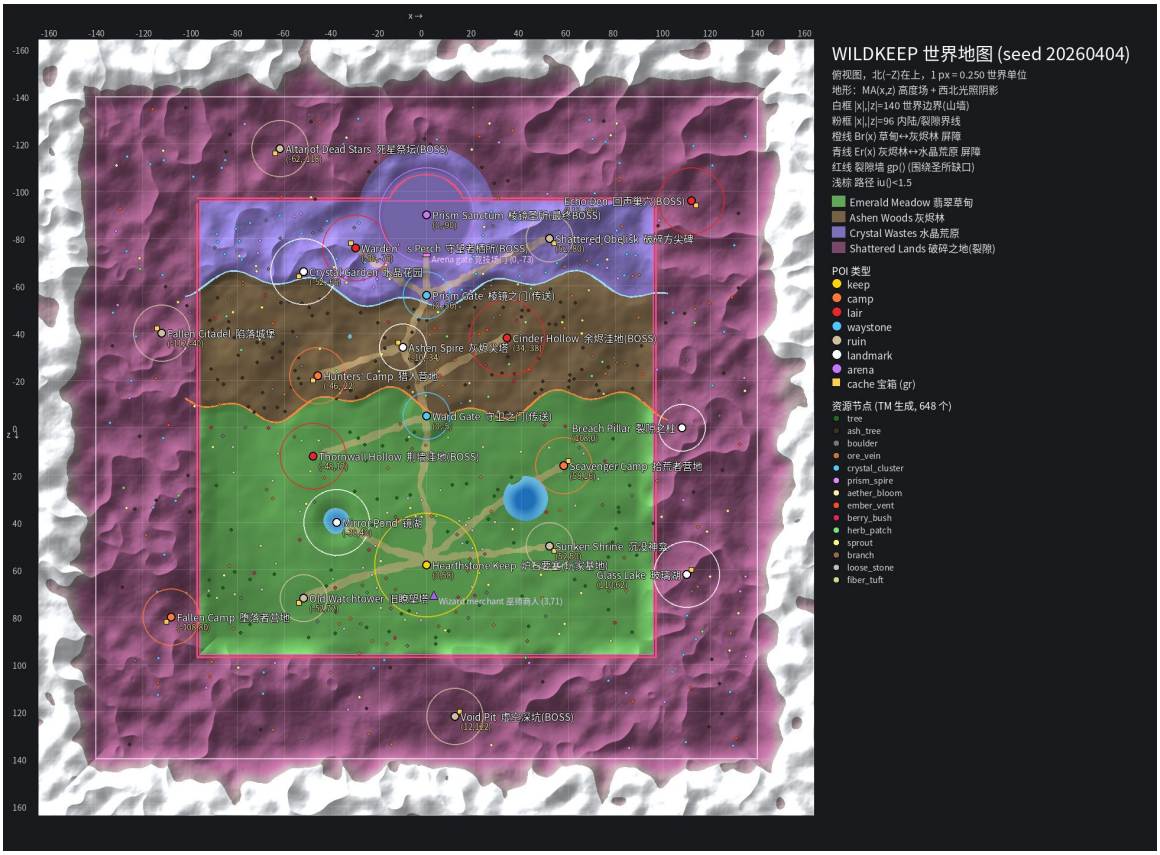
WILDKEEP 游戏资源与机制解析报告

来源：y8.com/games/wildkeep → item-cdn.y8.com/darkmilou/html5/wildkeep-y8/2026-10-10-065514-6d6e4c2285fc/ 引擎：three.js r170 + Vite 打包，无商业引擎。所有数据均从实际下载的代码与模型中提取，推断处标注【推断】。




x →
z ↓
-160 -140 -120 -100 -80 -60 -40 -20 0 20 40 60 80 100 120 140 160
Altar of Dead Stars 死星祭坛(BOSS) (-62, 115)
Prism Sanctum 棱镜圣所(最终BOSS) (0, 90)
Echo Den 回声巢穴(BOSS) (120, 90)
Shattered Obelisk 破碎方尖碑 (120, 60)
Warden's Perch 守望者栖所(BOSS) (-90, 30)
Arena 竞技场 (-70, 73)
Crystal Hearden 水晶花园 (-34, 48)
Prism Gate 棱镜之门(传送) (0, 0)
Fallen Citadel 陷落城堡 (-110, 44)
Ashen Spire 灰烬尖塔 (-27, 24)
Cinder Hollow 余烬洼地(BOSS) (34, 38)
Hunters' Camp 猎人营地 (-16, 22)
Ward Gate 守卫之门(传送) (0, -2)
Breach Pillar 裂隙之柱 (108, 0)
Thornwall Hollow 荆棘洼地(BOSS) (-45, 0)
Scavenger Camp 拾荒者营地 (99, 51)
Mirror Pond 镜池 (-38, 60)
Sunken Shrine 沉没神龛 (90, 72)
Hearthstone Keep 炉石要塞(安全基地) (0, 88)
Glass Lake 玻璃湖 (110, 82)
Old Watchtower 旧瞭望塔 (-85, 78)
Wizard merchant 巫师商人 (3,71)
Fallen Camp 堕落者营地 (-105, 90)
Void Pit 虚空深坑(BOSS) (12, 142)
WILDKEEP 世界地图 (seed 20260404)
俯视图，北(−Z)在上，1 px = 0.250 世界单位
地形：MA(x,z) 高度场 + 西北光照阴影
白框 |x|,|z|=140 世界边界(山墙)
粉框 |x|,|z|=96 内陆/裂隙界线
橙线 Br(x) 草甸↔灰烬林 屏障
青线 Er(x) 灰烬林↔水晶荒原 屏障
红线 裂隙墙 gp() (围绕圣所缺口)
浅标 路径 iu()<1.5
Emerald Meadow 翡翠草甸
Ashen Woods 灰烬林
Crystal Wastes 水晶荒原
Shattered Lands 破碎之地(裂隙)
POI 类型
keep
camp
lair
waystone
ruin
landmark
arena
cache 宝箱 (gr)
资源节点 (TM 生成, 648 个)
tree
ash_tree
boulder
ore_vein
crystal_cluster
prism_spire
aether_bloom
ember_vent
berry_bush
herb_patch
sprout
branch
loose_stone
fiber_tuft

WILDKEEP：Frontière des Créatures 技术栈分析

> 分析对象：`/workspace/work/wk/`（index.html、`assets/index-R4e691f4e.js` 共 1.80 MB／5523 行，美化后为 `game.pretty.js` 共 49,771 行，另有 32 个 GLB、1 张图集、2 个语言包）。 文中行号均指 **`game.pretty.js`**。标注 【推断】 的内容由命名、结构或常见做法推出，代码中没有直接证据。

---

0. 结论速览

项目	结论
引擎	不用游戏引擎，直接基于 three.js r170（`const Bl = "170"`，L400），用 Vite 打包为单个 ES module
渲染	WebGLRenderer + ACES 色调映射 + sRGB 输出 + PCFSoft 阴影；材质基于 MeshStandardMaterial，用 `onBeforeCompile` 注入轮廓光（rim）、风动、遮挡抖动淡出、自发光；水面和天空为自写 ShaderMaterial
后处理	EffectComposer → RenderPass → UnrealBloomPass → 自定义调色 ShaderPass（暗角、饱和度、冷暖色、受伤红边）→ OutputPass
资源	32 个 Quaternius 风格的蒙皮怪物 GLB（glTF-Transform v4.5.1 + KHR_mesh_quantization，GLB 内无贴图），共用 1024^2 调色板图集；主角是 base64 内嵌的 FBX（Kenney 风格 `characterMedium`）和 4 张内嵌 PNG 皮肤；场景、植被、道具全部程序化生成
画质系统	`QUALITY` 模块：设备打分、Worker 内 GPU 基准测试、LOW/MEDIUM/HIGH/ULTRA 四档 × 桌面/移动、7 级逐级降级、动态分辨率、移动端 FPS 上限（60/45/30）
音频	全部 WebAudio 实时合成，没有任何音频文件：6 套程序化配乐、约 35 种 SFX、生物叫声、环境风声／蟋蟀／鸟鸣，另有卷积混响和按优先级分配的发声数上限
平台	Y8 minimal SDK 2.0（插页广告 + 激励视频），排行榜和提交分数为空实现
多语言	源字符串是法语；en/es 字典各 1257 条，懒加载；用 MutationObserver 实时翻译 DOM
构建痕迹	原始 bundle 中混入了大段未压缩、带法语注释的代码（`const QUALITY`、`function wkAddIp`、`// modèles indispensables…`）→ 【推断】构建后经过手工补丁或合并

---

1. three.js 版本、打包模块与 Y8 SDK

1.1 版本

```
400: const Bl = "170",          // REVISION
405:     Kd = 2,                 // PCFSoftShadowMap
451:     El = 4,                 // ACESFilmicToneMapping
547:     ye = "srgb",            // SRGBColorSpace
```

1.2 打包进 bundle 的 three.js 核心与 addons（行号为大致起点）

模块	位置	证据或用途
three 核心（WebGLRenderer `tw`）	1–~19300；`class tw` L14776	`new tw({canvas, antialias, powerPreference:"high-performance"})` L45604
`InstancedMesh`（`Qf`）	L15885	植被和资源节点实例化（L39399）
`AnimationMixer`（`jf`）	L19378	怪物和主角动画
BufferGeometryUtils：`mergeGeometries(v5)`、`mergeVertices(qf)`	L19703 / L19793	程序化模型合并
GLTFLoader（`b5`）	L19883–~21500	加载怪物 GLB。扩展表完整（Draco、Meshopt、KTX2/basisu、webp/avif、quantization、各类 KHR_materials…，L20043 起），但代码没

模块	位置	证据或用途
		有注册 DRACOLoader/KTX2Loader/MeshoptDecoder，实际只用到 KHR_mesh_quantization
SkeletonUtils.clone（Ap）	L21514	克隆蒙皮场景，供同一怪物多次实例化
fflate（inflate）	~L30870	FBXLoader 的依赖（解压 FBX 中的 zlib 数组）
NURBSCurve / NURBSUtils	~L31104–31242	FBXLoader 的依赖
FBXLoader（e9）	L31246–~32780	解析内嵌的主角 FBX
CopyShader	L35294	后处理
Pass / FullScreenQuad（Tr/du）	L35330 / L35347	
ShaderPass（_p）	L35364	调色 Pass
EffectComposer（X9）	L35401	
RenderPass（Z9）	L35471	
LuminosityHighPassShader + UnrealBloomPass（br）	~L35500 / L35535	
OutputShader + OutputPass（j9）	L35714 / L35793	在后处理链末端做色调映射和色彩空间转换

