

Anomaly zone

Anomaly zone is a space restrictor which covers certain area and contains anomalies. Artifact is jumping inside, following specified path. Upon entrance, actor is instantly hit by force specified in zone settings, special post process effect is displayed and sound is muted (vacuum-like effect ☺).

If actor visits anomaly zone for a first time, infoportion is given (if specified in anomaly settings), so we can use it for some stats or achievements. Each anomaly zone is updated in both: on-line and off-line states, so no matter where actor is, anomaly zone is updated (to calculate probability to spawn artifacts, spawn artifacts during blowout, etc).

To spawn anomaly zone, select `spawns\lost_alpha\anomaly_zone` and give it some **unique** name. This name is then needed for setting zone parameters. Open file `config\lost_alpha\anom_zone\anom_fields.ltx` and create new section with name you gave to spawned anomaly zone and specify parameters.

Anomaly zone parameters:

- `spawn_af_on_zone_spawn` – whether spawn artifact when anomaly zone is being spawned; if this field is set to `true`, then artifact will be spawned when anomaly zone is spawned, that is – when new game is created (so this is a one-time-only spawn); you can define artifact type in field `af_on_zone_spawn`; default value is `false`, so no artifacts will be spawned on anomaly zone spawn
- `af_on_zone_spawn` – section name of artifact to spawn when anomaly zone is spawned; this field is utilized only if `spawn_af_on_zone_spawn` is set to `true`; default value is `af_medusa`, that is - jellyfish
- `idle_time` – time to spawn artifacts (in game minutes); after every `idle_time` minutes script will test conditions to spawn artifacts and depending on result artifacts will be spawned or not; default value is `10` minutes
- `cond` – condition list to spawn artifacts; each condition has to end with probability value; you can use value `0` (min) to disable artifact spawning or `100` (max) which means that after every `idle_time` minutes artifact will be spawned (artifact can be spawned only if there's no artifact already in zone); default value is `nil` which evaluates to `0`, that is – artifacts won't be spawned at all
- `on_blowout` – whether spawn artifacts after blowout or not; available values are: `always` – spawn artifact after each blowout, `never` – do not spawn artifacts after blowout or `use_cond`

- use condition list specified in [cond](#) field to calculate whether anomaly zone can spawn artifacts after blowout; default value is [never](#)

- [hit_type](#) – type of hit which will be used to harm actor inside anomaly zone; available values are: [chemical_burn](#), [burn](#), [radiation](#), [shock](#), [strike](#), [telepathic](#), [wound](#); default value is [radiation](#)

- [postprocess](#) – post process effect which will be displayed when actor enters anomaly zone; it can be either existing *.ppe file (good vanilla post process effects are: [dead_zone.ppe](#), [psy_antenna.ppe](#), [psy_antenna_indoor.ppe](#), [radiation.ppe](#)) or custom scripted post process (read below); if no post process is given (default value), then anomaly zone won't display any post process effect when actor is inside zone

- [npc_accessible](#) – Boolean value which determines whether NPCs can search this anomaly zone for artifacts; thanks to this we can block some anomaly zones so NPCs won't enter them; default value is [false](#), which means that NPC won't enter this anomaly zone

- [actor_first_visit](#) – infoportion which will be given when actor visit this anomaly zone for a first time; this can be used for stat or achievements purposes; by default, if no infoportion is given, then script won't give any infoportions

- [artefacts](#) – comma separated list with artifact section names and probability values; this defines what types of artifacts can be spawned by anomaly zone based on given probability values; this field is **necessary**, after all this is main purpose of anomaly zone – to spawn artifacts, so if you don't want this anomaly zone to spawn artifacts, you still have to specify at least one artifact here and then you can block spawning with [cond](#) field

- [af_paths](#) – artifact jumping path, that is waypoints names; you can specify more then one waypoint name (separated with comma) and then script will randomly pick one of paths (gives more randomness); this field is **necessary** – same reasons as with [artefacts](#) field

- [af_jump_speed](#) – speed coefficient for artifact which determines how fast artifact will jump; default value is [400](#)

- **spawn_anomalies** – Boolean value which determines whether anomaly zone should spawn anomalies or not; in other words, if this field is set to **true** then anomalies will be spawned via script; default value is **false** and to be honest, using level editor you have more options to spawn anomalies (for instance adjusting anomalies shape, offset and radius)
- **anom_pos** – name of waypoint which holds spawn positions for anomalies; this is usable only if you set **spawn_anomalies** field to **true**
- **anoms** – comma separated list with anomaly names and quantities; this determines what anomalies will be spawned; utilized only if **spawn_anomalies** field is set to **true**

Example – if you spawned anomaly zone called `test_esc_anom_zone`:

```
[test_esc_anom_zone]
spawn_af_on_zone_spawn = true
af_on_zone_spawn = af_cristall_flower
spawn_anomalies = true
anom_pos = esc_anom_zone_test_anompos
idle_time = 5
cond = 100
on_blowout = always
npc_accessible = true
af_jump_speed = 600
artefacts = af_medusa, 6, af_cristall_flower, 5
af_paths = esc_anom_zone_test_afpath
postprocess = fallout_light
hit_type = shock
actor_first_visit = barin_anim_test
```

Scripted post process effects

You can use already defined effects or add new ones (in that case add effect name to **all** section) – check file `config\lost_alpha\anom_zone\anom_zone_pp.ltx`. Each effect is defined as:

```
[effector_name]
color_gray = R, G, B [0.0-0.5, 0.0-0.5, 0.0-0.5]
color_base = R, G, B [0.0-1.0, 0.0-1.0, 0.0-1.0]
color_add = R, G, B [0.0-1.0, 0.0-1.0, 0.0-1.0]
noise = grain, intensity, fps [0.0-1.0, 0.0-1.0, 1-100]
```

```
dual = horizontal, vertical [0.0-1.0, 0.0-1.0]
gray = gray [0.0-1.0]
blur = blur [0.0-1.0]
```

You can use [ui_pp_test.script](#) to create new effects:



Attaching anomalies to anomaly zone – only for level editor

If you are spawning anomalies in level editor, then first spawn anomaly zone, give it some shape, radius, unique name and edit zone parameters (as described above). Then spawn anomalies, for each anomaly edit its [custom_data](#) field and add anomaly zone name. For instance, if you spawned anomaly zone called `esc_anom_zone` then every anomaly which belongs to this anomaly zone should have this custom data:

```
[anom_zone]
```

```
belongs_to = esc_anom_zone
```

This is required for script which needs to *know* what anomalies belongs to anomaly zone (because script needs to change anomaly restrictor type, so NPCs can evade those anomalies). If you decided to use script to spawn anomalies (field [spawn_anomalies](#) in anomaly zone settings) then script will automatically adjust [custom_data](#) field for those anomalies.