

# infected\_control 插件说明

## 1. 插件介绍

**infected\_control** 是针对《求生之路2》对抗模式的 SourceMod 插件，用于全面接管特感（特殊感染者）的刷新与传送逻辑。它关闭游戏原有导演对特感的刷新控制，由插件根据自定义算法生成特感<sup>1</sup>。插件的主要功能包括：

- **自定义特感刷新**：利用导航网格（Nav）计算合适的刷新位置，确保特感在幸存者视野外一定距离生成，避免不公平的刷新点。
- **刷新点智能搜索**：采用 fdxx 提供的思路，对地图 NavArea 进行预处理和分桶，将地图按进度划分为 0-100% 的桶区，特感将主要在靠近幸存者的位置附近刷新，并通过动态“扩圈”算法逐步扩大搜索范围寻找可用刷新点。
- **特感传送机制**：当特感长时间未与幸存者交战且处于幸存者视野外时，插件可将其传送到幸存者前方，以保持游戏压力（可配置启用）<sup>2</sup>。这有效防止特感掉队或卡在远处影响游戏节奏。
- **动态评分系统**：插件为每个潜在刷新点计算一个综合评分（距离、高度、流程、分散度四方面），挑选最优位置生成特感，从而提高进攻效果和公平性。
- **可配置参数丰富**：提供众多控制台变量（ConVar）供服务器管理者调整，如特感刷新距离、种类、数量上限、刷新间隔、AI 逻辑开关等，方便根据不同需求优化游戏体验。
- **特殊模式支持**：支持“全猎人模式”“全牛模式”等开关，一键切换特感类型刷新的模式；并兼容暂停插件、目标数量限制插件等，适用于竞技对抗环境。

总之，**infected\_control** 插件旨在让特感刷新更加智能、公平，增强对抗模式的策略性和观赏性，普通玩家无需懂代码即可感受到更合理的特感刷新节奏。

## 2. ConVar 设置说明

插件提供了大量控制参数（ConVars）来微调特感刷新行为。以下逐项列出各 ConVar 及其含义、默认值和推荐设置：

- **inf\_SpawnDistanceMin**（默认 250.0）<sup>3</sup>：特感刷新距离幸存者的最近距离下限，单位为游戏距离单位。值越大，特感刷新时离幸存者越远，越不容易被立即发现。一般保持默认即可，数值过小可能出现特感在幸存者眼前刷新的不公平情况，数值过大则特感过于远离不易形成进攻。推荐范围：250~500，根据地图开阔程度调整。
- **inf\_SpawnDistanceMax**（默认 1500.0）<sup>3</sup>：特感刷新距离幸存者的最远距离上限。插件在此距离之外不考虑刷新特感，以免特感刷出后赶路太久无法参与战斗。一般不需修改，极端大图可适当提高。对 Boomer、Spitter 这类支援型特感，插件内部会限定实际最大距离为 1200，以免刷得太过远<sup>4</sup><sup>5</sup>。
- **inf\_TeleportSi**（默认 1）<sup>6</sup>：是否开启特感超时传送功能。开启时，当特感在一定时间内未被幸存者看到且远离幸存者，会被瞬移到幸存者前方重新发动攻击。建议保持开启（1），能防止“掉队特感”影响游戏节奏。
- **inf\_TeleportCheckTime**（默认 5 秒）<sup>2</sup>：超时传送判定的时间阈值。当特感持续这么多秒未被任何幸存者看见，则认为其掉队，触发传送逻辑。可根据需要调节：降低该值（如 3 秒）会更频繁传送，增大则特感在后方逗留更久。默认 5 秒适中。

- **inf\_EnableSIoption** (默认63) <sup>7</sup>：特感启用类型的位掩码。用于控制允许生成哪些特感，掩码按位对应 Smoker、Boomer、Hunter、Spitter、Jockey、Charger 六种特感 <sup>8</sup> (bit0=Smoker, bit1=Boomer, ..., bit5=Charger)。默认63表示 (111111)\_2，六种特感全部允许生成。如需禁用某些特感，可按照位关闭相应位，例如想禁用 Spitter (bit3)，则设置值=63-8=55。一般保持默认除非有特殊玩法需求。
- **inf\_AllChargerMode** (默认0)：全牛模式开关。设为1时，插件强制只刷 Charger (牛)，其他特感不刷新 <sup>9</sup> <sup>10</sup>；设为0则正常随机。此模式适合练习躲牛等特殊玩法，默认关闭。**注意：**  
**inf\_AllHunterMode** 同时开启时，以全牛模式优先。
- **inf\_AllHunterMode** (默认0)：全猎人模式开关。设为1时只刷 Hunter (猎人)，其他不刷 <sup>9</sup>。默认关闭。两者 (全猎/全牛) 不建议同时开启，以避免冲突。
- **inf\_EnableAutoSpawnTime** (默认1)：是否开启自动增时 (动态刷新的时间调节)。开启时，插件会根据战局动态调整下一波特感的刷新间隔，例如在特感过早全灭或幸存者推进过快时，加快下一波特感的到来，以持续给幸存者压力 <sup>11</sup> <sup>12</sup>；在幸存者被压制时则适当延迟刷新。关闭时则每波特感严格按固定间隔刷新 (下文 SIInterval 值)。**推荐开启**，可使游戏节奏更灵活。
- **inf\_IgnoreIncappedSurvivorSight** (默认1)：传送检测是否忽略倒地/挂墙幸存者的视线。启用时，如果只有倒地或挂在墙边的幸存者能够看到特感，插件会将其视为“未被看到”，从而允许触发传送 <sup>13</sup>。这是因为倒地者无还击能力，不应阻止特感传送。一般保持开启，提高特感传送的有效性。
- **inf\_AddDamageToSmoker** (默认0)：对 Smoker 加强伤害开关 (单人游戏辅助)。开启(1)时，如果仅有单人玩家，Smoker在用舌头抓住玩家时将受到5倍伤害 <sup>14</sup>。这样单人也能更快打掉抓人的 Smoker，避免无人救援的僵局。多人游戏无此必要，默认关闭。单人战役或练习时可酌情开启。
- **l4d\_infected\_limit** (默认6)：一次最多同时存活的特感数量上限 <sup>15</sup>。默认6只特感，比原版对抗的4只略高，增加挑战性。可根据需求调整：4为经典对抗配置，6-8适合更高难度或多人插件 (例如8幸存者对8特感)。修改该值插件会同步调整游戏内部上限参数 <sup>16</sup> <sup>17</sup>。
- **versus\_special\_respawn\_interval** (默认16.0 秒)：特感刷新间隔秒数 <sup>15</sup>。即一波特感刷出后，下一波特感刷新的基础等待时间。默认16秒与原版相当，可调低以加快攻势 (例如12秒)，或调高放缓节奏。**注意：**开启 AutoSpawnTime 时，该间隔会被动态调整，并非每次严格固定。
- **inf\_DebugMode** (默认0)：调试模式。用于开发或管理查看插件内部运行信息。0=关闭；1=仅日志输出调试信息；2=控制台和日志输出；3=控制台+日志+激光束指示刷点 <sup>18</sup>。一般玩家无需开启，服务器管理员可在排查插件行为时设为2或3观察特感刷点、评分等详细信息。
- **inf\_NavBucketEnable** (默认1)：是否启用导航进度“分桶”筛选 <sup>19</sup>。开启时，插件将地图Nav区域按进度百分比分为101个桶 (0%~100%)，并主要在与目标幸存者邻近的桶范围内搜索刷新点，提高效率 and 合理性。设为0则不使用分桶，插件会全图搜索刷新点 (效率较低，不推荐关闭)。
- **inf\_NavBucketWindow** (默认10)：静态桶窗口半径 <sup>19</sup>。当动态联动关闭时，插件在目标幸存者所在桶的前后各多少个桶范围内搜索刷点。例如默认10表示在目标进度±10%的范围内寻找刷新位置。地图越大可适当增大此值，但过大可能搜索到离幸存者过远的位置。
- **inf\_NavBucketLinkToRing** (默认1)：桶窗口是否跟随“扩圈半径”动态调整 <sup>20</sup>。开启时，随着插件逐步扩大搜索半径 (ring值)，桶窗口半径将在设定的Min/Max范围之间线性变化，使远距离扩圈时搜索更多桶范围。默认开启。关闭则始终使用固定窗口 (**inf\_NavBucketWindow**) 大小。