没有打包 OrbitControls、three 的 Sky/Water 示例、CSS2DRenderer 或任何物理库。天空、水面、相机、碰撞都是自己写的。

1.3 Y8 SDK 集成（**yn** 对象，L46–160）

- index.html 以异步方式引入 `https://cdn.y8.com/minimal-sdk/2-0/y8.min.js`。
- 初始化：

```
56: dn = window.y8.sdk();
57: dn.init({ appId: "6ac9de676da5b1f7cb9fa002", autoLogin: !0 },
           { gameId: "287019", preloadAdBreaks: "auto", sound: "on" });
```

只在 `http(s):` 协议下初始化，3 秒超时，失败时静默降级（`Hi=false`）。

- 插页广告 `midgame`：`showAd({type:"next", name:"respawn"})`，两次之间至少间隔 180 秒（`Date.now()-h0 < 18e4`），只在死亡复活时调用（L46747）。
- 激励视频 `rewarded`：`showAd({type:"reward", name:"gold-boost"})`，看完奖励 +150 金币（L46727）。
- 播放广告时通过 `muteAds` 给游戏静音，并设置 `window.__y8Ad`。广告有 180 秒看门狗超时。
- `submitScore()` 返回 false，`leaderboard` 为 false，`loading`、`setPlaying`、`happy` 都是空函数 → 排行榜功能没有启用。
- L31–45 有一个 AES-GCM 加密函数 `em(s, key)`，同时定义了空字符串常量 `vu = ""`，二者都没有被调用。【推断】 这是分数加密提交的残留代码，目前已停用。

启动流程（L49735 起）：`await om()`（加载语言包）→ `um(document)`（启动 DOM 翻译）→ `await yn.init()` → 菜单。

---

2. 资源管线

2.1 GLB 怪物模型（实际打包了 32 个；任务说明里写的是 33 个，以磁盘为准）

所有 GLB 的共同点（解析各文件 JSON 头得到）：由 glTF-Transform v4.5.1 输出，`extensionsUsed = ["KHR_mesh_quantization"]`；POSITION 和 NORMAL 为 int16 归一化，TEXCOORD_0 为 uint16 归一化；不含 images/textures；材质名均为 `Atlas`（只有 flying_Dragon 叫 `Material`），双面渲染。运行时游戏会把材质整体替换成共享的图集材质（见 2.3），所以文件里的材质名没有影响。32 个文件共 6.01 MB。

文件（去 hash）	三角面	顶点	骨骼	动画片段
big_Alien	7676	4227	43	HitReact, Idle, Punch, Walk
big_BlueDemon	5800	3324	43	同上
big_Cactoro	6398	3866	43	同上
big_Demon	6712	4261	86	同上
big_Dino	5414	3266	43	同上
big_Frog	5016	2763	43	同上
big_MushroomKing	6044	3499	86	同上
big_Orc	7344	4412	86	同上
big_Yeti	6094	3415	43	同上
blob_Alien	2616	1485	8	Bite_Front, HitRecieve, Idle, Walk
blob_Cat	1772	1283	4	同上
blob_Chicken	2904	1692	4	同上
blob_Dog	1392	843	4	同上
blob_GreenSpikyBlob	4888	2858	8	同上
blob_Mushnub	1248	721	4	同上
blob_Mushnub_Evolved	3072	2079	12	同上
blob_PinkBlob	1030	587	8	同上
blob_Wizard	1674	1069	8	同上
blob_Yeti	2136	1473	4	同上
flying_Alpaking_Evolved	4684	3128	13	Fast_Flying, Flying_Idle, Headbutt, HitReact
flying_Armabee	2280	1260	13	同上
flying_Armabee_Evolved	3200	2022	13	同上
flying_Dragon	4562	2870	13	同上
flying_Dragon_Evolved	7440	4437	46	同上
flying_Ghost	3192	1837	38	同上
flying_Ghost_Skull	3146	2408	38	同上
flying_Glub	2528	1394	13	同上
flying_Glub_Evolved	4452	2426	13	同上
flying_Goleling_Evolved	5864	3688	13	同上
flying_Hywirl	3296	1759	38	同上
flying_Pigeon	3512	2139	13	同上
flying_Squidle	4880	2943	43	同上

面数为 1.0k–7.7k；三类模型各自共用一套四个动作，由 `iM` 表（L22175）按类别用正则匹配片段名：

```
blob:   { idle:/^Idle$/,        walk:/^Walk$/,        atk:/^Bite_Front$/, hit:/^HitRecieve$/ },
big:    { idle:/^Idle$/,        walk:/^Walk$/,        atk:/^Punch$/,      hit:/^HitReact$/ },
flying: { idle:/^Flying_Idle$/, walk:/^Fast_Flying$/, atk:/^Headbutt$/,   hit:/^HitReact$/ }
```

物种到模型的映射在 `ip` 表（L21990–22170），共 40 个物种，包括 `tint` 着色变体，例如 `blightbrute: {src:"big/Orc", tint:12099792}`，以及精英怪 `elite_*`。同一个 GLB 会被多个物种复用并用 tint 染色。运行时按目标身高 `h` 等比缩放模型，飞行类再加上 `hover` 悬浮高度。

预览图：`/workspace/work/wk/report/models_sheet_s.jpg`（32 个模型，带标注）和 `report/thumbs/*.png`（每个模型 320×320）。这些是离线软光栅渲染的 bind pose（T 姿势）静态图，颜色按三角形 UV 中心从图集取样，再加简单的 Lambert 明暗。它不是游戏内的实际画面，不含 rim 光和后处理。部分飞行模型远侧的翅膀被身体挡住。

2.2 来源判断：Quaternius「Ultimate Monsters」(CC0) — 【推断，可信度高】

- 目录分为 `big/`、`blob/`、`flying/` 三类，和该包的 Big / Blob / Flying 分类一致。
- 名称完全对应该包内的怪物：Mushnub(_Evolved)、GreenSpikyBlob、PinkBlob、Glub、Armabee、Alpaking、Goleling、Hywirl、Squidle、Cactoro、MushroomKing、BlueDemon、Ghost_Skull、Birb、Bunny、Fish、Tribal 等。
- 动画名有 Quaternius 的特征：`HitRecieve`（原包里就拼错了）、`Bite_Front`、`Fast_Flying`、`Flying_Idle`、`Headbutt`。
- 材质名都是 `Atlas`，配一张纯色块调色板贴图，这是 Quaternius 低模的典型做法。
- 原包每个怪物带有 Death、Run、Jump 等十几个动作，这里每个模型只剩 4 个片段 → 【推断】 开发者用 glTF-Transform 裁掉了不用的动画（`prune`），并做了量化（`quantize`），以减小体积。
- 参考：https://quaternius.com/packs/ultimatemonsters.html（50 个模型，许可证 CC0）；https://poly.pizza/bundle/Ultimate-Monsters-Bundle-5oyGWAmOB6（列出了 Birb、Fish、Tribal、Glub、Armabee 等）。

2.3 共享纹理图集 **`atlas-Bw1fvvpJ.png`**

- 1024×1024，RGB，只有 50 种颜色。由 32×32 像素的纯色方块组成：最左两列为全高的深灰（#1e1f23 左右）和中灰（#8b8f9e 左右），可能是眼睛、牙齿等公用色；其右有 30 行，每行 1–11 个色块，右侧和底部大面积为白色（未使用）。
- 【推断】 glTF-Transform 合并贴图时，把每个模型的调色板排成一行（30 行，对应 32 个模型中大部分；部分模型可能共用一行）。
- 加载与降采样（L22248–22249）：

```
as = QUALITY.render.shrink(await s, "atlas"), as.flipY = !1, as.colorSpace = ye,
as.anisotropy = QUALITY.texAniso(as)
```

`textureSize.atlas` 在 LOW 档为 256，MEDIUM 为 512，HIGH/ULTRA 为 1024，用 canvas 高质量降采样（L45301）。因为是纯色块，降采样几乎看不出差别。

- 每种 tint 只创建一个共享材质（od Map，L22266）：`MeshStandardMaterial({map: atlas, roughness:.8, color: tint})`，再套上 `c0` rim 光补丁。
- 放大预览：`/workspace/work/wk/report/atlas_preview.png`（最近邻放大到 2048×2048）。

2.4 6 个被引用但未打包的 GLB

```
22225: "./glb/big_Birb.glb":    "" + new URL("big_Birb.glb", import.meta.url).href,
22226: "./glb/big_Bunny.glb":   "" + new URL("big_Bunny.glb", import.meta.url).href,
...    big_Fish / big_Tribal / blob_Cactoro / flying_Tribal（同样没有 hash）
```

- 其余 32 个条目都指向 L362–393 带 hash 的资源常量（如 `hm = new URL("big_Alien-FxvvHkAq.glb", …)`），只有这 6 个文件名不带 hash，说明 Vite 构建时没有处理（发出）这些文件。
- 使用这 6 个模型的物种，是在一个未压缩的补丁函数里加入的（L44154）：

```
function wkAddIp(ip) { Object.assign(ip, {
  carrotail:{src:"big/Bunny"}, spikebeak:{src:"big/Birb"}, pricklehat:{src:"blob/Cactoro"},
  blightfish:{src:"big/Fish"}, blighttribal:{src:"big/Tribal"}, blightmask:{src:"flying/Tribal",
       hover:1.2} }); }
const WK_BASE = { blightfish:"blightcrawler", blighttribal:"blightspitter", blightmask:"blightbat" };
```

`WK_BASE` 让这 3 个新敌人沿用已有敌人的技能和行为配置（`wkEspec`）。

- 【推断】为什么没打包：这些条目和新物种是构建之后手工补进 bundle 的（原 bundle 中保留了可读标识符和法语注释，可作旁证），对应的 GLB 从未进入构建管线，也没有复制到 `assets/`。另一种可能是开发者故意删掉，用来控制体积或许可范围，但代码里没有证据支持这一点。
- 回退链路（代码证据）：
  1. `oM = async s => (await fetch(s)).arrayBuffer()`（L22237）不检查 `res.ok`。404 时拿到的是错误页的内容。
  2. `GLTFLoader.parse` 解析失败 → `console.warn("monster parse failed")`，结果为 `r(!1)` → `QUALITY.assets` 把该模型加入 `bad` 集合（L44310 起）。
  3. `A.ready(id)`：如果模型在 `bad` 中就返回 true，表示“允许生成”，见注释 L44407：« si le modèle est introuvable ou si le chargement a échoué, on retombe sur la version procédurale d'origine »（找不到模型或加载失败时，退回原来的程序化版本）。
  4. `l0(species)` 找不到解析结果时返回 `null`（L22265），调用方随即改用程序化拼装的生物（`Gt()` 骨架，L21905 起，用球体、胶囊、锥体等基础几何体拼出身体、腿、翅膀、尾巴，并由代码驱动行走、眨眼、呼吸、扑翼动画）。

