The prefabs are placed in sub-folders as well to make them more organized. These folders contain everything you need to create a game level and the prepared prefabs for the artwork and 3D models.
- **Scenes**  
The demo scenes are located here.
- **Scripts**  
Holds all C# scripts used in this package with an additional minimal folder included. This minimal folder contains very basic scripts for the minimal scene to make it easier to learn this pack for all those who are very new to Unity.

# Getting Started

After importing the package, you are ready to start playing and customizing the existing demo scenes or to create your own game from scratch. Simply adjusting the settings and properties of the different game scene elements, and seeing how the events are connected; will help you understand how this package functions. The scripts is well commented and you can just view them if you want to understand them better and use them inside your own game.

We have also added a minimal version which should be straightforward to understand with very basic scripts and less features which makes it very easy to follow.

You can view the video tutorials related for this pack at this Youtube Playlist:

<https://www.youtube.com/playlist?list=PLh5hgaVhes-i0tbuRE4usAewmUUmbSBXJ>

This is a free pack, which means we get over a 100 downloads per month, we most certainly won't be able to help you through your Unity learning process other than providing this pack for free as we need to dedicate our resources for our other packs and for the clients who have bought our products, but if you have tips on how to improve this pack you can always check this forum thread:

<https://forum.unity.com/threads/mines-evader-free-template.531522/>

# Updates & Notes

We have just completely rebuilt Mines Evader, we first released this pack as an asset store test before releasing **Shmup Baby**, and after revisiting it lately we felt it really needed an update and to improve it a bit.

The recreation changed few things if you happened to try the first release. but the essence of the pack is still the same and we believe the code now is better commented and is easier to understand and have more null reference safeguards.

It's rather unlikely that we would release future updates for Mines Evader, the process of creating asset store packs is rather time consuming and we would prefer to give our paying clients all the support they need and to focus our efforts on making new packs but we hope that this free game might help few developers who are just starting with Unity.

## Art Assets

If you want to build your own modular spaceships, you might consider downloading the free Star Sparrow asset: <http://u3d.as/BuK> **But If you want a MUCH bigger variety of spaceships you can check our Ultimate Spaceships Creator over here: <http://u3d.as/J0F>** and for a bigger collection of mines, you can check this asset: <http://u3d.as/Vqb> Thank you for downloading Mines Evader, we wish you all the best of luck in your game making!

## Notes:

- The size of the collider affects the speed and rotation behaviour of the player.
- The mine Flashing script needs to be referenced to the visual, if you add a Flashing script to a mine make sure you also connect the visual game object of that mine (i.e. the mine art object) to the My Material slot.
- Due to Unity's event system, if you make prefabs such as for the player or mine. Make sure to reconnect any associated events and action all over again.
- You will need to re-connect the mobile controls if you make a new player
- If you build the game and encounter any strange camera behaviour, such as the camera going out of the game field bounds, just double check your camera and game field sizes or try playing the scene again and rebuilding one more time, it might sort itself out.

# Demo Scenes

There are 4 demo scenes included in this package and are all located in the scenes folder. Please note that all of these demos have been created quickly and we are certain you can come up with better examples which are well balanced in terms of game design, we just wanted to show the pack capability, where you take it from here is up to you.

- **Spaceship**

This demo scene uses the star sparrow spaceship as a player with particle thruster effects, the spaceship animator uses the afterburner script for animating the engine particles of the spaceship. Which basically just turns the engine particles of the direction you are turning to.

- **Space Insect**

Yes! Insects cannot fly in space. But this is a special robotic vehicle that uses one of a kind hyper electromagnetic void propulsion system which manifests in what appears as wings flapping around! Anyway, we were just having fun with it and we kept it, exactly like the spaceship scene but with a different player animation.

- **2D**

This one uses 2D illustrations, very similar to the other scenes, but perhaps the main difference to note is that you make you don't enable any mine rotations. Because the mines would rotate randomly on all 3 axis, which you can imagine, does not work well when using a 2D sprite. It also only uses the Track mine and no the chain mine.

- **Minimal**

This one is very interesting, it uses different scripts. A far simpler version of the original scripts and as the name suggest with minimal capabilities and features for the level. No explosions, sound, UI etc. but it's really easy to get up to speed with it.