- **inf\_NavBucketWindowMin** (默认6) / **inf\_NavBucketWindowMax** (默认12) <sup>21</sup> : 动态桶窗口的最小/最大值。当当前扩圈半径ring较小时, 桶窗口收窄至Min (6); 当ring增大到阈值时, 桶窗口放宽至Max (12)。**inf\_NavBucketMinAt** (默认500.0) / **inf\_NavBucketMaxAt** (默认1500.0) 定义了对应ring距离阈值 <sup>22</sup> : ring ≤ 500时窗口=6, ring ≥ 1500时窗口=12, 中间按线性插值。一般无需修改这些值, 默认能适应大部分地图。只有在特感老刷不出来或总刷太远时, 可调整窗口大小上下限。
- **inf\_NavBucketFirstFit** (默认0) : 分桶找点时是否启用“命中即停”模式 <sup>23</sup>。默认0表示遍历所有候选点计算评分, 选择最佳的位置刷出特感; 设为1时则在找到第一个符合基本条件的刷点后立即返回, 不再比较其他点, 可能降低刷点品质但提高性能。建议保留默认0, 以确保特感刷在最优位置; 仅在极端性能需求下可考虑开启。
- **inf\_NavBucketIncludeCenter** (默认1) : 扫描桶序列时是否包括中心桶 <sup>23</sup>。中心桶即目标幸存者所在的Nav桶。默认1会把该桶也纳入刷点搜索 (但实际点仍需满足最小距离和不可见等条件), 有助于在幸存者附近 (同一区域但躲在角落等) 刷新特感。设为0则完全避开幸存者当前所在桶。一般保持开启, 可增加近距离埋伏的可能性。
- **inf\_DeathCooldownKiller** (默认1.0 秒) : 杀手类特感 (Hunter、Smoker、Jockey、Charger) 死亡后的最小补位冷却时间 <sup>24</sup>。即当某一杀手类特感被击杀后, 至少经过1秒插件才会生成同类的新特感, 避免同种特感过于频繁连刷。可适度调高如2~3秒以降低同类连续出现频率, 默认1秒已足够小确保流畅进攻。
- **inf\_DeathCooldownSupport** (默认2.0 秒) : 支援类特感 (Boomer、Spitter) 的死亡冷却时间 <sup>24</sup>。默认2秒, 比杀手类稍长, 主要防止短时间内连续出现双 Boomer 等情况。可根据需要调整, 默认值在多数情况下表现良好。
- **inf\_DeathCooldown\_BypassAfter** (默认1.5 秒) : 死亡CD绕过阈值 <sup>25</sup>。当距离上次成功刷出某特感已超过此秒数时, 插件暂时忽略其死亡冷却限制。简单来说, 如果某类特感一段时间没刷出来, 即使按正常CD未到时间, 也会允许刷出, 避免长时间“饿死”不刷。同理, **inf\_DeathCooldown\_Underfill** (默认0.5, 系数) 则用于低存活补充 <sup>26</sup> : 当场上存活特感数量低于上限 \* 0.5 (默认) 时, 也忽略死亡CD立即补充特感。两个参数组成“双保险”, 确保在特感严重不足或长时间没有特感时, 不会因为CD限制而不刷新的情况。
- **inf\_TeleportSpawnGrace** (默认2.5 秒) : 特感生成后的传送宽限时间 <sup>27</sup>。特感刚刚刷出后在此秒数内不会被传送, 即使未被幸存者看到也暂时不计入Teleport计时 <sup>28</sup>。这避免了特感出生即被瞬移的尴尬情况, 给特感一定时间接近幸存者。一般保持2.5秒即可。
- **inf\_TeleportRunnerFast** (默认1.5 秒) : “跑男快速传送”阈值 <sup>29</sup>。当检测到有幸存者远离队友疯狂前进 (俗称跑男) 且当前无其他传送排队时, 针对该幸存者相关的特感传送判定时间将取普通阈值和此值的较小者 <sup>30</sup>。默认1.5秒意味着跑男情况下特感只需1.5秒未被看到就可传送, 加快追击步伐 (但不低于0.8秒的安全下限 <sup>31</sup>)。建议保持默认, 有效打击落单猛冲的幸存者。
- **inf\_NavBucketMapInvalid** (默认1) : 是否映射无效流程的Nav区域到最近的有效桶 <sup>32</sup>。开启时 (默认), 如果某Nav区域没有正常的Flow值 (例如偏离主线的区域), 插件会将其归入邻近的有效进度桶, 避免这些区域被永久跳过或评分异常。一般应保持开启以充分利用地图上的可用隐藏点。
- **inf\_NavBucketAssignRadius** (默认0.0, 无限) : 重新分配无效Flow区域时的最大XY距离 <sup>33</sup>。即当上项开启时, 搜索邻近有效区域的半径限制。0表示不限制距离 (默认策略: 找到最近的有效桶)。通常无需修改, 除非出现偏离太远区域被不恰当地映射的问题, 可设一个限值控制映射范围。

- **inf\_NavCacheEnable** (默认1)：是否启用Nav分桶结果的**磁盘缓存**<sup>34</sup>。开启时，插件会将每张地图Nav区域的Flow桶划分结果保存到磁盘（data/infdb\_buckets 目录），下次加载该地图时直接读取，节省重新计算时间。建议开启，加快地图初次载入时的准备。若需测试Nav变化或调试，可暂时关闭清除缓存。
- **inf\_score\_w\_dist** / **inf\_score\_w\_hght** / **inf\_score\_w\_flow** / **inf\_score\_w\_disp** (默认值分别见下)：特感刷新点**评分系统的权重配置**<sup>35</sup>。这四个ConVar分别对应距离、高度、流程、分散度四项评分，每项包含6个数值，依次对应Smoker、Hunter、Spitter、Boomer、Jockey、Charger六种特感的权重。例如 **inf\_score\_w\_hght** 默认值 "2.20 1.10 1.60 1.40 0.60 0.60"<sup>36</sup> 表示高度分在计算各类特感刷新点分数时，Smoker权重2.20（非常重视高度优势），Hunter 1.10，Spitter 1.60，Boomer 1.40，Jockey/Charger仅0.60（不太依赖高度）。类似地距离分默认Smoker权重最高1.35，Boomer最低1.05，等。**一般玩家无需修改这些高级参数**，除非对插件默认的刷新点选择策略非常了解并希望微调特定特感的行为。保持默认即可得到较合理的结果。如需调整，确保6个值用空格分隔，并且总体不要将某项权重设为0（插件内部会将0当作1处理<sup>37</sup><sup>38</sup>）。
- **inf\_PathCacheEnable** (默认1)：是否启用路径不可达判定的缓存。插件会检测候选刷新点到幸存者位置的路径是否**完全不可通行**（如被地图隔断），若是则给予一个高罚分。启用缓存可将检测结果暂存，避免对相同路径重复计算。一般保持开启即可。
- **inf\_PathCacheQuantize** (默认50.0)：路径缓存距离量化步长。当上项开启时，插件对路径距离进行量化处理以提高缓存命中率。例如默认50单位，将实际距离截断为50的倍数再缓存结果，避免细微距离变化导致重复计算。通常不需更改。
- **inf\_flow\_bad\_penalty\_points** (默认30.0)：坏Flow**惩罚分**<sup>39</sup>。针对那些被映射为“流程不连贯”（badflow）的Nav区域，插件在Flow评分中额外扣除的分值上限。默认30分表示这些区域在流程评分上最多扣掉30分，使它们相对不那么有利但仍可能被选中（0则表示不扣分，完全信任映射结果；100则表示几乎完全避免这些区域）。一般保持默认即可，除非发现特感总爱刷在离路线太远的位置，可尝试调高惩罚分进一步降低这些点的评分。

以上 ConVar 默认值均已过优化调校，通常不需要大幅调整。**一般推荐配置**：除非明确知道某项的作用并希望改变插件行为，否则保持默认设置即可获得平衡的特感刷新体验。如果要调整，可在服务器配置文件中通过 `sm_cvar <名字> <值>` 设置，并结合插件日志观察效果。

### 3. #define 常量解释

插件源码中定义了一系列常量（#define 宏），控制着特感刷新算法的细节。这些常量通常不直接暴露给服务器管理员，但了解其作用有助于理解插件机制。以下是对影响特感刷新的关键常量的说明：