5. 如果整个 GLB 管线异常（例如图集加载失败），设置 `QUALITY.assets.failed = true`，并输出 "monsters unavailable, using procedural creatures"（L22253），所有物种都改用程序化模型。

- 结果：这 6 个物种在游戏中以程序化外观出现，每次会多发出一个 404 请求。

2.5 内嵌的 base64 资源

常量	行	内容	用途
`d9`	L32784	222,952 字符的 base64 → 167,212 B 二进制 FBX（`Kaydara FBX Binary`，由 Blender (stable FBX IO) 2.79 导出，模型名 **`characterMedium`**）	主角模型：`new e9().parse(b9(d9))`（L33588），缩放 `1.8/376`
`f9`	L32785–33317	3 段预烘焙的 JSON 动画：`idle`(1.067 s)、`run`(0.667 s)、`jump`(0.5 s)，按 30 fps 采样，骨骼名为 Hips、Spine、Chest、UpperChest、⋯、LeftHandIndex1-3	用 `AnimationClip.parse` 读取（L33598）
`Lp[0..3]`	L33318	4 张 1024×1024 PNG（41.7/35.0/40.1/51.6 KB）	主角皮肤；`Ed(a)` 中 `` `a = ``
`b9`	L33549	`atob` → `ArrayBuffer` 解码工具	
`Bd`	L33541	`tint_forest:110, tint_dusk:230, tint_ember:35, tint_ocean:170`	用 `v9()` 在 canvas 上做 HSV 色相旋转来给服装换色；跳过低饱和像素和肤色像素（色相 6–42° 且饱和度 < 0.72）

- 已导出到 `/workspace/work/wk/report/embedded/`：`hero.fbx`、`hero_skin_0..3.png`、`hero_skins_sheet.png`。
- 【推断，可信度中高】 主角来自 Kenney「Animated Characters」系列（CC0）。依据：模型名 `characterMedium`、Blender 2.79 FBX 导出、恰好 idle/run/jump 三个动作，以及皮肤贴图的平涂风格和 UV 布局（头、身体、鞋子分块），都与该系列吻合。代码里没有写明出处。
- 主角模型加载失败时有回退：`Cd()` 捕获异常，改用程序化主角 `Rx()`（L33572）。
- 其他小型内嵌资源：favicon（SVG data URI）、`Kl()` 用 canvas 生成的径向渐变粒子和云朵贴图（L21898）。

2.6 加载调度（**`QUALITY.assets`**，L44198）

- 关键集合 CRIT：`PinkBlob, Pigeon, big/Alien, big/Dino, blob/Wizard`，进入游戏前必须加载完（初始宠物、鸟、第一个敌人、用于预热 shader 的模型、商人）。启动进度条通过 `window.__wkBoot` 汇报。
- 其余模型在后台按“初始生物群系 → 敌人 → 后续生物群系 → 精英”的顺序加载：同时预取 2 个，解析严格串行，利用 `requestIdleCallback` 的空闲时间，战斗或 Boss 战期间推迟解析，标签页隐藏时暂停。各画质档在两个模型之间插入不同间隔（gap 20–140 ms）。
- 显存回收：长时间未使用的物种在 `free` 秒后 `dispose` 掉几何体（LOW 档 300 s，MEDIUM 480 s，HIGH 和 ULTRA 桌面端不回收）；`QUALITY.mem` 释放骨骼 DataTexture（L44716，注释写不释放的话“约每分钟泄漏 28 张纹理”）。

- 预热：`warmMonsters()` 先把 4 个模型放到相机下方 60 m 处，执行 `renderer.compileAsync`（L45670），避免第一次遇敌时因编译 shader 卡顿。

---

3. 渲染

3.1 Renderer 设置（L45603–45606）

```
this.renderer = new tw({ canvas, antialias: !$p && A < 2, powerPreference: "high-performance" }),
this.renderer.outputColorSpace = ye,          // sRGB
this.renderer.toneMapping = El,               // ACESFilmic
this.renderer.toneMappingExposure = 1.05,
this.renderer.shadowMap.type = Kd;            // PCFSoftShadowMap
```

只在非触屏且 DPR < 2 时开启 MSAA。WebGL 上下文丢失时提示用户；上下文恢复时写入 `wk_forceLow` 后重载页面，下次以 LOW 画质启动（L45636、L49747）。

3.2 统一的「风格化 PBR」补丁 `c0()`（L21794–21815）

所有程序化物体、怪物、主角和地形都使用 `MeshStandardMaterial`（多为顶点色），并通过 `onBeforeCompile` 注入以下效果，全局 uniform 放在 `Kt`（L21535）：`uTime, uCam, uPly, uWind, uRimCol(0x9fc4ff), uRimK, uFade, uSunDir`。每帧在 L45892 更新。

- 轮廓光（rim / fresnel），叠加在 `opaque_fragment` 之前：

```
float rimF = pow(1.0 - clamp(dot(normalize(normal), normalize(vViewPosition)),0.,1.), 2.6);
outgoingLight += uRimCol * rimF * RIM * uRimK * (0.35 + 0.65*(N·up*0.5+0.5)) * diffuseColor.rgb;
```

强度按材质设定（如 `bark .25、foliage .55、crystal 1.0`），以常量形式写进 shader，并用 `customProgramCacheKey` 区分着色器程序。效果偏卡通，但不是分段的 toon ramp，光照仍是 PBR。

- 风动（wind）：顶点属性 `aw` 是 CPU 端按高度计算的权重（`Sn()` 中 `w * pow(Z,1.3)`），支持 `USE_INSTANCING` 下的世界坐标：

```
float ph = _w.x*0.31 + _w.z*0.23; float sw = sin(uTime*1.5+ph) + 0.5*sin(uTime*2.7+ph*1.9);
transformed.x += sw*aw*0.10*uWind; transformed.z += cos(uTime*1.2+ph*1.3)*aw*0.07*uWind;
```

- 遮挡淡出（fade）：树叶和树皮如果位于相机和玩家之间（以 2.4 m 为半径的圆柱内），用 Interleaved Gradient Noise 做抖动 discard，透视出玩家，同时避免半透明排序问题（L21810）。
- 自发光（emit）：`totalEmissiveRadiance += diffuseColor.rgb * k`，用于 glow 和 crystal 材质。
- 材质预设 `FA`（L21816）：`vcol, foliage(wind+fade), bark(wind+fade), vcolGlow, crystal, metal, soft, cloth`。
- 程序化建模工具：`M`（L21778）提供基础体缓存，包括 box、cyl、cone、sph、ico、oct、tor、cap、噪声扰动球 `blob`、车削体 `lathe`、叶片 `leaf`、多

层伞形 `tier`、Catmull-Rom 管道 `tube`。`Sn()`（L21595）把多个部件合并成一个几何体，并烘焙顶点色：高度渐变 `c→c2`、伪 AO（`ao .32`）、半球光（`hemi .22`）、3D 值噪声抖动（`jitter .07`）。整个世界没有贴图，所有颜色都在顶点里。

3.3 地形（L41040–41185）

- 网格分辨率由画质档决定（`drawDistance.terrain` 为 110–320 段），在树、岩石周围烘焙暗色 AO 斑块。
- 片元中加入程序化细节：多层值噪声做亮度扰动、大尺度冷暖色变化、条纹（`st`），细节在 30–120 m 范围内逐渐淡出。
- 整块地形切成 8×8 = 64 个子网格，共用顶点缓冲、各自使用独立的索引和包围球，用于视锥剔除。

3.4 水面（**`zy()`**，L41186–41285）

自写 `ShaderMaterial`，透明，不写深度，合并了 `UniformsLib.fog`：

- 顶点：两组正弦波起伏，幅度按水深（`aDepth`，由地形高度计算）衰减。
- 片元：两层滚动值噪声扰动法线 → 浅水/深水颜色插值 → 菲涅尔混入天空色 → Blinn 高光（`pow 120`）→ 用岸边深度加噪声生成泡沫 → `pond4` 带额外的“水晶”色调 → 透明度随深度变化。片元中自行包含 `tonemapping/colorspace/fog` 片段。

3.5 天空、光照、雾与昼夜（**`Hy`**，L41380–41530；关键帧 **`es`**，L41286）

- 天空：半径 380 的反面球体加自写 shader，包括顶部到地平线的渐变、太阳光晕（`pow 6` 与 `pow 64` 两层）、地平线辉光，以及两层程序化山脊剪影（1D 值噪声 `ridge(az)`）。没有后处理时（LOW 档），由 shader 自己做 gamma 校正（`enc` uniform）。
- 夜空：500 颗星（Points），太阳和月亮圆盘，14 片 Sprite 云（canvas 渐变贴图，随时间漂移）。
- 光照：一个 `HemisphereLight` 加一个投射阴影的 `DirectionalLight`。按 `es` 中 9 个时间关键帧（0、4.5、6、8、13、17、19、20.5、24 时）插值天空色、太阳色和强度、半球光颜色和强度、雾色。夜间把月光作为方向光。
- 阴影：shadow camera 为正交投影，范围 `±shadowExtent`（32–48 m），`bias -6e-4`，`normalBias .05`，贴图 1024–4096。阴影相机的目标位置按 texel 大小取整对齐（`this.snap`），防止阴影边缘闪烁。
- 雾：线性 `Fog`，起止距离由画质档设定（如 HIGH 为 100–250，移动端 LOW 为 40–100）。雾色随时间关键帧变化，并按生物群系染色（woods 带一点绿，wastes 和 rift 带紫红；`tintCol` 按 0.22–0.42 的比例混入）。

3.6 后处理（**`K9`**，L35843–35870；调色 shader **`q9`**，L35815）

```
35852: this.composer = new X9(A, l),                // RT: HalfFloat（支持时），HIGH 档 MSAA×4
       this.composer.addPass(new Z9(e, t)),         // RenderPass
       this.bloom = new br(res, r==="high" ? .38 : .3, .55, .92),  // UnrealBloom 强度/半径/阈值
```

```
this.composer.addPass(this.bloom),
this.grade = new _p(q9), this.composer.addPass(this.grade), // 调色
this.composer.addPass(new j9)                // OutputPass
```

- 调色 shader：饱和度（`sat .98`）、按亮度混合冷暖色（暗部 `cool(.93,.98,1.06)`，亮部 `warm(1.04,1,.94)`）、暗角（`vig`）、受伤时屏幕边缘泛红（`hurt`）。
- 由画质档的 `postProcessing` 决定：LOW 为 `"off"`（直接渲染、无 bloom）；MEDIUM 为 `"basic"`（无 MSAA，bloom .3）；HIGH 和 ULTRA 为 `"full"`（MSAA×4，bloom .38）。降级到第 4 级以上时 `bloom:0`，关闭泛光；`effects.bloomRes` 可以降低 bloom 的分辨率。
- 粒子：两套 `Points`，自写 shader（L37672），分为 soft（普通混合）和 glow（加法混合），点大小随距离衰减，有上限。另有环境漂浮光点 `motes`，数量随画质 40–200。