# Package Components

## Index

Game Manager & UI	8
Mines Evader Camera	9
Spawner	10
Player	11
Player Anim	12
Afterburner	12
Track Mine	13
Chain Mine	15
Bullet Burst	16
Flash	17
Explosion	17
Stars	18
Repeater	18



# Game Manager & UI

The game manager is an essential component for this pack as it finds and stores the player, handles damage done to it, ends the round and controls the UI.

All UI elements need to be referenced to it.

The UI is very simple and you will find it in every scene in the pack (except the minimal one), you have the in game UI which holds the health bars, the score and hit effect.

The phone controls which are only active if you are building for mobile, and the game over screen with the high score. All these can be changed easily and you can see how the connections are made inside the game manager.

## Notes:

- You need to match the player health amount to the health bars you have in the UI.
- The "HoldButton" script is used with the phone controls UI and needs to have the player reference to it to function and make the player move, if you start a new scene make sure to connect it.

## Requires:

- **Audio source**
- **UI references**

## Fields:

- **Game Field:** Here you will need to determine how big the game field is, this will determine where the player, mines and camera can move and interact.
- **UI Reference:**
  - **In Game:**
    - **Health Bar:**
      - **Size:** How many health bars do you need, this has to match player health.
      - **Element:** You reference here the full health bars, and make sure that underneath them you have the empty versions so that once this is deactivated you see the empty versions.
    - **Hit Effect:** This effect is to indicate when the player is hit.
    - **Score:** You current score text reference.
  - **Game Over:**
    - **Game Over Canvas:** The entire game over canvas needs to be referenced here so that it's active once the round ends.
    - **Final Score:** The text for the final score.
    - **High Score:** The text for the high score.
- **Phone:** You need to reference the Phone UI here, this will activate it when you are building for mobile.

- **Hit Effect Duration:** For how long the Hit Effect image you referenced above will last when the player is hit.

## Mines Evader Camera

The camera is a simple component that follow the player and stays within the bounds of the game manager.

There are public fields for this camera as it simply uses the player position to follow it, finds the camera size and game field so it can find its own field.

If you want you can simply choose to make a static camera instead of following the player and delete this one.

### Requires:

- **Game Manager's Game Field**
- **Player**

# Spawner

The spawner is an essential component for spawning mines, the level of complexity you find in the script has to do with the ability of the spawner, to spawn different waves made up of different mines with an increment rate. A much easier to understand script would be the one in the minimal version. This spawner will produce mines infinitely and you have the option of having a limit on the total number of mines on screen and to spawn these mines away from the player.

## Requires:

- **Game Manager's Game Field**
- **Player**
- **Mines References**

## Fields:

- **Waves:**
  - **Size:** The number of waves that will be spawned on every level
  - **Element:**
    - **Mines:**
      - **Size:** The mines that this wave will be able to select from randomly for spawning, for example you keep this size at one, you will have only one type of mine spawning for this wave, but if you use 3, then this wave will spawn these 3 mines and pick from them randomly.
      - **Element:** You will link here the mines to spawn from.
    - **Spawn Rate:** How many mines are spawned per second for this wave.
    - **Wave Time:** For how long will this wave last
    - **Time For Next Wave:** The time in between this wave and the next wave to give the player some breathing time to finish off the first wave.
- **Rate Increment:** The increase in the spawn rate with every new level, to make the game harder as time progresses.
- 
- **Distance From Player:** How far away from the player will the mines spawn, so that you don't see the mines popping up, make sure this number is not too big or there will be no place for the mines to spawn from.
- 
- **Max Number of Mines:** The maximum number of mines on screen at one time, this will actually skip spawning a mine if the total number has been reached.

# Player

The player is of course a central component, the player is needed not only to play the game, but the game manager stores it, and the camera and mines are always looking up its position to know how they can move.

The script is straightforward as it basically moves the player through the rigid body but some complexity exists due to the cool down time and the mobile controls. If you are having any trouble understanding it you can always look up the minimal version of the script

## Requires:

- **Rigidbody 2D**
- **Collider 2D**
- **Game Manager's Game Field**

## Fields:

- **On Destroy Event:** this event is important so that you can end the round once the player is destroyed, there are no player lives in this pack, only health bar. Once you are destroyed the game should end. To do so you should connect the game manager to this event and then pick Game Manager, Round End as an action.  
You can also connect to it the player explosion to enable it once the player is destroyed.
- **Speed:** The movement speed of the player
- **Rotation Speed:** The rotation speed, this needs to change if you change the size of the collider.
- **Health:** The health value of the player, this must match the number of health bars you have in the UI
- **Cool Down Time:** Immunity time after each hit the player gets.
- **Get Hit Clip:** The sound the player makes when it's hit

## Player Anim

We have added a simple script that starts an animation state once the player turns left or turns right, you can see the space insect player or the spaceship player to see how it works, but it's relatively straight forward, this script is added as an extra component for the player script.