- **BAIT\_DISTANCE** = **200.0**<sup>4</sup>：诱饵距离。插件的“反诱饵”机制相关阈值。当幸存者之间距离过大时，插件可能判定某位幸存者为“bait”诱饵，引导特感攻击他或加速传送。200单位大致相当于幸存者非常近的距离（约4米）。具体逻辑中，当所有幸存者均集中且彼此距离<200时，认为没有跑男或诱饵；若有人远超此距离，则触发相应逻辑来处理跑男（见下）。
- **RING\_SLACK** = **350.0**<sup>4</sup>：扩圈弹性距离。当插件逐步扩大特感搜索半径（ring）时，会给予一定的“弹性”距离余量。特别是在候选刷新点处于比幸存者高的地方时，允许适当增加搜索半径。350单位是弹性基础，上限可达其两倍<sup>40</sup><sup>41</sup>。简单说，**RING\_SLACK**提供了垂直方向的容差，使得高处的刷新点即使水平距离稍远，也能在合理ring范围内被选中，提高高打点的可能性。
- **SUPPORT\_EXPAND\_MAX** = **1200.0**<sup>4</sup>：支援特感最大扩圈。对 Boomer、Spitter 这两类支援型特感，插件限制其搜索半径不会超过1200，即使 `inf_SpawnDistanceMax` 设得更高，也以1200为封顶<sup>42</sup>。这

是因为支援特感行动和攻击距离有限，刷得太远难以发挥作用。该常量保证支援特感不会刷出在过远处。

- **LOW\_SCORE\_EXPAND** = 100.0<sup>43</sup>：评分不足时扩圈步长。每当一次刷新点搜索未找到符合条件的位置，插件会将当前搜索半径ring增加100单位，然后重试<sup>44</sup>。这个100就是扩圈增量。若多次扩圈仍无结果，就一直每次+100，直到达到上限（SpawnDistanceMax或SupportExpandMax）。因此LOW\_SCORE\_EXPAND越大，插件每次找不到点时扩圈越快，但可能跳过一些中间距离的潜在点；太小则扩圈过慢可能尝试次数过多。100是均衡值，无需改动。
- **ENABLE\_SMOKER** ~ **ENABLE\_CHARGER** = (1<<0)...(1<<5)<sup>45</sup>：这些常量用于构造前述的特感类型掩码（inf\_EnableSloption）。每个常量代表允许某特感的比特位，如ENABLE\_SMOKER=1表示Smoker位。它们组合起来生成默认的63（即二进制111111）。玩家层面直接用inf\_EnableSloption设置即可，不必单独理会这些宏。
- **SPIT\_INTERVAL** = 2.0（秒）<sup>46</sup>：Spitter吐痰间隔常量。插件会记录每个Spitter上次吐痰的时间，用于决定某些刷出或传送策略。例如插件避免在已有一个Spitter刚吐完痰后的极短时间内再次生成新的Spitter，以防止吐痰频率过高。2.0秒是一个基准间隔，用于判断Spitter技能冷却是否完成等<sup>47</sup>。玩家无需调整，它在代码中固定。
- **RUSH\_MAN\_DISTANCE** = 1200.0<sup>5</sup>：判定跑男距离。当有幸存者距离队友超过此距离（1200单位，大约相当于《求生之路2》中的一个视野距离）时，插件认为出现了“跑男”行为<sup>48</sup><sup>49</sup>。这会触发两方面：1）传送判定加快（对应TeleportRunnerFast）：插件将更快传送落后的特感到前方截击跑男<sup>30</sup>；2）目标选取偏向跑在最前的幸存者<sup>50</sup><sup>51</sup>，让特感优先生成并攻击他，从而惩罚远离队伍的行为。该距离在代码中定义，不可配置。
- **FRAME\_THINK\_STEP** = 0.02（秒）<sup>5</sup>：插件内部帧循环步长。infected\_control 插件在后台以0.02秒为周期循环检查并处理特感队列生成，相当于每秒运行50次逻辑，用于迅速刷出多只特感。这个高频循环由OnGameFrame驱动<sup>52</sup><sup>53</sup>。Frame步长越小，特感几乎同时刷新的效果越好，但也增加CPU开销。0.02即一帧，大致与游戏帧率同步，勿修改。
- **SUPPORT\_SPAWN\_DELAY\_SECS** = 1.0（秒）<sup>54</sup>：支援特感延迟刷出时间。插件在每波刷新开始时，优先确保杀手类特感出现，然后隔1秒再允许支援类特感（Boomer/Spitter）加入队列<sup>55</sup><sup>56</sup>。这样可以避免一波中全是Boomer/Spitter而缺少攻击型特感的情况。例如每波开始的前1秒内若杀手类不足，插件只挑选杀手类入队，1秒后再补充支援类。此常量固定为1秒，不直接暴露。
- **SUPPORT\_NEED\_KILLERS** = 1<sup>54</sup>：支援需要杀手标志。配合上项使用，当值为1时，每波刷新至少保证有1个杀手类特感。如果队列中本来全是支援类且没有杀手类，插件会强制用一个杀手类替换其中一个支援类<sup>55</sup>（因为默认SUPPORT\_NEED\_KILLERS=1且killersNow<1会触发preferKiller逻辑）。这保证了每波攻击都有实质威胁。该值为宏定义，始终为1。
- **分散度参数**：插件为避免特感总从同一方向出现，设计了一套分散度判定参数：
  - **SEP\_TTL** = 3.0 秒<sup>57</sup>：记录最近刷新的位置（Nav区域）在此时间内不可重复使用。即刚刚刷出过特感的点会进入冷却3秒，迫使下次换个点。<sup>58</sup>中还定义了**NAV\_CD\_SECS=0.5** 秒，用于给尝试刷失败的Nav点一个0.5秒的短暂冷却<sup>59</sup>。
  - **SEP\_RADIUS** = 80.0：距离判定半径。当两个潜在刷点距离小于80单位时，视为过近，在分散度评分中会受额外惩罚，鼓励刷点不要过于集中。
  - **SECTORS\_BASE** = 6，**SECTORS\_MAX** = 8，**DYN\_SECTORS\_MIN** = 3<sup>58</sup>：分散扇区划分基数和动态范围。默认将幸存者周围360°划分为6个扇区；插件可根据当前存活特感数动态调整最多8个扇区、最少3个扇

区<sup>60</sup>。特感刷点若总在同一扇区，会被判定分散度不佳。插件偏好在不同扇区交替刷特感，以包围幸存者。

- **罚分/加分常量：**例如 `SECTOR_PREF_BONUS_BASE` = -8.0（负值代表奖励分）表示优先未用扇区时可获-8分加成（分数越低越优先）；`SECTOR_OFF_PENALTY_BASE` = 4.0表示特感若不在优选扇区则加4分（略劣）；`RECENT_PENALTY_0/1/2_BASE` = 3.6/2.4/2.0 分别是针对最近1、2、3次使用过的刷点扇区的惩罚分值<sup>61</sup>。这些数值越高，表示越不愿重复，用负值表示奖励使用全新方向。插件综合这些参数计算“分散度分”，保障特感进攻方向多样。

上述宏均在源码中定死，一般不需也无法修改。它们体现了作者对游戏机制的理解和调优，比如**扩圈节奏**（`LOW_SCORE_EXPAND`）决定了特感刷不出时如何逐步扩大范围、**跑男距离**决定了何时判定有人离队过远等。了解这些常量有助于理解插件背后的设计思路，但正常使用时只需通过前述ConVar配置来影响行为。

## 4. 评分系统介绍

插件的**特感刷新点评分系统**决定了特感将刷在哪个具体位置。每当需要生成特感时，插件会搜集多个候选位置，并从四个维度对每个位置打分，综合权重计算总分，选择**最低分**（即最优）的地点刷出特感<sup>62</sup>。评分系统旨在量化“这个刷点好不好”的概念，分值越低表示位置越有利于特感进攻。评分维度和机制如下：