3.7 画质档位（**`QUALITY.PROFILES`**，L42534）

	LOW	MEDIUM	HIGH	ULTRA
pixelRatio 上限	1	1.5	2	2
阴影 / 贴图大小	关	1024	2048	4096
阴影投射者	none	important	most	all
粒子数	700	1200	2000	3000
植被密度	0.25	0.6	1.0	1.25
场景距离 near/far	26/110	34/160	46/210	54/250
雾	60–150	80–190	100–250	120–300
同屏角色上限	19	27	35	40
模型变体上限	2	3	4	99
图集贴图大小	256	512	1024	1024
后处理	off	basic	full	full
发声数上限（桌面/移动）	10/8	16/12	28/16	40/40

各档都有 `mobile` 覆盖值（例如移动端 HIGH 实际接近桌面 MEDIUM：DPR 1.5、阴影 1024、basic 后处理、仅 important 投射阴影、阴影距离 28 m）。移动端永远不会进入 ULTRA。

自动选档（L42800–43060）：

1. 设备分类：根据 `pointer:coarse`、`hover:none`、触点数、UA、`userAgentData.mobile`、iPadOS 等信号投票，得出 desktop、tablet 或 mobile。
2. 打分：基础分（桌面 70、平板 46、手机 38），再根据 `deviceMemory`、CPU 核数、屏幕像素量和 GPU 字符串分类加减（L42850 `classifyGpu`：识别 Adreno、Mali、PowerVR、Apple、RTX/RX、Intel Iris Xe 和软件渲染器）。
3. 菜单显示后，在 Web Worker + OffscreenCanvas 中跑隐形基准：约 12 ms 的 CPU 浮点运算，再用 128^2 全屏 ALU pass 测 GPU，约 60 ms，结果折算成“十亿次三角函数运算/秒”（Gops/s），按 `8*log2(gops/ref)` 调整分数（L42913–43037）。整个过程不占用主线程。

4. 历史优先：localStorage.wk_qhist 保存实际跑出的档位和上限，有效期 30 天，按设备签名区分。

运行时自适应（pm，L43049–43262，每秒执行一次）：

- 用 实际 FPS / 目标 FPS 的比值 r 做滑动平均：2 秒均值低于 0.6（移动端 0.67）时大幅降级；3 秒均值低于 0.8 时降一级；移动端 5 秒均值低于 0.95 时轻微降级；连续 8 秒都不低于 0.95 时升级。
- CPU 瓶颈识别：CPU 帧时间超过帧预算的 70% 时，不再降分辨率，改为提高 level。
- 先调分辨率 scale（桌面最低 0.75，移动最低 0.55），再沿 7 级 LV 阶梯（L42724）逐级削减：装饰物密度和缩放、资源节点、特效、漂浮光点、远处动画频率（1→0.25）、阴影更新间隔（每 1/2/3 帧一次）、bloom、角色数。
- 阶梯用尽后，移动端把 FPS 上限从 60 降到 45 再到 30，然后降一个画质档并记住这个上限。升档非常保守：要连续 40 秒没有可再提升的项。有防反复机制：bounces 计数，upLock 最长到 90 秒。
- 切换画质档需要重建场景，只在不战斗、没有 Boss 时进行（tryPending）。
- URL 调试参数：?qnoadapt 关闭自适应，?qpm。

3.8 LOD、实例化与剔除

- 植被和资源节点（Wy/ry，L39360–39505）：每种 kind|variant 组合对应一组 InstancedMesh，分 LOD0、LOD1、LOD2 三级，由 me(n, builder, hasLod) 生成，每级是独立的程序化模型。repack() 按与焦点的距离分配：大于 near 用 LOD1，大于 near*2.3 用 LOD2，大于 far 或 cull 剔除；此外还剔除视线方向背后的实例（Vt）。每个实例带有颜色抖动（instanceColor 0.86–1.12）。
- 静态合并（RENDER.mergeStatic，L44974）：按 材质 + 阴影类别 + receiveShadow + 48 m 分块 + renderOrder 分桶，把物体烘焙为大网格，减少 draw call。道具按包围球半径分为 big、mid、small 三类，按画质档决定是否投射阴影（castClass）。屏障类物体预先合并出开启和关闭两个版本（barrierPosts）。
- 角色剔除：自写 6 平面视锥检测（sphereVisible）；12 m 内的角色、Boss 和正在闪烁的角色始终可见；只有 shadowRange（28–56 m）内的角色投射阴影。
- 逻辑 LOD（QUALITY.logic，L44741）：AI 按距离分为每帧、30 Hz、10 Hz、4 Hz 四档更新，画面外的角色只做最少计算；骨骼动画在 anim 距离外降到每 2、3、4 帧更新一次，但玩家、盟友、Boss 和正在战斗的敌人始终满频。
- 怪物蒙皮网格设置了 frustumCulled=false（避免蒙皮包围盒错误导致误剔除），改由上面的角色剔除负责。

3.9 移动端处理

- 触屏 UI（`$9`，L35871）：浮动摇杆加镜头拖动区；攻击、弹弓、跳跃、交互、捕捉、治疗、切换装置、锁定目标、冲刺等按钮；双指缩放（镜头距离 4–12.5，保存在 `wk_camzoom`）。`QUALITY.fix` 中对触屏做了大量优化：点击敌人锁定、自动冲向锁定目标、捕捉无需瞄准、点击判定阈值等（L43493 起）。
- 竖屏时 FOV 收窄到 66（桌面 74，横屏 62）。
- 省电：打开面板时，或 45 秒无操作且不在战斗中，降到 30 fps（`QUALITY.vis.low`）；根据 rAF 间隔估算屏幕刷新率（支持 60–240 Hz），把帧率限制到目标值。
- `resize` 和 `orientationchange` 去抖；后台标签页暂停音频和资源加载；从后台恢复时重置帧时间和性能统计。
- 默认画质：触屏从 LOW 起步，桌面从 HIGH 起步，基准测试完成后再调整。

---

4. 音频（**`Ay`** 类，L36040–36407；乐曲表 **`ny`**，L36439–36600）

完全由 WebAudio 实时合成，没有任何音频文件。

4.1 信号图

```
master ← { musicG ← musicBus（加卷积混响）, sfxG, ambG } → destination
```

- 混响：2.2 秒立体声脉冲响应，由 `random * (1-t)^2.6` 生成，经 ConvolverNode 以 0.32 的湿声比例返回音乐通道（L36079–36087）。
- 共享一段 2 秒白噪声缓冲，所有噪声类音效都从中取材。
- 环境声：风声用循环噪声经 420 Hz 低通，截止频率由 0.12 Hz 的 LFO 以 ±180 Hz 调制，音量随生物群系变化（wastes .13、woods .07、meadow .045）；蟋蟀用 4300 Hz 正弦波加 18 Hz 振幅调制，只在 meadow 的夜晚出现；鸟鸣是随机的正弦滑音对，只在白天的 meadow 出现。
- 基础合成单元：`tone(freq, dur, wave, vol, bus, delay, glideTo, attack)`（振荡器 + 包络 + 可选的指数滑音），以及 `noise(dur, vol, f0, f1, filterType, delay, Q)`（滤波噪声，截止频率做指数扫频）。

4.2 程序化配乐

- 用 90 ms 的 `setInterval` 做 look-ahead 调度（提前 0.4 秒排程），每小节分为 16 个十六分音符，和弦每小节换一次。
- 每小节在第一拍铺 pad：三和弦，三角波（高 2 个八度）加正弦波（高 1 个八度），慢起音。
- 旋律记谱用字符串，如 `"E5:4 G5:4 E5:2 D5:2 C5:4"`，由 `ey/ty/Xr`（L36408）解析成 MIDI，再通过 `bn = 440*2^((m-69)/12)` 转成频率。旋律只在第奇数轮循环出现，或设置了 `melAlways` 时每轮都出现；主旋律另叠加一个八度、音量 35% 的正弦泛音（实际倍频 2.003，带轻微失谐）。

- 合成乐器：`pluck`（三角波 + 八度正弦，短衰减）、`bass`（正弦 + 八度三角）、`kick`（120→42 Hz 正弦下滑）、`snare`（带通噪声 + 三角波）、`hat`（7.5 kHz 高通噪声）、`stab`（锯齿波和弦）。

模式	BPM	和弦进行	调性 / 音阶	主音色	节奏层
explore	92	C Am F G C Em F G	C 大调	sine	琶音拨弦；第二轮起加 kick 和 hat
menu	76	Am F C G	A 小调 / C 大调	triangle	慢速
night	64	Dm B♭ F C	D 小调；无旋律，用 D 小调五声音阶（`[0,3,5,7,10]`，从 D5 起）的随机钟声，由确定性哈希 `cc()` 生成	—	低音长音
combat	138	Em C G D	E 小调（旋律里有 F♯/C♯，借用 D 大调）	sawtooth	切分 kick、snare 在 2/4 拍、八分 hat、stab
boss	152	Dm B♭ Gm A	D 和声小调（大三和弦 A 作属和弦，旋律含 C♯）	sawtooth	更密的底鼓、加花军鼓
victory	112	C G Am F	C 大调	triangle	主旋律始终出现

切换逻辑 `currentMusic()`（L45940）：菜单 → 胜利 → Boss → 战斗 → 夜晚 → 探索。

4.3 音效与生物叫声

- 约 35 种 SFX（L36160–36330）全部参数化合成，例如：
  - `hit` = 带通噪声 + 180 Hz 方波向下滑音
  - `capture_ok` = C 大调琶音 523/659/784/1047/1319 Hz
  - `levelup` = 六音方波琶音
  - `boss_warn` = 110/116 Hz 锯齿波拍频三连
  - `footstep` 的滤波频率随地表变化（wastes 3 kHz）

  每次播放随机微调 ±6% 音高，并用 `gate()` 防止同一音效短时间重复触发。
- 生物叫声 `voice(type, pitch, state)`：分 flyer、heavy、plant、support、elemental、armored 等类型，按物种的 `pitch` 和状态（受伤 ×1.3，攻击 ×0.8）合成。
- 发声数管理（`QUALITY.audio`，L44468）：用 64 槽的环形缓冲记录每个声音的结束时间（基于音频时钟），统计当前仍在发声的数量。按优先级 0–4（Boss/捕捉/胜利 > 战斗 > UI > 生物 > 环境）给各级分配上限份额（`FR=[1.7,1.2,.85,.65,.45]`），超出就丢弃，保证重要音效不被挤掉。