This script is activated from inside the player component if it's turning left or right and switches the animation state accordingly. For the space insect we just have a simple wing flapping animation that switches from left to right and for the spaceship we made a quick afterburner script to turn on and off the particles thrusters.

### Requires:

- **Player Component**
- **Collider 2D**
- **Game Manager's Game Field**

### Fields:

- **Animator Reference:** Reference here the object with the animator attached, make sure your animator has a "TurningLeft" state and a "TurningRight" state with the exact spelling, (try to duplicate the existing examples).

## Afterburner

The afterburner is a special case script, only implemented for the spaceship, we needed it to disable/enable the emission of particles in the thrusters for the spaceship when it's turning. It must be located inside an animation controller as this script is a `StateMachineBehaviour` and not a `MonoBehaviour`. It must also find the thruster with the exact name or else it will start throwing our errors.

This script only enables and disables the thrusters particles emission based on the player turning direction.

### Requires:

- **Animation State**
- **To Find a Particle System**

### Fields:

- **Thruster Name:** This is the particle system you wish to turn on and off based on Player Rotation, it must match exactly or it will throw out hash ids errors.

## Track Mine

The track mine is the basic to start with, actually the second mine (Chain Mine) isn't exactly very different, but we thought that doing the mines the way we did will show you a good example of inheritance and different coding practices.

This track mine finds the player and moves towards it, changes speed as it gets closer, will start a detonation process if the player gets within a certain distance and can explode and damage the player and if you collide with it you can get damaged as well.

We have also added an optional bullet burst effect and a flashing effect when detonation starts.

Most of the content of this mine script is actually located inside MineBase, this way you could easily have a base to start with a new type of mine. The code for the track and chain mines can still be further optimized and improved but just wanted to leave something for you and for us to get back to working on our other assets.

If you find any trouble understanding inheritance, you can look up the minimal script to see a simple version of this script

### Requires:

- **MineBase script to exist (You cannot delete MineBase.cs)**
- **Player**
- **Game Manager's Field**

### Fields:

- **Damage Player Event:** You need this event to be able to inflict damage on the player and set the damage value, Connect the game manager to it and choose Damage Player function.
- **On Destroy Event:** You need this event for 2 reasons, to add an explosion effect and to also add score through the game manager's add score.
- **Start Detonation Event:** You probably could only use this event to start a flash effect, but it's up to you, you could also add any other action to it.
- **Start Speed:** The start speed of the mine, this is the idle speed. The speed the mine will move towards the player with after it spawns even if the player is outside the detection radius.
- **Detection Radius:** When the player is detected within this radius the mine will start changing its speed according to the distance of the player to the mine.
- **Detonation Radius:** When the player enters the radius, a detonation process will start which will explode the mine after the detonation timer ends.
- **Explosion Radius:** The radius in which the player is damaged if it's located inside it
- **Detonation Timer:** The time between the player entering the detonation radius and the mine exploding.

- **Allow Collision Damage:** If ticked, the player will be damaged if it collides with the mine.
- **Random Start 3D Rotation:** Gives the mine a random start rotation, make sure to untick if you are using a 2D sprite.
- **Detonation Clip:** the sound made when the mine starts the detonation process.
- **Rotation Speed:** This rotation speed in the track mine is also dependent on the mine speed, as the mine changes speed when it approaches the player, this will also change to give the mine a dynamic effect.
- **End Speed:** The speed of the mine when the player distance to the mine is zero.

# Chain Mine

The chain mine is exactly like the track mine with few differences, it also inherits from MineBase. The main difference being that it applies an impulse force instead of lerping the speed, and we put a special case if it hits something other than the player to change that force value. Again we just used different mine types to show the inheritance function more than actual functionality.