- **距离分（Distance Score）：**衡量刷点与幸存者之间距离的适宜程度。既不能过近被立即发现，也不能过远错失进攻机会。插件会设定一个“理想距离范围”（由SpawnDistanceMin/Max决定），候选点距离在此范围中心附近得分最低，更接近或更远都会增加分值。这样特感倾向于在中等距离埋伏。例如默认下Smoker理想距离较远，因此Smoker的距离权重最高1.35<sup>36</sup>，会强烈倾向选距幸存者远一些的位置；相反Boomer距离权重低（1.05），更允许近距离出现以奇袭。
- **高度分（Height Score）：**衡量刷点相对幸存者高度差的优势。通常高于幸存者的点对特感更有利（例如Smoker在高处更难被打、Hunter高跳扑击更突然），而低于幸存者可能劣势。插件计算候选点的高度分：高出幸存者一定范围可获得**负分奖励**（降低总分），低于幸存者太多则增加分值惩罚。例如Smoker默认高度权重最高2.20，非常偏好高点<sup>36</sup>；Boomer/Spitter也有一定高度加成；而Charger、Jockey高度权重只有0.6，说明他们不太依赖高度。如果某刷点高出幸存者一层楼左右，会得到明显更好的高度评分，从而更可能被选中。
- **流程分（Flow Score）：**这里的“流程”指地图推进进度。插件预先为地图每个Nav区域计算了一个**流程百分比**（0%=起点，100%=终点），代表该区域在地图通关路线中的相对位置。然后比较刷点所在位置与目标幸存者当前所在位置的流程差异：
  - 若候选点在**幸存者前进路线的前方**（流程百分比更高），通常更理想——特感可以埋伏前方，迫使幸存者进入伏击圈。
  - 若在**幸存者身后**（流程更低），特感相当于被幸存者越过，进攻价值低，会被扣分。
  - 若在支线或偏离主线路径（“raw flow”无效区域），则标记为badflow，在Flow评分中还会额外扣分（默认为30分上限，见前述FlowBadPenaltyPoints）。插件会尝试将badflow区域映射到最近的主路流程，但依然给予轻微惩罚以避免特感老刷在离队伍路线过远的死角。

综合而言，**流程分**确保特感刷在幸存者尚未到达的区域（埋伏在前），或者在侧翼，而不会生成在他们已经经过的背后区域（背后刷新的特感威胁较低）。例如Boomer的流程权重较高1.35，更倾向出现在幸存者前方拐角；而Hunter权重1.00，则相对不太Care流程位置<sup>63</sup>。插件计算一个基础Flow分（0~100，数值越高表示刷点相对越靠后不利）并减去高度惩罚（针对badflow情况下候选点低于幸存者的额外惩罚）得到最终流程分<sup>64</sup>  
<sup>65</sup>。

- **分散度分（Dispersion Score）：**用于鼓励**进攻方向多样化和刷点分散**。插件将幸存者周围划分成若干扇区，并记录最近几次特感刷出的扇区和具体Nav区域：

- 若候选刷点处于一个近期未曾使用的新扇区，则获得**奖励**（降低分值）<sup>61</sup>；反之如果重复最近用过的方向，则增加扣分。比如默认“新方向奖励”是-8分，每重复一次最近用过方向罚+（2.0~3.6）分递减<sup>61</sup>。
- 如果候选点与上一次刷出的点距离过近（<80单位），也会被扣分，防止两个特感刷在几乎同一个位置被一网打尽。
- 每个具体Nav区域在被用过后会进入3秒冷却（SEP\_TTL），冷却未过再次选中会被直接否决<sup>66</sup>。

分散度分保证了特感不会扎堆从同一处攻击。例如一波中一个Hunter刷在左侧高处，则下一只特感更可能刷在右侧或后方，形成夹击。此外即便方向相同也会尽量隔一定距离刷出。Charger和Jockey对分散度要求最低（权重仅0.8和0.9），意味着它们有时可以配合一起从同侧进攻；而Boomer分散度权重最高1.30，希望它常不同于上一次的位置现身，以提高突然性<sup>67</sup>。

插件综合以上四项计算每个候选点的**总分**：

总分 = 距离分\*W\_dist + 高度分\*W\_hght + 流程分\*W\_flow + 分散度分\*W\_disp （W为该特感类型对应权重）

<sup>62</sup>。权重的默认配置如前述，一般使得**Smoker**最看重高度和距离，**Boomer**看重分散度和流程位置，**Charger**和**Jockey**则相对平均。插件会选取总分最低的点刷出特感，如果启用了FirstFit模式则可能在找到第一个满足基本条件的点时就提前刷出<sup>23</sup>。这种评分系统使刷出的特感更具威胁和合理性：例如Smoker经常在远处高台出现，Hunter可能从侧翼扑来，Boomer不会老从同一个角落冒出。这为对抗双方都提供了更丰富的策略体验——幸存者不能只盯一个方向，感染者也能更聪明地利用地形。

注：可以通过设置调试模式为3（`inf_DebugMode 3`）来观察插件实时打印的评分细节。插件会在控制台输出每个候选点各项得分以及最终选择结果，方便高级玩家或管理员了解评分系统的运作原理并验证调整效果。

## 5. 找点系统介绍

特感的**刷出位置寻找系统**（简称“找点系统”）是插件的核心之一。它决定特感具体会刷新在地图上的哪个坐标。插件运用了导航网格（Nav Mesh）信息，通过**分桶筛选** + **扩圈扫描**的方式，高效地找到适合的隐藏刷新点。其逻辑细节如下：

- **Nav分桶预处理**：加载地图时，插件使用 Left4DHooks 接口遍历所有Nav区域，按照其距离地图终点的远近计算“通关进度百分比”（Flow值）<sup>68</sup>。然后将Nav区域按百分比0~100分成101个桶（FLOW\_BUCKETS=101）<sup>69</sup>。第0桶代表地图起点附近区域，100桶代表终点安全屋附近区域，中间值依次平铺地图路线。这相当于把地图主路径按进度切成了101段。这个分桶结果会缓存到磁盘以加快下次载入<sup>34</sup>。
- **目标幸存者选择**：每次刷新特感前，插件都会决定一个“目标幸存者”（Target Survivor），即这次进攻主要围绕哪个幸存者展开<sup>50</sup><sup>70</sup>。默认情况下，如果没有特殊情况，插件会从当前存活且未倒地的幸存者中**随机**选择一名作为目标<sup>71</sup>。这样特感可能集中袭击随机的一个人，避免总盯同一人。如果发现队伍中出现“跑男”（某人生存者远离队友太远），插件会优先把最领先的那个跑远幸存者作为目标（`bPickRushMan=true`）<sup>50</sup>，以便特感专门刷在他附近打击其孤立行为<sup>51</sup><sup>72</sup>。另外，若集成了目标限制插件（`si_target_limit`）且某幸存者本回合已被攻击很多次，则插件会暂时跳过他选别人<sup>73</sup>，防止“集火”单人过度。确定目标幸存者后，插件记录其所在Nav区域的桶编号s。
- **桶窗口筛选**：插件接下来仅在目标幸存者附近的一系列桶范围内搜索可刷新的Nav区域，以减少无关区域的考虑。默认使用“半径±N桶”的方式：桶窗口初始大小取`inf_NavBucketWindow`（默认±10）<sup>74</sup>。例如目标位于桶50，则初始考虑桶40~60之间的所有Nav区域作为候选。这意味着特感一般只会刷在距目标前后各约10%的地图进度内的范围，不会离他太近的已经经过区域或太远的前方区域。**动态桶窗口**如果开启（`inf_NavBucketLinkToRing=1`），则桶半径会根据当前扩圈半径大小动态调整在`NavBucketWindowMin` ~ `Max`之间<sup>21</sup>：在特感搜索范围较小时用较小桶窗，范围扩大时桶窗也同步放

大，以防过远扩圈时漏掉更前面的区域<sup>75</sup><sup>76</sup>。桶映射时，若某Nav区域因地图结构没有Flow值（如支线死胡同），插件会将其临近映射到最近的有效桶上，以免直接跳过<sup>32</sup>。