---

5. 多语言（fr 为基础语言，en、es 为翻译）

- 游戏内所有源字符串（UI、物品、描述、toast、`aria-label`）都直接用法语写在代码里，index.html 也是 `lang="fr"`，加载文字为 “Chargement…”。
- 语言选项为 `Mu = [en, fr, es]`（L195）。从 `localStorage.wk_lang` 读取，没有保存过时默认 **`"en"`**（L212）。

- `om()`（L224）在选择 en 或 es 时动态 `import()` 对应字典（`en-AVvZq18N.js` 85 KB，`es-uGgwbeGv.js` 92 KB，各 1257 条，其中 126 条带 `{0}` 占位符），并修改 `document.title`。
- 翻译方式：
  - 精确匹配：先规范化空白再查表，也会查全大写版本。
  - 模板匹配：把带 `{n}` 的条目编译成正则，按字面部分长度从长到短排序后依次匹配，捕获的参数递归翻译；`{0} {1}` 这种相邻占位符会先合并。
  - 启发式拆分（`lm`）：按 · 分隔、表情前缀、`(n/m)` 计数后缀、数量前缀、句子边界拆开后逐段翻译。
  - 结果缓存，超过 6000 条时清空。
- 应用方式：`um(root)` 在 `#ui` 和 body 上挂 MutationObserver，监听新增节点、文本变化和 `title/placeholder/aria-label` 属性，实时翻译 DOM，游戏逻辑完全不用感知语言。代价是每次 UI 更新都要多做一次查表；匹配不到时直接显示法语原文。

---

6. 其他值得注意的点

- 构建后补丁：原始 bundle 有 5523 行，其中混有可读的 `const QUALITY = (() => {…})()`（L42470）、`QUALITY.fix/assets/audio/save/vis/mem/logic/render` 等模块，以及 `function wkAddIp`、`WK_BASE`、大量法语注释。注释里还有“replace l'ancien adaptRes”“comme avant”（“替换旧的 adaptRes”“和以前一样”）这类措辞。【推断】 开发者在 Vite 产物上直接追加或替换了一个性能优化层和新物种，没有走完整构建，这也解释了 2.4 节中 6 个 GLB 缺失、文件名没有 hash 的现象。
- 存档：`localStorage`（`wk_main` 等），4 秒去抖，30 秒（有变化或移动超过 6 m）／120 秒兜底自动保存，在 `pagehide` 和标签页隐藏时也会保存；离线挂机收益最多按 30 分钟、半速结算。
- 世界生成使用固定种子的 PRNG（`ve(seed)`、`TM(20260404)`），每次生成的布局可复现。

---

7. 产出文件

路径	内容
`/workspace/work/wk/report/tech.md`	本报告
`/workspace/work/wk/report/atlas_preview.png`	图集最近邻放大到 2048^2
`/workspace/work/wk/report/models_sheet_s.jpg`	32 个 GLB 的总览图（1920×1186，标注面数、骨骼数、动画片段数）
`/workspace/work/wk/report/thumbs/*.png`	32 张单模型缩略图（320^2，bind pose）
`/workspace/work/wk/report/embedded/hero.fbx`、`hero_skin_0..3.png`、`hero_skins_sheet.png`	从 bundle 中解码出的主角 FBX 和皮肤
`/workspace/work/wk/report/render_thumbs.py`、`gltf_raw.py`、`make_sheet.py`	可复现的离线渲染脚本：用纯 numpy 解析 GLB，并正确处理 KHR_mesh_quantization 的归一化属性

局限：缩略图是 bind pose 的近似着色，不等同于游戏画面；文中关于美术来源（Quaternius、Kenney）和构建方式的判断均为推断；没有在浏览器中实际运行游戏验证运行时行为。

WILDKEEP（Frontière des Créatures）生物 / 敌人系统逆向分析

> 源：`/workspace/work/wk/game.pretty.js`（49,771 行，three.js r170 + Vite 打包，游戏代码约从 21,500 行开始）、`assets/en-AVvZq18N.js`（FR→EN 字符串表，1,257 条）、`glb_summary.json`。 所有 Lxxxxx 均为 `game.pretty.js` 的行号。标 【推断】 的为根据代码行为推出、无直接注释或字符串佐证的结论；其余均为代码原文直接给出。 附带机读数据：`/workspace/work/wk/report/creatures_data.json`（由 node 直接 eval 打包中的数据表导出：Xn 物种、uM 敌人、B9 敌人招式、ip 模型映射、qe 元素、He 稀有度、cd 克制表）。

0. 关键符号速查（压缩名 → 含义）

符号	位置	含义
`Xn`	L22334（+`wkAddSp` L43970 追加 3 种）	可捕获生物物种表（28 种）
`Ge`	L23187	Xn 的 id→def 字典
`uM / rp`	L23273 / L23905（+`wkAddEn` L44076 追加 3 种）	敌人（Corruption/Blight、精英、Boss）表（20 种）
`B9`	L33880	敌人招式表（specs）
`WK_BASE / wkEspec`	L44184 / L44189	新增 blight 敌人借用旧敌人招式
`y9`	L33819	生物 `ability` → 战斗 spec 转换
`qe / cd / ga`	L23200 / L23188 / L23197	元素显示表 / 克制环 / 克制倍率函数
`He / sp`	L23232 / L23270	稀有度表（潜力、刷新权重）/ 稀有度→捕获器档位
`ip / iM / sM`	L21978 / L22169 / L22190	物种→GLB 映射 / GLB 家族→动画剪辑正则 / GLB URL 表
`Gt`	L21906	程序化动画 rig（bob/腿/翅/眨眼…），GLB 模型也套在它里面
`l0`	L22264	GLB 实例化 + AnimationMixer 驱动
`Cp / Px`	L29379 / L29735	生物 / 敌人 rig 构建（先试 GLB，失败则程序化）
`oc`	L34174	战斗单位类（玩家队友、野生生物、敌人、精英、Boss 共用）
`S9 / R9 / z9`	L34705 / L34812 / L34964	野生 AI 状态机 / 队友 AI / Boss AI
`Ys`	L27896	伤害公式
`ux / hx / dx`	L27849 / L27857 / L27886	捕获概率 / 捕获合法性 / 捕获掷骰
`xt / WM / xp / _o`	L27254 / L27265 / L27286 / L27245	收服个体属性 / 创建个体 / 加经验 / 升级曲线
`px`	L27905	击杀敌人结算（掉落、XP、羁绊）
`QUALITY.fix`	L43493	后期补丁层（支援型普攻、低等级玩家仇恨缩减、首只伙伴减伤等）

1. 物种 / 敌人全表

1.1 结构与来源

- 可捕获生物（`kind="creature"`）：Xn 原始 25 种（L22334–23186）+ `wkAddSp` 追加 `carrotail / spikebeak / pricklehat`（L43970–44074）= 28

种。名称本身即为英文（FR 版与 EN 版同名；EN 表里没有这 3 个追加物种的条目，界面直接显示 id 同名字符串）。

- 敌人（`uM`，L23273–23904）：9 种普通 Blight + 7 精英 + 1 Boss；`wkAddEn` 追加 `blightfish / blighttribal / blightmask`（L44076–44153）= 20 种。
- NPC：`npc_merchant`（`blob/Wizard`，h=1.9）只是商人模型，不参与战斗（L42050）。
- 模型来源统一由 `ip`（L21978）决定，`wkAddIp`（L44155）再补 6 个新物种。GLB 实际 URL 在 `sM`（L22190）；其中 `big_Birb / big_Bunny / big_Fish / big_Tribal / blob_Cactoro / flying_Tribal` 通过 `new URL("…glb", import.meta.url)` 引用，不在本次提供的 assets 目录中（标 ⚠️）。其余 32 个 GLB 全部被引用，无闲置文件。
- 关于 **`QUALITY.fix.PROC`**（L43497）：列出 18 个“改回程序化模型”的物种，并有 `F.prune` 删除其 `ip` 条目——但全文件没有任何地方调用 **`F.prune`**（grep `prune` 仅命中定义处），所以在这份构建里这 18 种仍使用 GLB；程序化版只在 GLB 加载失败时作为后备（`lM()` catch → `ll.clear()`，L22258–22259；`QUALITY.assets.ready` 在模型未就绪时阻止该物种刷新，L44407–44414 注释“la version procédurale ne doit jamais apparaître par hasard”）。
- “Evolved / 变异 / 变体”：代码中不存在进化机制（`grep -i "evolv|mutat|shiny|variant"` 在游戏逻辑中零命中）。`*_Evolved` 只是 GLB 资源名，直接作为另一个物种的模型（例如 `sporelord = blob/Mushnub_Evolved`，`pyrelope/elite_cryptwing = flying/Dragon_Evolved`）。“变体”在本作中表现为：
  - Blight 变体：复用生物 GLB + `tint` 染色（`blightleaper`=Frog、`blightwisp`=Ghost、`blightthornling`=GreenSpikyBlob、`blightspore`=Mushnub、`blightdrake`=Dragon、`blightbrute`=Orc…）。染色在 `l0` 里作为材质 `color` 乘到图集贴图上（L22270–22279，按 tint 缓存材质 `od`）。
  - 新 Blight 借招：`WK_BASE = {blightfish:"blightcrawler", blighttribal:"blightspitter", blightmask:"blightbat"}`（L44184），`oc` 构造时 `this.specs = B9[r.id] ?? wkEspec(r.id, B9) ?? []`（L34277）——这 3 种没有自己的招式，深拷贝基础敌人的招式。
  - 精英 (elite)、Alpha（`elite_cinderhorn` 名为 “Alpha Cinderhorn”、`elite_blightalpha` 名为 “Alpha de la Corruption”——Alpha 只是命名，机制上同为 `kind:"elite"`）、Boss（`boss_aethon`，三阶段）。
  - 个体差异：收服个体有 `pot`（潜力）随机倍率，和羁绊星级（见 §3）。

1.2 稀有度表 **`He`**（L23232）

稀有度	FR / EN	order	potential（加到 pot）	spawn（栖息地刷新权重）	捕获器最低档 **`sp()`**
common	Commun / Common	0	0	1	1
uncommon	Peu commun / Uncommon	1	0.04	0.6	1
rare	Rare / Rare	2	0.08	0.3	1
epic	Épique / Epic	3	0.12	0.14	2
mythic	Mythic / Mythic	4	0.18	0.05	3

```
// L23270
function sp(s) { return s === "mythic" ? 3 : s === "epic" ? 2 : 1 }
```

