

# Simple 2D Enemy KI

## **Please Note (important):**

The Simple 2D Enemy KI use two custom layers:

Ground

Enemy

With the layer 'Ground' the scripts are checking isGrounded true or false. The layer 'Enemy' is used to ignore layer collision between enemies.

I added the class '/Simple 2D Enemy KI/Editor/Tags.cs' to create the both layers automatically on project startup. The gameobjects in the scenes contains the component 'Generate Level Layer and Tags'. You don't need it in your own project. This is only for setup the scenes after importing the asset in your new project.

## **How to use**

### **EnemyAutoWalk (Scene 1)**

Create the playfield as you want. The field edges needs the tag 'FieldEdge' and the floors need the layer 'Ground'.

Define EnemyStatePoints in your scene: Create an empty game object and add the script 'EnemyStatePointController' as a component. You can configure what the enemy should do, when the enemy collide with it. There are four scopes:

- Jump up
- Jump forward
- Move forward
- Change direction

Okay, now copy the 'EnemyAutoWalk'-Prefab in your scene and setup the 'EnemyMoveAutoScript':

Player following	If you have activated the (optional) 'Player Following', the enemy jumps forward, if the player is above the enemy. Or the enemy move forward if the player is below the enemy
Enemy Auto Period Jump	The enemy makes random jumps upwards
Time Delay Min	Random auto jump up Min-Value
Time Delay Max	Random auto jump up Max-Value
Grounded Check Radius	If your enemy has other dimensions as in my example, you must define another value
Max Speed	Speed of the enemy
Jump Force Up	How hard should the enemy jumps up
Jump Force Up Forward	Value for forward-jump on a EnemyStatePoint
Jump Though Walls	Deactivate the enemy jump trigger on starting jumping up. Re-activate the enemy jump collider again, if the enemy has arrived the highest jump-coordinates.

If you want to change the enemy animations, make sure that the parameters and transitions in the animator are the same as in my example.

Please note: If you define a top frame on your field and the option 'Jump Through Walls' is enabled, you must add the script 'Field Top Fix' as a component. If the player or the enemy collide with the top frame, the game objects trigger collider will be disabled on OnTriggerEnter and TriggerStay and would be enabled again on TriggerExit. So the player and the enemy could not leave the gamefield.

Used scripts:

- EnemyMoveAutoScript
- EnemyStatePointController
- FieldTopFix

## **Enemy Player-Following (Scene 2)**

Just enable the Player-Following on the  
'EnemyMoveAutoScript'-Component.

Used scripts:

- EnemyMoveAutoScript
- EnemyStatePointController
- FieldTopFix
- EnemyHitAndDieController

**UPDATE: I updated this scene with**

- **Player shoot enemy**
- **Enemy die-animation**
- **Enemy and Player-Respawn**

To use enemy hit and die just add the EnemyHitAndDieController to the enemy and set a gameobject as respant-point in the scene. Add this respawn point to the EnemyHitAndDieController-Component.

Use 'Left CTRL' to shoot bullets.

After player death the player will be destroyed and the player prefab will be instantiate completly new at the position of the PlayerRespawnPoint. To change the respawn-position, change the x and y value at the PlayerRespawnPoint-Prefab.

### **Enemy Auto-Shoot (Scene 3)**

Just add the 'EnemyAutoShootController' to your enemy as a component and add the 'EnemyShot'-Prefab.

If the opponent sees the player, the enemy shoots automatically after some delayed time.

Used scripts:

- EnemyMoveAutoScript
- EnemyStatePointController
- FieldTopFix
- EnemyAutoShootController

### **Enemy-Bouncing (Scene 4)**

Create the playfield as you like. The field edges needs the tag 'FieldEdge' and the floors need the layer 'Ground'. Copy the Enemy-Bouncing Prefab in your scene and setup the speed variable.

The child gameobject of the Bouncing-Enemy contains the 'Bounce Trigger'-Script. This script define the tags and layers with which the Bouncing-Enemy can collide.

Used scripts:

- BounceTrigger
- EnemyBounceWalkScript

### **Enemy Waypoint-Walk (Scene 5)**

Copy the EnemyFlying Prefab in your scene and setup the speed variable.

Subsequently define empty game objects as waypoints in your scene and add it to the list 'Waypoint Positions' of your EnemyFlying Prefab.

If you want to change the animation, select your EnemyFlying game object and open the animation 'Window', then replace the sprites with your own.

Used scripts:

- EnemyWaypointWalker

### **Enemy Shoot horizontal and vertical (Scene 6)**

I used the Waypoint-Walk-Enemy for this example. Add the 'EnemyHorizontalVerticalAutoShoot'-Script to your game object and make your setup:

Shot	The 'EnemyShot'-Prefab
Shot After Time Min	The Enemy deploy random shoots. This is the Min-Time-Value
Shot After Time Max	Max-Time-Value

Used scripts:

- EnemyHorizontalVerticalAutoShoot

### **Contact Information:**

E-Mail: [support@droidspirit.com](mailto:support@droidspirit.com)  
Website: <http://www.droidspirit.com>