- **严格的扫描顺序：** 在确定需要检查的桶集合后，插件不会盲目随机挑选点，而是按照预设顺序**由近及远**地扫描这些桶中的Nav区域：
  - 首先如果配置允许，可能包括**中心桶**（目标所在桶）<sup>77</sup>；然后从目标桶开始，按照“前进N、后退M”交替的模式向外扩展检查<sup>78</sup><sup>79</sup>。初始设置是前进方向2个桶、后退1个桶的批次，即顺序类似：目标桶(s) → s+1 → s+2（前2）→ s-1（后1）→ s+3→s+4（前2下一轮）→ s-2（后1下一轮）→ ... 以此类推<sup>80</sup><sup>81</sup>。这个顺序保证更**靠前（幸存者前进方向）的区域优先**，偶尔插入一些后方区域检查，以防身后有合适埋伏点。并且算法会动态调整批次大小：如果发现前方桶加入候选的次数比后方多出超过4次，则下轮起同时增加前后批次各+1<sup>82</sup>，逐渐加大扫描面。如果启用了FirstFit模式（inf\_NavBucketFirstFit=1），一旦在某桶找到合格刷点就立刻停止，不再继续后续桶<sup>23</sup>。若FirstFit关闭，则会遍历所有桶范围内的可能点并对每个点评分，最后取分最低者。
- **候选点条件检查：** 对于每个被扫描到的Nav区域，插件调用Left4DHooks提供的 **L4D\_FindRandomSpot** 在该区域内找一个具体坐标<sup>83</sup>。找到坐标后需通过多重条件校验，确保它适合刷特感：
  - **距离窗口：** 刷点与目标幸存者的距离必须在SpawnDistanceMin和当前ring（扩圈半径）之间<sup>84</sup>。SpawnMin保证不会太近，ring则是当前搜索的外半径。刚开始ring=SpawnMin，若找不到会逐步增加。插件默认初始ring≈250，逐步增加100单位<sup>44</sup>。
  - **视线遮蔽：** 刷点不能**直接可见**任意幸存者。插件通过多种方式确保这一点：包括模拟幸存者眼睛和脚的位置进行视线(trace)检测，以及检查Nav属性中的“不可见”标记等<sup>85</sup>。若 IsPosVisibleSDK 判定处于幸存者视野，则此点作废<sup>86</sup>。同时 inf\_IgnoreIncappedSurvivorSight 参数决定是否无视倒地者的视线（默认为无视），避免倒地幸存者视线干扰刷点判定。
  - **空间可用：** 刷点必须有足够空间放置特感模型，不会卡在墙里或掉落悬空。插件利用引擎函数检验生成特感的碰撞箱是否会与地图地形重叠（WillStuck测试）。此外插件确保该Nav区域在短时间内未被用过（NavCooldown）以增加分散性<sup>87</sup>。
- **类型匹配：** 某些特感有特殊需求，例如Smoker需要足够的高度差拉人，如果一个点完全无法实现拉扯作用，插件可能视为次优。不过一般这部分通过评分系统自然体现，不是硬性剔除条件。
- **扩圈与导演兜底：** 如果当前扩圈半径ring内按上述桶筛选遍历完**仍无任何合格点**（即没有通过上述条件检查的候选点），插件会触发“扩圈”流程：
  - **扩大ring：** 将搜索半径增加LOW\_SCORE\_EXPAND（100）单位，但不超过SpawnDistanceMax（对Boomer/Spitter则不超过SupportExpandMax 1200）<sup>88</sup><sup>44</sup>。同时如果开启动态桶窗口，则根据新的ring调整桶范围大小<sup>75</sup>。然后重新按照上述顺序扫描更多远处的区域。
  - **达到上限：** 如果扩圈后ring已经达到最大上限（即nextR = maxR）仍未找到点<sup>89</sup>，插件会启用**导演兜底方案**：调用游戏内置的特殊感染刷出位置（FallbackDirectorPos）寻找一个点<sup>90</sup>。这是最后的尝试，由游戏AI导演给出一个可能的位置，即使这个位置平时不满足严格条件（例如离得很远或略被看见边缘），插件也会尝试在此生成特感。
  - **兜底结果：** 若导演兜底找到并成功刷出特感<sup>91</sup>，则记录该次刷新的坐标用于分散度管理，同时将当前ring缩短至80%（略收缩），以免下一次仍从最大距离开始找<sup>92</sup>。如果连兜底都未能刷出（极少见情况，例如地图确实无隐蔽点），插件会在日志中记录失败并暂时跳过刷新<sup>93</sup>。
- **记录与反馈：** 一旦找到最终合适的刷点并成功生成特感，插件会：
  - 将该位置及其所在扇区记录到最近刷新列表，用于后续分散度计算<sup>94</sup>。



- 将该Nav区域加入冷却表，默认0.5秒内不再使用相同区域 <sup>59</sup>。
- 如果开启DebugMode>=1，会在日志中打印本次选择的刷点坐标、各项评分以及简要原因（例如“因为前方无点扩大范围到X”或“fallback@max成功”）。
- 在最高调试模式下（3），还会画出一条激光束从刷点指向目标幸存者，方便管理员在游戏中直观看到特感刷新的地点和朝向。

综上，找点系统结合“分桶筛选 + 扩圈扫描 + 多条件校验 + 评分排序”，确保每次刷新特感都尽可能生成在合理且具有威胁的位置。同时通过冷却和分散逻辑，避免重复利用同一点或同一方向，提升游戏的公平性和不确定性。一般玩家可能不感觉复杂算法的存在，只会觉得特感总是从各种出其不意但又说得通的地点冒出来，这正是该找点系统的功劳。

(提示：管理员可使用指令 `sm_infospawn`（如有提供）或在Debug模式下观察插件输出，来获取特定位置Nav的进度桶编号、Flow值等信息，以进一步了解刷点逻辑。)

## 6. 特感生成流程

有了前面的铺垫，我们来看插件完整的特感生成流程。这里用流程步骤梳理特感从“需求出现”到“生成刷出”的整个过程，以及插件在游戏中的工作节奏：

### 生成条件触发

1. **回合开始准备：** 每局对抗开始（地图加载或每回合重新开始）时，插件首先执行初始化工作，包括：禁止游戏导演自动刷特感（`director_no_specials` 设为1）<sup>1</sup>、重置内部状态（计数器、队列等）<sup>95</sup>、读取并应用当前ConVar配置、构建导航分桶缓存<sup>96</sup>。此时并不会立刻刷出特感，通常在幸存者还没出安全室阶段插件处于待命状态。
2. **延迟与首次进攻：** 为避开开局瞬间刷特感给幸存者没有准备时间，插件通常会在回合开始后稍作延迟再启动特感攻势（默认约数秒）。具体触发首次刷新的条件可以是：
3. 幸存者打开安全屋门或走出一定距离（插件通过幸存者Flow进度或位置变化判断“战斗正式开始”）。
4. 或一个固定的延迟计时器（比如10秒）倒计时结束。源码中 `Timer_SpawnFirstWave` 负责将 `bLate` 置 `true` 并启动首波刷新的计时<sup>97</sup>，通常由相应事件触发。

一旦条件满足，插件将 `bLate` 标志设为 `true`，表示进入战斗阶段，然后立即调用 `StartWave()` 开始第一波特感生成<sup>97</sup>。

### 每波刷新与队列处理

1. **建立刷出队列：** 每次开始新的一波（包括首波）时，插件会根据当前配置和局势决定本波要刷出的特感数和种类。典型情况下，每波会生成 `l4d_infected_limit` 指定数量的特感（默认6只）<sup>98</sup>。插件创建对应大小的生成队列（spawn queue），然后依次填入具体的特感类别：
2. 插件首先获取当前场上每类特感存活数量，与各类允许的上限（游戏CVar，如 `z_smoker_limit` 等）比较，计算尚可刷新的数量 `cap`<sup>99</sup><sup>100</sup>。例如默认每类上限1（Boomer/Spitter）或2（Hunter等），如果场上已经有1只Boomer，则Boomer的 `cap` 为0，不会再刷Boomer。
3. 考虑模式锁定：若 `inf_AllChargerMode` 或 `inf_AllHunterMode` 开启，队列将强制填满相应特感类型<sup>9</sup><sup>101</sup>，忽略正常随机。
4. **优先保证杀手类存在：** 插件检查当前存活和队列中的杀手类数量，如果为0且还在首秒内，则优先在队列中安排至少1个杀手类（Charger/Hunter/Smoker/Jockey）<sup>55</sup>。这利用了先前提到的 `SUPPORT_SPAWN_DELAY` 机制，确保不会出现一波刷出的全是Boomer/Spitter而缺少攻击手的情况。