1.3 可捕获生物全表（基础值 = 1 级、pot=1）

字段说明：spd 为移动速度（m/s）；capture 为基础捕获率；“模型”中 h 为 GLB 归一化到的高度（l0: scale = h / bboxHeight，L22286），再乘物种 scale（C p L29382）；hover 为悬浮高度（飞行类）。“程序化后备”列为 Cp 中的构建器（q t 四足 L28511 / ec 鸟/蛾 L28778 / Tx 蘑菇 L28885 / bd 岩甲 L28986 / Cx 浮游支援 L29091 / Sx 元素体 L29222；未列 case 的物种走 default 四足 L29716）。

#	id	名称 FR / EN	元素	稀有度	原型 archetype	性格	栖息群系	HP	ATK	DEF	SPD	capture	诱饵	捕获器最低档	模型 (ip)
1	fluffern	Fluffern / Fluffern	verdant 🌿	common	quad	docile	meadow	40	6	3	3.2	0.62	sweet	1	blob_PinkBlob.glb
2	twigwing	Twigwing / Twigwing	verdant 🌿	common	flyer	docile	meadow	30	5	2	4.4	0.58	sweet	1	flying_Pigeon.glb hover=0.8)
3	puddlepup	Puddlepup / Puddlepup	tide 💧	common	quad	neutral	meadow	44	7	3	3.6	0.55	sweet	1	blob_Dog.glb (h=(
4	sparkhopper	Sparkhopper / Sparkhopper	storm ⚡	common	quad	neutral	meadow	34	8	2	4.6	0.5	mineral	1	big_Frog.glb (h=1
5	pebblepaw	Pebblepaw / Pebblepaw	stone 🪨	common	armored	neutral	meadow	52	6	8	2.6	0.5	mineral	1	blob_Yeti.glb (h=(

#	id	名称 FR / EN	元素	稀有度	原型 archetype	性格	栖息群系	HP	ATK	DEF	SPD	capture	诱饵	捕获器最低档	模型 (ip)
	aw														
6	emberkit	Emberkit / Emberkit	ember 🔥	uncommon	slender	neutral	meadow, woods	38	10	3	4.8	0.45	savory	1	blob_Cat.glb (h=0
7	bloomcap	Bloomcap / Bloomcap	verdant 🌿	uncommon	plant	docile	meadow, woods	46	7	5	2.2	0.45	sweet	1	blob_Mushnub.glb
8	dewdrake	Dewdrake / Dewdrake	tide 💧	uncommon	support	docile	meadow	42	6	4	3.4	0.4	sweet	1	flying_Glub.glb (h= hover=0.8)
9	cindertusk	Cindertusk / Cindertusk	ember 🔥	uncommon	heavy	aggressive	woods	90	14	8	3.4	0.38	savory	1	big_Dino.glb (h=1
10	thornstalker	Thornstalker / Thornstalker	verdant 🌿	uncommon	slender	aggressive	woods	62	15	4	5.2	0.36	savory	1	blob_GreenSpikyB (h=0.95)
11	gloomwing	Gloomwing / Gloomwing	arcane 🔮	uncommon	flyer	aggressive	woods, wastes, rift	48	13	3	5	0.36	arcane	1	flying_Ghost.glb (hover=1)
12	ashm	Ashmaw / Ashmaw	ember 🔥	rare	slender	aggressive	woods, rift	80	19	5	5.6	0.28	savory	1	big_Demon.glb (h

#	id	名称 FR / EN	元素	稀有度	原型 archetype	性格	栖息群系	HP	ATK	DEF	SPD	capture	诱饵	捕获器最低档	模型 (ip)
	aw														
13	mossback	Mossback / Mossback	stone 🪨	rare	heavy	neutral	woods, rift	130	11	14	2.2	0.3	mineral	1	big_MushroomKin (h=1.45)
14	rainstrider	Rainstrider / Rainstrider	tide 💧	rare	slender	docile	woods	66	12	5	5.4	0.32	sweet	1	blob_Chicken.glb
15	shardhide	Shardhide / Shardhide	stone 🪨	rare	armored	aggressive	wastes, rift	120	20	16	3.2	0.26	mineral	1	big_Yeti.glb (h=1.
16	voltcrest	Voltcrest / Voltcrest	storm ⚡	epic	flyer	aggressive	wastes, rift	85	24	6	6.2	0.2	mineral	2	flying_Dragon.glb hover=1.3)
17	prismoth	Prismoth / Prismoth	arcane 🔮	epic	support	docile	wastes, rift	70	12	7	4.2	0.2	arcane	2	flying_Hywirl.glb (hover=1)
18	pyrelope	Pyrelope / Pyrelope	ember 🔥	epic	elemental	neutral	wastes, rift	95	26	8	5	0.18	savory	2	flying_Dragon_Evo (h=1.3, hover=1.1
19	tempes	Tempestail / Tempestail	storm ⚡	mythic	elemental	aggressive	wastes, rift	130	32	10	6	0.12	arcane	3	flying_Squidle.glb hover=1.2)

#	id	名称 FR / EN	元素	稀有度	原型 archetype	性格	栖息群系	HP	ATK	DEF	SPD	capture	诱饵	捕获器最低档	模型 (ip)
	tail														
20	aetherling	Aetherling / Aetherling	arcane 🔮	mythic	elemental	docile	wastes, rift	120	28	12	4.6	0.1	arcane	3	flying_Ghost_Skul (h=1.1, hover=1.1
21	sporelord	Sporelord / Sporelord	verdant 🍃	epic	plant	docile	rift	110	16	10	3.4	0.1	arcane	2	blob_Mushnub_Ev (h=1.3)
22	hexabuzz	Hexabuzz / Hexabuzz	storm ⚡	epic	flyer	neutral	rift	85	24	7	5.6	0.1	sweet	2	flying_Armabee_E (h=1.1, hover=1.2
23	maelglub	Maelglub / Maelglub	tide 💧	epic	flyer	docile	rift	100	18	9	4.4	0.1	sweet	2	flying_Glub_Evolve (h=1.2, hover=1)
24	golemurk	Golemurk / Golemurk	stone ⛰️	epic	heavy	neutral	rift	150	22	16	3.6	0.1	mineral	2	flying_Goleling_Ev (h=1.5, hover=0.9
25	aurelpaca	Aurelpaca / Aurelpaca	arcane 🔮	mythic	support	docile	（无；仅商人出售）	130	20	12	4.8	0.05	arcane	3	flying_Alpaking_Ev (h=1.5, hover=1)
26	carro	Carrotail / Carrotail	verdant 🍃	common	slender	docile	meadow	34	5	2	4.6	0.6	sweet	1	big_Bunny.glb (h= 内缺失

#	id	名称 FR / EN	元素	稀有度	原型 archetype	性格	栖息群系	HP	ATK	DEF	SPD	capture	诱饵	捕获器最低档	模型 (ip)
	tail														
27	spikebeak	Spikebeak / Spikebeak	storm ⚡	uncommon	slender	neutral	woods	52	15	6	4.2	0.4	savory	1	big_Birb.glb (h=1. 内缺失
28	pricklehat	Pricklehat / Pricklehat	stone 🪨	uncommon	plant	neutral	wastes	58	13	9	2.6	0.38	mineral	1	blob_Cactoro.glb ⚠️包内缺失

额外字段：`gloomwing.nocturnal = true`（L22688，全代码无读取处，死字段）；`aetherling.night = true`（只在“迁徙事件”和“发现系统”的候选过滤中用！`m.night` 排除它，L47996 / L42293）。

商人出售的生物（`Kp`，L41918；`buyCreature` L46700 → `Ac(state, sp, lv)`，个体同样随机 pot）：

物种	等级	价格（金币）
fluffern	3	150
puddlepup	3	160
twigwing	4	180
emberkit	5	320
mossback	8	900
prismoth	10	1500
sporelord	10	2400
hexabuzz	10	2600
maelglub	12	2800
golemurk	12	3000
aurelpaca	12	3400（`biomes: []`，野外不刷新，唯一来源）

1.4 生物被动 / 主动技能

主动技能 `ability` 经 `y9()`（L33819）转成战斗 spec：`bite→melee(windup .5, arc 1.8)`、`proj→proj(windup .65, min 2, speed 16)`、`bolt→bolt(.6, min 2)`、`quake→quake(.9, max=range*.95, radius=range)`、`dash→dash(.75, min 2, speed 15, width 1.8)`、`beam→beam(.9, min 2, width 1.3)`、`thorns→thorns(.6, count 3, arc .7, speed 15)`、`heal/buff→(.3)`；统一 `recover: .5`。

id	被动 FR → EN	被动说明 (FR)	技能 FR → EN	kind	power	cd	range	work	workPower	scale
fluffern	Atterrissage doux → Soft Landing	Subit 10 % de dégâts en moins tant que le groupe est en pleine santé.	Mordillage feuillu → Leaf Nip	bite	1	1.6	2	farm,training	1	0.9
twigwing	Traînée de pollen → Pollen Trail	Le compagnon regagne 1 PV toutes les 4 s.	Explosion de pollen → Pollen Burst	heal	14	7	9	healing,farm,workshop	0.9	0.8
puddlepup	Pelage humide → Damp Coat	Résiste un peu aux dégâts de Braise (-10 %).	Tir éclaboussant → Splash Shot	proj	1.1	2.4	10	farm,healing	1	0.9
sparkhopper	Charge statique → Static Charge	Chances de coup critique +4 %.	Bond électrique → Zap Hop	bolt	1.2	3	8	station	1.1	0.8
pebblepaw	Plaques rocheuses → Rocky Plating	Défense +10 %.	Roulé-boulé → Roll Out	dash	1.3	3.5	6	station,training	1	0.85
emberkit	Cœur chaleureux → Warm Heart	Les feux de camp proches brûlent plus fort : +20 % de charbon produit.	Crachat de cendres → Cinder Spit	proj	1.3	2.2	11	campfire	1.2	0.85

id	被动 FR → EN	被动说明 (FR)	技能 FR → EN	kind	power	cd	range	work	workPower	scale
bloomcap	Nuage de spores → Spore Cloud	Les ennemis proches infligent 5 % de dégâts en moins.	Volée d'épines → Thorn Volley	thorns	0.7	3.2	9	farm,healing	1.25	0.85
dewdrake	Rosée du matin → Morning Dew	Les cultures proches poussent 10 % plus vite.	Soin de rosée → Dew Mend	heal	22	8	10	farm,healing,workshop	1.2	0.85
cindertusk	Cuir épais → Thick Hide	Résistance au recul.	Charge de défenses → Tusk Charge	dash	1.6	4	7	campfire,training	1.3	1.15
thornstalker	Camouflage → Camouflage	Le premier coup inflige +25 % de dégâts.	Coup d'épines → Thorn Lash	bite	1.5	2.4	2.4	farm,training	1.1	1
gloomwing	Vision nocturne → Night Sight	Chances de coup critique +6 % la nuit.	Rayon crépusculaire → Dusk Beam	beam	1.2	3	10	workshop,training	1	1
ashmaw	Chasseur de meute → Pack Hunter	Attaque du groupe +8 % quand deux compagnons de Braise sont actifs.	Fente de flammes → Flame Lunge	dash	1.7	3.4	7	campfire	1.5	1.1
mossback	Rempart → Bulwark	Subit 15 % de dégâts en moins ; les alliés proches en subissent 8 % de moins.	Rempart moussu → Mossy Bulwark	buff	0.25	9	8	station,training,healing	1.4	1.3
rain	Voile de brume → Mist Veil	Quand il travaille à la base, tous les	Danse de la pluie → Rain Dance	heal	28	9	11	farm,healing	1.45	1.15