Note: Of course many of the fields are the same because it inherits most of its values from MineBase

## Requires:

- **MineBase script to exist (You cannot delete MineBase.cs)**
- **Player**
- **Game Manager's Field**

## Fields:

- **Damage Player Event:** You need this event to be able to inflict damage on the player and set the damage value, Connect the game manager to it and choose Damage Player function.
- **On Destroy Event:** You need this event for 2 reasons, to add an explosion effect and to also add score through the game manger's add score.
- **Start Detonation Event:** You probably could only use this event to start a flash effect, but it's up to you, you could also add any other action to it.
- **Start Speed:** The start speed of the mine, this is the idle speed. The speed the mine will move towards the player with after it spawns even if the player is outside the detection radius.
- **Detection Radius:** When the player is detected within this radius the mine will start changing its speed according to the distance of the player to the mine.
- **Detonation Radius:** When the player enters the radius, a detonation process will start which will explode the mine after the detonation timer ends.
- **Explosion Radius:** The radius in which the player is damaged if it's located inside it
- **Detonation Timer:** The time between the player entering the detonation radius and the mine exploding.
- **Allow Collision Damage:** If ticked, the player will be damaged if it collides with the mine.
- **Random Start 3D Rotation:** Gives the mine a random sart rotation, make sure to untick if you are using a 2D sprite.
- **Detonation Clip:** the sound made when the mine starts the detonation process.
- **Speed Threshold:** The threshold of the speed that prevents the mine from exceeding a certain speed.
- **Impulse Force:** The force used to push the mine if it detects the player inside the detection radius.
- **Chain Force:** The force used to change the impulse force if the mine happens to hit an object other than the player such as another mine or a background object.
- **Enable Mine Rotation:** If enabled the mine will use the impulse force to apply torque to the mine.



## Bullet Burst

You have the option to emit bullets when the mine is destroyed or whenever any certain event occurs. This script might appear huge, but most of it is just there to help you setup the particle system and change its settings. Very few parameters are actually actively controlling the particle system.

The way Unity's shuriken particle system is created, forces us to find and references the modules in the way you see in the script, you could see more tutorials about the subject, but just imagine that there are many modules each with their own parameters and this is the way to access them. A lot of work has been done so that you can simply add a particle system, a bullet burst script and not worry about the settings.

### Requires:

- **A particle system**
- **Player**
- **Game Manager's Field**

### Fields:

- **Bullets Hit Player Event:** This event is important to make the bullets actually damage the player, just like the mine, you will need to connect the game manager to it and pick damage player with the damage value.
- **Bullets Material:** The material you want the bullets to have, usually a transparent sprite material. We have included couple of suitable bullet materials.
- **Distance:** How far will the bullets travel before being destroyed.
- **Speed:** The speed of the bullets.
- **Bullets Number:** How many radial bullets emitted, these bullets are distributed radially.
- **Emit Distance:** This distance makes the bullets emit from a distance away from the source (i.e. instead of emitting from a dot, they will emit from a circle, show bullets path explains it better).
- **Collider Size:** Very important to set a size for this, this determines the collision size of each bullet.
- **Fire Clip:** the sound made when the bullets are fired.
- **Show Bullets Path:** Shows you the direction and path of the bullets since you can't see them in particles inside the editor as the script only modifies them when the game is run.

## Flash

This flash effect is a quick and simple effect to give your mine a flash effect when it starts the detonating process to give you a visual feedback if the mine will explode soon, it needs to reference a material; so it must be applied only to objects which has a renderer and a material.

To apply this effect, you usually have it added to the visual of the mine, then on the mine On Detonation event you add the visual to the event then pick start flash function.

### Requires:

- **Renderer (Material)**
- **An event to start the flash effect**

### Fields:

- **Total Flash Time:** The entire duration of the flash effect process.
- **Time Between Flashes:** Each flash duration (flash frequency).
- **Flash Color:** The color that will be overlaid on top of the material.

## Explosion

The explosion script is needed to simply postpone playing the explosion animation and then to destroy the explosion object after it has finished, it also needs to unparent the explosion from the destroyed object, otherwise your explosion will be destroyed before it gets the chance to play itself. It also plays the explosion sound.

The only field you have the explosion clip, the explosion itself and the animator need to be already present in the object you are adding the explosion to

Usually you would add this script to an animated explosion sprite, link this explosion to a mine or player object, then reference the explosion inside the mine or player OnDestroy Event and choose the explode function to start when the OnDestroy event occurs.