5. **按稀缺度选择：** 在无特殊偏好时，插件倾向于选择目前场上**最稀缺**的特感类型优先入队 <sup>102</sup>。例如如果前一波刷了很多Hunter但没刷Smoker，那么下一波会优先刷Smoker，使各种类出场机会均衡（除非某类被掩码禁用）。
6. **随机兜底：** 如果经过上述规则筛选仍未决定类型，插件会在可用类型中随机挑选 <sup>103</sup>。挑选时会跳过不允许（被掩码禁用）的类型、已达到上限的类型，并考虑死亡冷却：若当前不处于“放宽CD”状态，刚死过的类型会被跳过 <sup>104</sup>。
7. 将选定类型加入生成队列（ArrayList）。如果队列未滿（小于iSiLimit），继续挑选下一个类型 <sup>105</sup> <sup>106</sup>，直到填满本波所需数量或无可用类型为止。
8. **死亡CD与放宽：** 在将选定类型最终放入队列前，插件会再次检查其**死亡冷却**是否满足 <sup>107</sup>。如果该类型仍在CD中且目前没有触发放宽条件（见ConVar DeathCooldown系列），插件会尝试用“替补”类型：
9. 替补优先再尝试找一个可用的杀手类 <sup>108</sup>，如没有则再随机找任意可用类型 <sup>109</sup>。
10. 如果找到替补，则用替补类型替换原类型入队，并记录日志说明 <sup>110</sup>。
11. 如果没有任何替补可选（极端情况：所有类型都在CD或者被禁），插件会打印日志  
`no eligible class -> wait`，然后**本帧暂不入队**，等待下一帧再重新尝试 <sup>110</sup> <sup>111</sup>。这样做是为了宁可稍微推迟，也不违反冷却规则。
12. **进入生成循环：** 队列准备就绪后，插件立即进入特感刷出的执行循环。插件主逻辑在OnGameFrame中以每帧0.02秒的频率运行 <sup>112</sup> <sup>113</sup>。对于生成而言，每帧会**尝试刷出队列中的一只特感** <sup>114</sup> <sup>53</sup>：
13. **检查上限：** 如果当前场上特感总数已达上限iSiLimit，则暂停生成 <sup>113</sup>（以防极端情况下某一帧内队列未更新导致超刷）。
14. **Teleport优先：** 如果传送队列中有待传送的特感且场上特感总数未滿，则先执行一次传送生成 <sup>114</sup>（见后述传送流程）。
15. **正常生成：** 否则，若spawn队列非空且总数未滿，则调用 `TryNormalSpawnOnce()` 处理队列头部的一个特感刷新 <sup>115</sup>。该函数执行先前详述的**找点和评分流程**，为队首的特感类型选点并尝试生成：
  1. 从队列头取出类型want <sup>116</sup>。
  2. 再次判断类型上限是否已达到（例如队列里准备刷Charger但此刻场上Charger没死又满上限了），如果是则丢弃该项（队列头出队）跳过 <sup>117</sup>。
  3. 检查死亡CD：如果仍在CD且不应放宽，则把该类型移到队列末尾、跳过当前生成（等下次再刷） <sup>118</sup>。这对应上面提到的CD轮换机制，例如队列头是Charger但刚死不久，则本轮让它排队末尾等待，下一只先刷别的。
  4. 选定目标幸存者 = gST.targetSurvivor。**注意：** 插件在每次正式生成操作前已经通过定时器Timer\_CheckSpawnWindow或TeleportTick选好了本轮的targetSurvivor <sup>119</sup> <sup>120</sup>，因此这里直接使用该值。
  5. 设置初始搜索距离ring = SpawnDistanceMin，并根据want类型决定最大搜索半径maxR = SpawnDistanceMax或SupportExpandMax <sup>121</sup>。
  6. 调用 `FindSpawnPosViaNavArea(want, targetSurvivor, ring, teleportMode=false, ...)` 寻找刷点 <sup>122</sup>。内部执行前述的桶序列扫描和条件筛选，找到最佳位置pos。如果找到了ok=true但pos仍被幸存者可见，则标记ok=false放弃（安全检查） <sup>86</sup>。
  7. 尝试生成：如果ok为true，则调用 `DoSpawnAt(pos, want)` 实际刷出该特感 <sup>123</sup>。生成可能失败（如引擎实体溢出、位置非法等少见情况），spawnOk表示结果。
  8. **成功处理：** 若ok && spawnOk都为true，表示特感已成功生成在pos：
    - 记录调试信息：调用 `LogChosenSpawnScore` 保存该刷点的评分详情（供日志或后续分析）。

- 更新Nav冷却：将所在Nav区域记入冷却表NavCooldown，持续 NAV\_CD\_SECS 秒（默认0.5秒）不再选用<sup>94</sup>。
  - 更新最近刷点列表：计算该点相对幸存者的扇区中心，记录到lastSpawns列表中<sup>124</sup>。
  - 维护计数：总刷出数totalSI+1，存活计数siAlive对应类型+1<sup>125</sup>。队列头已完成，出队，队列长度spawnQueueSize-1，队列待刷总数siQueueCount-1<sup>125</sup>。
  - 触发攻击指令：执行 nb\_assault 指令立即指挥AI发起进攻<sup>126</sup>。这会让生成的特感AI不再傻站着而是立刻行动（相当于求生之路AI Director的“进攻号令”）。
  - 调整扩圈参数：计算下一个起始ring = 本次ring \* 0.7，限制在SpawnMin~SpawnMax范围内<sup>127</sup>。这意味着如果本次扩圈到了较远才刷出，下次从稍近一些距离重新开始；如果本次很快在近点刷出，下次依然从最低距离开始搜索。
  - 打印调试日志，说明本次spawn成功及ring调整情况<sup>128</sup>。
  - 函数返回，结束本帧生成流程。
9. 失败处理：如果ok为false（找不到点）或者spawnOk为false（尝试生成失败）：
- 若确实尝试过生成但失败，且拿到了Nav区域索引arealdx，则对该区域调用短冷却（0.8秒），避免立即又尝试在同一区域刷出。
  - 扩圈续搜：计算 nextR = 当前ring + 100 (LOW\_SCORE\_EXPAND) 且不超过maxR，更新 gST.spawnDistCur = nextR<sup>129</sup>。
  - 打印调试日志，指出本次扩大搜索半径到nextR<sup>130</sup>。
  - 导演兜底：如果扩圈后已经达到最大半径（gST.spawnDistCur接近maxR），则调用 FallbackDirectorPosAtMax 尝试获取导演推荐点<sup>90</sup>，并用 DoSpawnAt 生成：
  - 成功则类似上面成功处理：记录分散位置<sup>131</sup>、计数、自身siQueueCount-等，并将当前ring乘0.8收缩<sup>92</sup>。
  - 成功与否皆打印日志提示。如果失败则该特感暂时刷不出来<sup>93</sup>。
  - 无论扩圈是否达到上限，函数本次调用结束（未成功刷出则不会移除队列头，也不减siQueueCount，意味着该特感留在队列中等待下次帧再尝试）。
10. 循环继续：返回OnGameFrame后，spawn队列若未空且场上未达上限，下一帧（0.02秒后）会再次调用TryNormalSpawnOnce，可能是同一个特感（如果刚才没刷出）继续扩圈，或队列下一个特感。如果队列空了或特感数满则不再调用生成，等待下一波时机。
16. 波次结束判定：插件通过 siQueueCount 跟踪每波应刷出的特感总数，每成功刷出一只就减1<sup>125</sup>。当siQueueCount降为0时，表示这一波预定的特感都已生成上场。此时并不意味着立即开始下一波，而需要等待一定时间（刷新间隔）。插件有一个每秒检查计时的定时器 Timer\_CheckSpawnWindow<sup>132</sup> 会处理波次间隔：
17. 该定时器每秒增加计数 lastSpawnSecs<sup>133</sup> 并检查是否到了触发下一波的时机。默认对抗下，要求经过至少 versus\_special\_respawn\_interval 秒（16秒）后才生成下一波特感<sup>11</sup>。插件实际使用1.5倍间隔作为延迟：当发现上一波已刷完（siQueueCount < limit），则设置bShouldCheck=false并启动一个定时器gST.hSpawn在 1.5 \* SiInterval 秒后调用StartWave开始下一波<sup>11</sup><sup>134</sup>。这样确保两波之间有一定缓冲，不会贴得过紧。
18. 如果开启了自动增时（inf\_EnableAutoSpawnTime=1），则 Timer\_CheckSpawnWindow 有另一套条件<sup>12</sup><sup>135</sup>：例如当所有攻击型特感都被击杀且场上没有排队特感时，或者场上总特感数低于上限的1/4且无队列时，亦或已经过去了一半刷新间隔时间无论如何，都将触发提前刷新下一波（此时使用正常间隔而非1.5倍）<sup>135</sup>。这些条件意味着：若幸存者很快清理了大部分特感，则下一波不会傻等16秒之久，可能8秒后就来了；反之如果还有不少特感存活则按正常/延长间隔来。
19. 该定时器也考虑特殊情况暂停刷新：如果场上出现Tank或者一半以上幸存者倒下，为了不给幸存者雪上加霜，会暂缓下一波直到局势恢复<sup>136</sup>。
20. 一旦定时器条件满足启动下一波，它会调用 StartWave() 并重置相应标志，循环往复。