id	被动 FR → EN	被动说明 (FR)	技能 FR → EN	kind	power	cd	range	work	workPower	scale
strider		travailleurs se fatiguent 20 % moins vite.								
shardhide	Carapace de cristal → Crystal Carapace	Renvoie 10 % des dégâts de mêlée.	Séisme d'éclats → Shard Quake	quake	1.4	5	5	station	1.6	1.25
voltcrest	Surcharge → Overcharge	Chaque 4e attaque déclenche une foudre en chaîne.	Éclair en chaîne → Chain Bolt	bolt	1.6	3.2	12	station	1.8	1.2
prismoth	Réfraction → Refraction	Réduit les coûts en matériaux de l'atelier lorsqu'il y est affecté.	Égide prismatique → Prism Aegis	buff	0.3	8	10	workshop,advanced,healing	1.8	1.1
pyrelope	Aura ardente → Blazing Aura	Brûle légèrement les ennemis proches.	Anneau infernal → Inferno Ring	quake	1.8	6	6	campfire,advanced	1.9	1.25
tempestail	Œil du cyclone → Eye of the Storm	Les alliés de Foudre gagnent +15 % d'attaque.	Fouet de tonnerre → Thunderlash	bolt	2.2	3.6	13	station,advanced	2.4	1.4
aetherling	Lien d'éther → Aether Link	Octroie +10 % de productivité à tous les travailleurs tant qu'il est affecté.	Rayon d'éther → Aether Beam	beam	2	3.4	13	workshop,advanced,training	2.6	1.3
spor	Nuage de spores → Spore Cloud	Soigne légèrement tout le	Éruption de spores → Spore Burst	heal	26	8	10	farm,healing,workshop	2.2	1.2

id	被动 FR → EN	被动说明 (FR)	技能 FR → EN	kind	power	cd	range	work	workPower	scale
elord		groupe en continu.								
hexabuzz	Dard électrique → Electric Stinger	Chances de coup critique +8 %.	Salve de dards → Stinger Volley	bolt	1.6	3.2	11	station,training,workshop	2	1.2
maelglub	Marée apaisante → Soothing Tide	Le compagnon regagne 2 PV toutes les 4 s.	Vague de soin → Healing Wave	heal	30	8	10	healing,farm,station	2.1	1.2
golemurk	Peau de granit → Granite Skin	Subit 12 % de dégâts en moins.	Choc sismique → Seismic Shock	quake	1.7	5	6	workshop,advanced,training	2.4	1.3
aurelpaca	Toison dorée → Golden Fleece	Réduit les coûts en matériaux de l'atelier lorsqu'il y est affecté.	Bénédiction dorée → Golden Blessing	buff	0.4	9	12	workshop,advanced,healing,training	3	1.3
carrotail	Croquant vitaminé → Croquant vitaminé	Le compagnon regagne 1 PV toutes les 4 s.	Croque-carotte → Croque-carotte	heal	16	8	9	farm,healing	1.2	1
spikebeak	Plumes hérissées → Plumes hérissées	Défense +10 %.	Coup de bec → Coup de bec	bite	1.3	1.8	2.4	station,training	1.3	1
prickleh	Épines vives → Épines vives	Chances de coup critique +4 %.	Salve d'épines → Thorn Volley	thorns	1.1	2.4	10	farm,training	1.3	0.9

id	被动 FR → EN	被动说明 (FR)	技能 FR → EN	kind	power	cd	range	work	workPower	scale
at										

被动在代码中的真实实现情况（逐一 grep 物种 id 字面量核对；“未实现”= 除数据表/模型/商店外没有任何引用）：

物种	描述效果	实际代码
twigwing, carrotail	每 4 s 回 1 HP	✅ R9 L34819：`regenT>4 → hp+1`（仅作为队友时）
maelglub	每 4 s 回 2 HP	❌ 未实现
pebblepaw	防御 +10%	✅ `xt` L27260 `def ×1.1`；另外 `damageAlly` L46539 再 ×0.85 受伤（描述未提及的额外减伤）
spikebeak	防御 +10%	✅ `xt` L27260
sparkhopper, pricklehat	暴击 +4%	✅ `critChance` L34296
gloomwing	夜间暴击 +6%	⚠️ 代码在 L34296，但 `game.night` 每帧被强制设为 `false`（L45919 `this.night = !1`），永不生效
hexabuzz	暴击 +8%	❌
voltcrest	每第 4 次攻击连锁闪电	✅ `fire()` bolt 分支 L34617：`atkCount % 4 === 0` → 7 m 内弹射 1 个目标，power×0.6
tempestail	（描述：雷系队友 ATK +15%）	❌ 描述效果未实现；但代码让它每次 bolt 都连锁（L34617：`this.defId === "tempestail"`）
cindertusk	抗击退	✅ `knockResist = .6`（L34271）；mossback 同样 .6（描述写的是减伤）
mossback	自身 -15% 伤害、周围 -8%	❌（只有 knockResist）
emberkit	营火 +20% 木炭	✅ L27454 `md(s,i,"emberkit") ? 1.2 : 1`
dewdrake	农田 +10% 生长	✅ L27459 `+.1`
rainstrider	基地工人疲劳 -20%	✅ L27437 `n = .8`（全局，只要它在任意岗位）
aetherling	全体工人产能 +10%	✅ L27436 `Ax(1.1)`
prismoth, aurelpaca	工坊材料减免	⚠️ 无物种判断；工坊/高级工坊减免是通用的 `gd()`：Σ workPower×0.08，上限 30%（L27497–27500）
fluffern, puddlepup, bloomcap, thornstalker, ashmaw, shardhide, pyrelope, sporelord, golemurk	各种战斗被动	❌ 全部未实现

1.5 成长（每级）

```
// L27254 收服个体（队友/仓库）
function xt(s) {
  const A = Ge[s.sp], e = 1 + .085 * (s.lv - 1), t = LM(s);          // LM = 1 + .02*星级
  return {
    maxHp: Math.round(A.hp * e * s.pot),
    atk: +(A.atk * e * s.pot * t).toFixed(1),
    def: +(A.def * e * s.pot * t * ((s.sp === "pebblepaw" || s.sp === "spikebeak") ? 1.1 : 1)).toFixed(1),
    spd: A.spd }}
// L34266 野生生物（无 pot、无星级）
const a = 1 + .085 * (t - 1); this.maxHp = Math.round(i.hp * a), this.atk = i.atk * a, this.def = i.def * a
// L34275 敌人：只有 crawler/spitter/swooper 随等级成长，精英/Boss 固定 a=1
const a = r.kind === "crawler" || r.kind === "spitter" || r.kind === "swooper" ? 1 + .085 * (t - 1) : 1;
```

- 每级线性 +8.5% 基础值（HP/ATK/DEF 同步；SPD 不变）。Lv30 = 基础 ×3.465。

- 潜力 pot（L27268）：`pot = 0.92 + rand()*0.16 + He[rarity].potential`，保留 3 位小数。common ∈ [0.92,1.08]，mythic ∈ [1.10,1.26]。
- 羁绊星级（L27183–27212）：阈值 `ia = [10,30,65,110,170]`，0–5 星，ATK/DEF ×(1 + 0.02×星)。
- 工作产能也随等级：`ir()` = `workPower × (1 + .06(lv-1)) ×` 能量系数（L27363）。

野生生物各等级基础属性（HP/ATK/DEF，pot=1，由公式计算）：

id	Lv1 HP/ATK/DEF (野生)	Lv10	Lv20	Lv30	Lv30 收服个体 pot 区间 HP
fluffern	40/6.0/3.0	71/10.6/5.3	105/15.7/7.8	139/20.8/10.4	128–150
twigwing	30/5.0/2.0	53/8.8/3.5	78/13.1/5.2	104/17.3/6.9	96–112
puddlepup	44/7.0/3.0	78/12.4/5.3	115/18.3/7.8	152/24.3/10.4	140–165
sparkhopper	34/8.0/2.0	60/14.1/3.5	89/20.9/5.2	118/27.7/6.9	108–127
pebblepaw	52/6.0/8.0	92/10.6/14.1	136/15.7/20.9	180/20.8/27.7	166–195
emberkit	38/10.0/3.0	67/17.7/5.3	99/26.2/7.8	132/34.7/10.4	126–147
bloomcap	46/7.0/5.0	81/12.4/8.8	120/18.3/13.1	159/24.3/17.3	153–179
dewdrake	42/6.0/4.0	74/10.6/7.1	110/15.7/10.5	146/20.8/13.9	140–163
cindertusk	90/14.0/8.0	159/24.7/14.1	235/36.6/20.9	312/48.5/27.7	299–349
thornstalker	62/15.0/4.0	109/26.5/7.1	162/39.2/10.5	215/52.0/13.9	206–241
gloomwing	48/13.0/3.0	85/22.9/5.3	126/34.0/7.8	166/45.0/10.4	160–186
ashmaw	80/19.0/5.0	141/33.5/8.8	209/49.7/13.1	277/65.8/17.3	277–322
mossback	130/11.0/14.0	229/19.4/24.7	340/28.8/36.6	450/38.1/48.5	450–523
rainstrider	66/12.0/5.0	116/21.2/8.8	173/31.4/13.1	229/41.6/17.3	229–265
shardhide	120/20.0/16.0	212/35.3/28.2	314/52.3/41.8	416/69.3/55.4	416–482
voltcrest	85/24.0/6.0	150/42.4/10.6	222/62.8/15.7	295/83.2/20.8	306–353
prismoth	70/12.0/7.0	124/21.2/12.4	183/31.4/18.3	243/41.6/24.3	252–291
pyrelope	95/26.0/8.0	168/45.9/14.1	248/68.0/20.9	329/90.1/27.7	342–395
tempestail	130/32.0/10.0	229/56.5/17.7	340/83.7/26.2	450/110.9/34.7	495–568
aetherling	120/28.0/12.0	212/49.4/21.2	314/73.2/31.4	416/97.0/41.6	457–524
sporelord	110/16.0/10.0	194/28.2/17.7	288/41.8/26.2	381/55.4/34.7	396–457
hexabuzz	85/24.0/7.0	150/42.4/12.4	222/62.8/18.3	295/83.2/24.3	306–353
maelglub	100/18.0/9.0	177/31.8/15.9	262/47.1/23.5	347/62.4/31.2	360–416
golemurk	150/22.0/16.0	265/38.8/28.2	392/57.5/41.8	520/76.2/55.4	541–624
aurelpaca	130/20.0/12.0	229/35.3/21.2	340/52.3/31.4	450/69.3/41.6	495–568
carrotail	34/5.0/2.0	60/8.8/3.5	89/13.1/5.2	118/17.3/6.9	108–127
spikebeak	52/15.0/6.0	92/26.5/10.6	136/39.2/15.7	180/52.0/20.8	173–202
pricklehat	58/13.0/9.0	102/22.9/15.9	152/34.0/23.5	201/45.0/31.2	193–225