### Requires:

- **Sprite Renderer**
- **Animator**
- **An event to start the explosion (typically an OnDestroy event)**

### Fields:

- **Total Flash Time:** The entire duration of the flash effect process.
- **Time Between Flashes:** Each flash duration (flash frequency).
- **Flash Color:** The color that will be overlaid on top of the material.

## Stars

The stars script is a script to generate stars particles automatically, the procedure of referencing a particle system is of course the same as the one we used in the bullet burst script, but in this script we have less parameters to control.

If you are intimidated by Unity's shuriken particle system it might be better to start with this script rather than the bullet burst one.

### Requires:

- **A Particle System**

### Fields:

- **Stars Size:** The size of each star.
- **Stars Lifetime:** For how long each star exists.
- **Stars Rate:** The rate of spawning stars.
- **Max Stars Number:** The maximum number of stars on screen.
- **Random Stars Lifetime:** If ticked it will use the stars lifetime as a maximum value and use a random number between zero and stars lifetime.
- **Random Stars Size:** If ticked, it will use the stars size as a maximum value and use a random number between zero and stars size.
- **Stars Material:** The sprite material used for the stars.

## Repeater

The repeater script is a simple script to repeat a linear transform movement, it's basically just a Mathf repeat function. We used it to have nebulas or clouds on top of the background and to keep them looping through the game. This is by far the easiest script to understand in this pack.

### Requires:

- **Nothing**

### Fields:

- **Repeat Distance:** The distance that will determine when to repeat the movement from the start.
- **Speed:** The speed of movement.

# FAQ

- I've changed and customized this package, and I am planning to build a game I am going to sell, monetise or use ads in. Is it Okay?  
Yes.
- Can I use the music and sound effects you have included with this package for this other game I am making?  
Yes.
- Can I use some of your awesome scripts in another game I am making?  
Yes.
- Do I have to credit you in my game?  
No, but it would be nice if you did!
- I dragged and dropped stuff from the prefabs into the scene and it's not working?  
Did you reconnect all the events and references? Even those hidden inside the mine visual for flashing effect for example. Is UI connected to the game manager, the explosions to the "On Destroy Event" etc..
- The mines explode or make a bullet burst but the player is not damaged?  
Is the damage value of that mine script "On Hitting Player" zero?
- This asset is crashing and it's not working, it's all your fault! Fix it. Now!  
We are absolutely sorry to hear that, we have tried our best to make this package as solid as possible and we apologize if we caused you any inconvenience. if you haven't already; please check the video tutorials and this documentation thoroughly and see if there is anything you've done differently than what we demonstrated, also try reimporting the package, and tracking your steps, one by one. Use debug logs and see what's happening. This package is relatively small and we are certain if you retrace your steps and have some time you could rebuild a better version of it from scratch. Because this is a free package we really can't help all who download it. The volume of downloads is huge and we get some E-mail of users who are adding features to it and then ask us to make it work, if you have any programming questions please do check the forums at Unity, they are great for getting help and will be able to help you much faster than what we can. Thanks a lot and we hope you understand our situation.
- I absolutely love this asset! how can I help or support you guys?  
Thank you so much! You are a true gem! If you really like our work, you might want to check the rest of our package on the asset store and see if there is anything you need to buy!  
<https://assetstore.unity.com/publishers/24304>

# Terms of Use

This package can be used in any commercial or non commercial video game, you may edit it and release your own game or use one of its components in any of your applications.

You are allowed thus to use this script or any of its components in your own video game or creation. This also includes the artwork, music and sound effects included with this package.

You are not allowed to re-distribute the package in its entirety or any of its elements or assets in any shape or form. You are not allowed to use the scripts, artwork or music in an asset or pack you plan to publish on the Unity asset store or on any other website and you may not give them away for free or for a price.

For example, you may modify this script and build a game and release it on the app store or the play store. You may use the spaceships, mine models, or music in your game. But you cannot give away the models, scripts or package for free or for a price on the asset store, or any other website.



**WWW.EBALSTUDIOS.COM**

**<https://assetstore.unity.com/publishers/24304>**

**Huge Thanks For The Music & SFX Provided By  
WWW.KLEMELKSTUDIOS.COM**