## 传送与补位

1. **掉队特感传送：** 插件另有一个每秒运行的定时器 `Timer_TeleportTick` 专门监视特感是否需要传送<sup>137</sup>。它循环检查每个存活的特感Bot：
2. 若该特感不符合传送资格（比如玩家控制的特感或已在幸存者视野中等），跳过<sup>138 139</sup>。
3. 对符合条件的特感，先检查**出生宽限**：如果它生成后还没超过`TeleSpawnGrace`秒（默认2.5秒）<sup>28</sup>，则不累积传送计时并跳过，以免刚刚出来立刻传送<sup>140</sup>。
4. 然后检测该特感是否在**任一幸存者视线外**<sup>141</sup>（这里以幸存者眼睛位置做LOS判断）。如果是不可见：
  - 若是Smoker且安装了特殊Smoker技能插件（如新舌头机制），插件会检查它的舌头是否冷却好，未准备好则暂不传送并清零计数（避免把还不能攻击的Smoker传送浪费机会）<sup>47 142</sup>。
  - 将该特感的不可见计数 `teleCount[c]` 加1（每秒累加）<sup>143</sup>。
  - 计算当前需要的不可见秒数阈值：通常是`TeleCheckTime`（默认5秒）；如果检测到目前存在“跑男”且传送队列空闲，则取`TeleCheckTime`和`TeleportRunnerFast`较小值，但不低于0.8秒<sup>30</sup>。例如有跑男时阈值会降到1.5秒。
  - 若 `teleCount >= needSecs`，则满足传送条件：确定该特感的类型`zcx`，如果是1~6（普通特感），将该类型加入**传送队列**（`teleport queue`）<sup>144</sup>。同时：
    - 传送队列计数`teleportQueueSize+1`；
    - 从场上存活计数`siAlive`中将该特感移除（相当于视为“暂时消失”）<sup>145</sup>；总数`totalSI`减1；
    - （插件并不会真的杀掉它，只是从统计上减去，这样后续可以刷新新特感不受它占位的影响）。
    - 打印日志说明将某特感加入传送队列。
5. 简而言之：一个特感如果在幸存者看不见的位置停留超过5秒（跑男情况1.5秒），它就会被认为需要传送，插件把它“收回”放入`teleport`队列中准备稍后重新生成。
6. **执行传送刷新：** `OnGameFrame`每帧循环会优先处理传送队列<sup>114</sup>。当`teleportQueue`非空且场上特感数未达上限时，调用 `TryTeleportSpawnOnce()` 进行一次传送刷出<sup>114 53</sup>。这个流程与普通生成类似，但有些不同点：
  7. 传送队列中的类型`want`取出后，不经过队列填充，因为`tele`队列里已经确定了具体某只特感的类型。
  8. 也不检查死亡CD，因为传送的特感本来就是存活的，不算新刷，在冷却机制中它的死亡时间未更新，不应用死亡CD。
  9. 搜索刷点时参数`teleportMode=true`<sup>146</sup>，这可能使`FindSpawnPos`函数使用略不同的条件（例如允许稍微离得更近？以及Teleport传送出来的特感通常希望出现在幸存者前方更远一点）。实际上Teleport的初始`ring`使用单独维护的 `teleportDistCur`，初始等于`SpawnMin`<sup>147</sup>。
  10. 找点和扩圈逻辑基本相同（也在Nav桶中找，只是不需要考虑类型上限等问题）。若需扩圈，同样每次+100，Teleport没有像支援那样的特殊上限，就用`SpawnMax`（默认1500）。
  11. 一旦找到点则调用 `DoSpawnAt` 生成。这其实是**将该Bot瞬移到新地点**——游戏上看就是它突然出现在幸存者前方的新位置继续进攻。
  12. 传送刷出成功后，不增加总特感计数（因为只是移动了原有特感），但`teleport`队列长度 `teleportQueueSize`减1<sup>148</sup>。插件也会重置该传送距离参数`teleportDistCur`，为下一次传送做准备<sup>149</sup>。
  13. Teleport生成同样遵循扩圈->兜底流程。如果在最大范围都没找着点，也会调用导演兜底。成功则将特感传送出来，失败则暂且留它在队列下次再试。
  14. 每次传送成功后，插件会将该Bot对应的上一次生成时间`g_LastSpawnTime`更新<sup>150</sup>，防止立即又被当作新刷出特感算宽限；`teleCount`计数也清零。
15. **特感死亡与补位：** 当特感被击杀时（`PlayerDeath`事件），插件做如下处理：

16. 若死亡的是普通特感（类型1~6），调用 `TouchDeathCooldownOnce(zc)` 记录死亡时间 <sup>151</sup>。如果该类型当前没有冷却在进行，则启动冷却计时（`g_LastDeathTime`设为now）；如果已在冷却期内再次死亡则忽略不刷新时间戳 <sup>152</sup> <sup>153</sup>。这样保证死亡CD的计算准确。
17. 将对应类型的存活计数`siAlive`减1，总数`totalSI`减1 <sup>154</sup>。
18. 对于玩家控制的Tank之外的Bot特感，如果是Spitter以外，都在0.5秒后将Bot踢出（Kick），以释放bot槽位 <sup>155</sup>（因为L4D2对抗模式特感槽固定，如果不踢Ghost会占位）。这与插件的平行插件可能有关，但总之Spitter死亡留下酸液需要多留一会，不立即踢）。
19. 如果场上总特感数因此低于某阈值并且AutoSpawn开启，CheckSpawnWindow定时器会更早触发下一波补位（见前述Underfill机制）。
20. 此外，当某只特感死去，它原先占用的刷新队列配额就空出。插件的每帧逻辑在检测到 `totalSI < iSiLimit` 且`bLate=true`时，会调用`MaintainSpawnQueueOnce`试图补充队列 <sup>113</sup> <sup>156</sup>。这意味着如果还有后续波次没启动，插件可能立即为空位准备新的待刷特感，不一定非等到下一波整体刷新。尤其当AutoSpawnTime开启时，出现特感损失就可能触发提前补充。

## 插件执行时序总结

- **回合开始 (RoundStart):** 插件StopAll重置状态，禁用导演刷特感 <sup>1</sup>、建立或读取Nav缓存，准备就绪但不主动刷出。
- **安全室阶段:** 幸存者未出安全室或刚出不久，插件等待 `bLate` 条件成立。期间Tank不会刷出特感，直至满足首波触发条件。
- **首波启动:** 幸存者行动触发`Timer_SpawnFirstWave`，将`bLate=true`，创建spawn队列并很快开始刷出首波特感。
- **进行中 (战斗阶段):**
  - **每帧 (0.02s):** 插件按照队列顺序刷特感，多个特感在极短时间内接连出现。先确保满足冷却/上限等，再找点生成。如果队列空或满员则空转。
  - **每秒:** 插件检查是否该安排下一波、处理特感传送资格和跑男检测。遇Tank或幸存者倒地多会暂缓下一波 <sup>136</sup>。传送检测则每秒增加计数，符合条件就将特感回收进队列准备传送 <sup>30</sup> <sup>157</sup>。
  - **事件驱动:** 特感死亡事件即时更新冷却和计数 <sup>151</sup>；吐痰事件记录时间 <sup>158</sup>；暂停事件使插件暂停计时器 <sup>132</sup>等。插件与Pause插件联动，暂停时Kill定时器以防影响计时 <sup>132</sup>。
  - **波次切换:** 当一波特感全部生成后，进入冷却间隔。默认16秒后下一波，但Auto模式下可能更早。期间如果幸存者推进很快或者攻击型特感都阵亡，插件会提前准备下一波（`lastSpawnSecs`达到一半间隔等条件） <sup>12</sup>。
  - **循环往复:** 如此反复，直到地图回合结束。每波都可能由2~6只特感组成，看配置和实际存活情况，波与波之间有短暂喘息供幸存者整理。
- **回合结束 (RoundEnd):** 插件接收到事件后，调用StopAll清理队列、计时器，并重置Nav缓存等 <sup>159</sup>。幸存者进入安全室或回合终止时不再刷新新的特感，等待下局开始。

通过以上流程可以看出，插件在每个阶段都有相应的控制：起始阶段不给幸存者“秒特感”的不公平；进行中每帧、每秒多层面确保特感不断但不至于扎堆或不合理；结束阶段及时清理停止。不管一般玩家是否理解这些细节，他们直观体验到的就是特感**按一定节奏波次出现**，在正常情况下每隔十几秒一波攻势；如果他们效率高提前清场，下一波会更早出现；如果出现脱战特感，很快就会瞬移到前方继续攻击，不会干等浪费时间。反之若坦克出现或队伍半灭，插件稍缓出怪以留些生存空间。这些都极大提升了对抗模式的流畅度和公平性。

## 7. 插件执行顺序

最后，从宏观上总结`infected_control` 插件在游戏中的各阶段执行顺序，帮助理解其工作节奏：

- **回合初始化阶段（地图载入/Reset）：** 插件拦截并取消原版特感生成 <sup>1</sup>。调用StopAll重置上一局残留的数据和计时器 <sup>95</sup>。开始Nav分桶构建（或读取缓存） <sup>96</sup>。在RoundStart事件中，插件设置一个短计时器在1秒后重置幸存者状态和应用上限（ApplyMaxSpecials） <sup>95</sup>，确保游戏开局状态正常（例如强制刷新倒地的幸存者等）。此时`bLate=false`，表示尚未进入特感攻击阶段。

- **幸存者出安全室：**一般在RoundStart几秒后，幸存者打开安全门或离开起始区域。插件通常在检测到幸存者开始行动后，通过Timer\_SpawnFirstWave将bLate置为true<sup>97</sup>。这同时启动了两个重要的循环定时器：
- **每秒检查定时器 (hCheck)**<sup>160</sup>：每1.0秒执行一次Timer\_CheckSpawnWindow，用于监控波次间隔和跑男状态等<sup>132</sup>。
- **每秒传送定时器 (hTeleport)**<sup>161</sup>：每1.0秒执行一次Timer\_TeleportTick，监测特感不可见时间以决定传送<sup>137</sup>。

然后插件立即调用StartWave()开始首波生成<sup>162</sup>：计算cap并填充spawn队列、设置spawnDistCur=SpawnMin、标记bShouldCheck=true等<sup>163</sup><sup>164</sup>。从这一刻起，插件进入战斗节奏。

- **每帧逻辑循环 (OnGameFrame)：**bLate=true后，每个游戏帧(~0.02s)插件执行OnGameFrame函数<sup>112</sup>：
  - 若当前特感总数 < 上限，则调用MaintainSpawnQueueOnce尝试补足spawn队列<sup>113</sup>（比如某特感死亡腾出名额，则立刻准备补一个新的）。
  - 若传送队列不空且总数未满足，则优先执行TryTeleportSpawnOnce进行一次特感传送<sup>114</sup>。
  - 否则若生成队列有待刷特感且总数未满足，则执行TryNormalSpawnOnce刷出队列头特感<sup>115</sup>。
  - 每帧最多处理一只特感的生成或传送，因此如果一波有多只特感，实际上会在连续的多帧内于极短时间逐个刷出。例如6只特感大约会在 $6 \times 0.02 = 0.12$ 秒内全部出现，看起来几乎同时。OnGameFrame循环保证了生成的高实时性和顺序控制。
- **每秒逻辑检查 (Timer\_CheckSpawnWindow)：**并行于帧循环，每秒插件都会：
  - 增加计时lastSpawnSecs++<sup>133</sup>，表示距离上次开始刷新的时间。
  - 如果当前未进入战斗阶段（bLate=false），定时器直接停止（Plugin\_Stop）以节省资源<sup>133</sup>。
  - 如果当前没有待检查（bShouldCheck=false）或已经安排了下波计时器（hSpawn有效），则本次不做操作<sup>165</sup>。
  - 否则检查特殊情况：若幸存者进入撤离车或半数倒地且时间未过半间隔，则推迟刷新<sup>136</sup>。
  - 然后根据AutoSpawn开关决定刷新波次逻辑：
    - **固定间隔模式 (AutoSpawnTime=0)：**如果当前场上存活特感数等于上限（表示上一波尚未开始？实际一般不会出现，因为开始波时siQueueCount==limit但很快刷出就<limit），则重置计时<sup>11</sup>；否则（表明上一波已刷完或部分刷完），设置bShouldCheck=false并启动一个延迟计时器hSpawn，在 $SiInterval * 1.5$ 秒后调用StartWave<sup>166</sup>。这实现了“波与波之间1.5倍间隔”的效果。
    - **自动动态模式 (AutoSpawnTime=1)：**如果出现“所有杀手类已倒且队列空”或“场上特感 $\leq$ 上限1/4且队列空”或“已过半个基础间隔”的任一情况<sup>12</sup>，则同样判断：
      - 若当前场上已满上限（一般不会，除非bug），则重置计时；
      - 否则关闭bShouldCheck并设定一个计时器在 $SiInterval * 1.0$ 秒后调用StartWave<sup>167</sup>（注意这里乘1.0即正常间隔，不延长）。相当于满足上述条件时下一波提前到来，比固定模式快。
  - 这样，无论哪种模式，当满足条件时都会安排StartWave的定时调用。一旦计时到，StartWave再次执行新的波次生成逻辑。
- **特感传送监督 (Timer\_TeleportTick)：**每秒插件会：
  - 暂停处理：如果安装了Pause插件且游戏正在暂停，则本次不执行<sup>168</sup>。
  - 检查是否需要开启或关闭“跑男模式”：通过CheckRushManAndAllPinned函数计算幸存者分散情况<sup>169</sup><sup>120</sup>。如果找到一个幸存者远离队友且没有在他周围一定范围内的其他人，同时当前场上特感不多，则开启跑男标记bPickRushMan=true并记录rushManIndex<sup>72</sup><sup>170</sup>（以及发送Forward通知，可

用于HUD提示)。若原本有跑男状态但条件不再满足(如他被抓住或队友跟上来了),则关闭跑男状态<sup>171</sup>。这个状态会影响ChooseTargetSurvivor选择目标时是否固定选该跑男<sup>50</sup>和Teleport传送计时阈值的采用<sup>172</sup>。

- 遍历每个感染者Bot客户端ID:
  - 如果该Bot不可用或不是特感Bot,跳过<sup>138</sup>。
  - 如果它当前不满足传送的基本资格(例如已在幸存者视野中、有玩家控制等),跳过<sup>139</sup>。
  - 否则按前述逻辑计算其不可见累积时间teleCount,并根据跑男情况取阈值needSecs<sup>30</sup>。然后判断是否达到传送条件,达到则把它的类型加入teleport队列<sup>157 144</sup>。
- 该定时器保证了**每秒都会检查并累加**,所以一只特感一旦离开幸存者视野,每过1秒teleCount+1,超过5秒就会进入传送队列。跑男情况下最快1.5秒就进队列。进入队列后,其原Bot实体仍在游戏中但插件统计上视为已移除(siAlive减1),这样spawn队列那边可以考虑刷新的补位了。
- **传送执行(OnGameFrame 部分):** 如前所述,每帧循环会优先处理传送队列。如果teleportQueue非空,插件会将队首类型传送刷出:
  - 这实质上将对应的Bot重新定位刷出相当于一个新特感,但不影响计数(因为之前已经从计数里减掉)。
  - 传送刷出也用FindSpawnPosViaNavArea来找点,只是teleportMode=true可能略有不同策略(例如可能略微更远以出现在队伍前方)。
  - 传送出来的特感通常会骤然出现在幸存者前方或侧面,玩家视角看就像凭空冒出,其实是后方那只瞬移过来了。这提高了进攻连续性和压迫感。
  - 传送成功后清除teleCount并恢复计数,如果teleportQueue还有更多则下一帧继续传送下一只,直到队列清空。
- **循环与回合结束:** 插件就这样循环执行“生成-间隔-再生成”的周期,直到幸存者抵达安全室或被全灭,触发RoundEnd事件。RoundEnd时插件StopAll停止所有计时器和清空队列<sup>159</sup>。由于director\_no\_specials始终为1,若还有未结束特感也会被游戏清理。之后等待下一回合RoundStart,重新开始上述流程。

以上即为插件在游戏中各阶段的调用顺序概览。从每回合开始的初始化、幸存者离开安全室触发首波、到每帧每秒的循环调度、波次控制、特感传送、再到回合结束清理, **infected\_control** 时刻在运行,智能地掌控特感刷新的节奏和位置。对于一般玩家,这一切都是在幕后进行的,他们体验到的只是对抗节奏更紧凑、特感更狡猾难测;对于管理员或有技术兴趣的玩家,这份文档揭示了插件内部秩序井然的调度机制和思路,帮助大家更好地配置和运用此插件。

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29  
30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58  
59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87  
88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115  
116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143  
144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171

<sup>172</sup> infected\_control.sp

file:///file-2goZr6VQoKHJirJxRp4Lyc