1.6 敌人全表（**uM** + **wkAddEn**）

* `range` 与 `windup` 两个字段在代码中从未被读取（`grep "edef\."` 只命中 `edef.cd`，L34523）；真实攻击距离与前摇来自下方 `B9` 招式表。`cd` 只以 `×0.5` 的形式加到每次出招后的冷却里。

#	id	名称 FR / EN	kind	元素	HP	ATK	DEF	SPD	aggro	leash	range*	windup*	cd	XP	掉落 item min-max@chance	scale	模型
1	bl	Blightcrawler / Blightcrawler	crawler	stone ⛰️	45	9	2	3.6	12	28	2.2	0.7	1.8	14	essence 1-1@0.6;	1	big_A

#	id	名称 FR / EN	kind	元素	HP	ATK	DEF	SPD	aggro	leash	range*	windup*	cd	XP	掉落 item min-max@chance	scale	模型
	lightcrawler														fiber 1-2@0.7; meat 1-1@0.3		
2	blightspitter	Blightspitter / Blightspitter	spitter	tide 💧	38	11	1	2.8	16	30	12	0.9	2.6	18	essence 1-2@0.7; herb 1-1@0.4	1	blob_
3	blightbat	Blightbat / Blightbat	swooper	storm ⚡	30	8	1	5.4	14	32	2	0.5	1.5	14	essence 1-1@0.6; fiber 1-1@0.5	0.9	flying (h=0.
4	blightbrute	Blightbrute / Blightbrute	crawler	stone ⛰️	95	13	5	2.7	11	26	3	1	2.4	30	essence 1-2@0.7; ore 1-2@0.5; meat 1-2@0.5	1.25	big_C tint=#
5	blightleaper	Blightleaper / Blightleaper	crawler	tide 💧	42	10	1	4.2	14	30	2.4	0.6	1.8	20	essence 1-1@0.6; meat 1-1@0.5; fiber 1-2@0.5	0.95	big_F tint=#
6	blightwi	Blightwisp / Blightwisp	swooper	arcane 🔮	26	10	0	4.6	15	32	11	0.7	2	22	essence 1-2@0.7; aether_dust 1-1@0.2	0.9	flying hover

#	id	名称 FR / EN	kind	元素	HP	ATK	DEF	SPD	aggro	leash	range*	windup*	cd	XP	掉落 item min-max@chance	scale	模型
	sp																
7	blightthornling	Blightthornling / Blightthornling	spitter	verdant 🍃	52	9	3	2.4	14	28	10	0.9	2.4	22	essence 1-1@0.6; herb 1-2@0.6; fiber 1-2@0.5	1	blob_ (h=0.
8	blightspore	Blightspore / Blightspore	spitter	verdant 🍃	34	8	1	2.2	12	24	8	0.9	2.6	18	essence 1-1@0.6; berry 1-2@0.5; herb 1-1@0.4	0.85	blob_ (h=0.
9	blightdrake	Blightdrake / Blightdrake	swooper	ember 🔥	115	15	4	4.4	18	36	14	1	2.2	48	essence 2-3@0.9; ember_core 1-1@0.35; resin 1-2@0.5	1.15	flying hover tint=#
10	elite_mossmatriarch	Matriarche Thistle / Thistle Matriarch	elite	verdant 🍃	320	16	6	3.2	16	36	3.2	1	2.4	140	essence 4-6@1; herb 4-6@1; plank 3-3@1	2	big_C
11	elite_	Alpha Cinderhorn / Cinderhorn Alpha	elite	ember 🔥	520	24	9	4	18	40	3.4	1.1	2.2	260	ember_core 2-2@1; resin 4-6@1; essence 5-8@1	2.3	big_C

#	id	名称 FR / EN	kind	元素	HP	ATK	DEF	SPD	aggro	leash	range*	windup*	cd	XP	掉落 item min-max@chance	scale	模型
	cinderhorn																
12	elite_prismwarden	Gardien Prismatique / Prism Warden	elite	arcane 🔮	760	32	12	3.6	20	42	3.8	1.2	2.2	420	prism_shard 3-4@1; aether_dust 1-2@1; essence 6-9@1	2.4	程序化
13	elite_blightalpha	Alpha de la Corruption / Blight Alpha	elite	storm ⚡	420	22	8	4.2	18	60	3.2	1	2.2	200	essence 5-8@1; ingot 1-2@1; brick 2-3@1; herb 2-3@1	2.1	big_B (h=1.
14	elite_voidstalker	Rôdeur du Vide / Void Stalker	elite	arcane 🔮	950	38	13	4.2	20	44	3.6	1.1	2.1	600	prism_shard 3-5@1; aether_core 1-1@0.5; essence 8-12@1	2.3	big_D tint=#
15	elit	Aile-Crypte / Cryptwing	elite	storm ⚡	820	36	10	4.8	22	46	4	1.1	2	580	prism_shard 2-4@1; aether_dust 2-3@1;	2.1	flying (h=1. tint=#

#	id	名称 FR / EN	kind	元素	HP	ATK	DEF	SPD	aggro	leash	range*	windup*	cd	XP	掉落 item min-max@chance	scale	模型
	e_cryptwing														essence 8-12@1		
16	elite_hollowking	Roi Creux / Hollow King	elite	verdant 🍃	1100	34	15	3.4	18	40	4	1.2	2.3	650	ingot 2-3@1; aether_core 1-1@0.6; essence 10-14@1	2.5	big_M (h=1.
17	boss_aethon	Aethon, le Titan Creux / Aethon, the Hollow Titan	boss	arcane 🔮	2600	38	14	3	40	60	5	1.3	2	1500	aether_dust 5-5@1; essence 20-20@1	4.2	程序化
18	blightfish	Blightfish / Blightfish	crawler	tide 💧	62	10	2	4.2	12	28	2.4	0.6	1.6	18	essence 1-1@0.6; meat 1-2@0.5; fiber 1-1@0.4	1	big_F 包内
19	blighttribal	Blighttribal / Blighttribal	spitter	arcane 🔮	50	12	3	3	15	30	11	0.9	2.4	24	essence 1-2@0.7; herb 1-1@0.4; fiber 1-1@0.3	1	big_T 包内
20	blig	Blightmask / Blightmask	swooper	ember 🔥	34	9	1	5.2	14	32	2	0.5	1.5	18	essence 1-1@0.6; fiber 1-1@0.5	0.9	flying hover

#	id	名称 FR / EN	kind	元素	HP	ATK	DEF	SPD	aggro	leash	range*	windup*	cd	XP	掉落 item min-max@chance	scale	模型
	htmask																

敌人等级：普通 Blight 由栖息地 `levelFor(biome)` 决定（§4.4），每级 +8.5%；精英、Boss 永远以 1 级生成且不随等级缩放（`spawnEnemy(a.enemy, x, z, 1)` L46987；Boss L47914）。Boss 召唤的小怪为 12 级（L45794）。

1.7 敌人招式表 **B9**（L33880–34167）

`pe(id,name,kind,{…})` 默认值：`windup .7, recover .6, power 1, min 0, max 3`（L33868）。`weight` 为加权随机权重（默认 1）。

敌人	招式 id	名称 FR → EN	kind	windup	recover	power	min	max	其他
blightcrawler	bite	Claquement de mandibules → Mandible Snap	melee	0.7	0.6	1	0	2.5	arc=1.6
blightcrawler	lunge	Bond → Pounce	dash	0.6	0.6	1.1	3	6.5	speed=13, width=1.6, weight=0.6
blightspitter	spit	Glaire acide → Acid Glob	proj	0.9	0.6	1	3	13	speed=13
blightspitter	smack	Claque → Smack	melee	0.7	0.6	0.8	0	2.4	arc=1.5
blightbrute	slam	Écrasement → Crush	quake	1	0.6	1.1	0	4.2	radius=4.2
blightbrute	swipe	Gifle → Slap	melee	0.8	0.6	1	0	3	arc=1.8
blightleaper	leap	Saut sauvage → Wild Leap	dash	0.55	0.6	1.2	3.5	9	speed=17, width=1.5
blightleaper	claw	Griffure → Scratch	melee	0.55	0.6	0.9	0	2.3	arc=1.5
blightwisp	orb	Orbe sombre → Dark Orb	proj	0.7	0.6	1	3.5	12	speed=11
blightthornling	fan	Salve d'épines → Thorn Volley	thorns	0.9	0.6	0.7	3	11	arc=0.9, speed=13, count=3
blightthornling	smack	Coup d'épine → Thorn Strike	melee	0.7	0.6	0.8	0	2.4	arc=1.5
blightspore	puff	Nuage de spores → Spore Cloud	burst	0.9	0.6	0.6	0	6	speed=8, count=8
blightspore	spit	Crachat de spores → Spore Spit	proj	0.9	0.6	0.8	3	9	speed=11
blightdrake	flame	Rayon de braise → Ember Beam	beam	1	0.6	1.1	4	14	width=1.3
blightdrake	dive	Piqué → Dive	dash	0.6	0.6	1	2	8	speed=15, width=1.6, weight=0.7
blightbat	dive	Piqué → Dive	dash	0.5	0.6	1	1.5	8	speed=14, width=1.5
elite_mossmatriarch	slam	Fracas d'épines → Thorn Slam	quake	1.1	0.6	1.2	0	5	radius=5.2
elite_mossmatriarch	spray	Pluie d'épines → Thorn Spray	thorns	0.9	0.6	0.7	3	14	arc=1.1, speed=14, count=5
elite_mossmatriarch	swipe	Coup de liane → Vine Swipe	melee	0.8	0.6	1	0	3.8	arc=2
elite_cinderhorn	charge		dash	1.1	0.6	1.5	5	16	

敌人	招式 id	名称 FR → EN	kind	windup	recover	power	min	max	其他
		Charge ardente → Blazing Charge							speed=17, width=2.6
elite_cinderhorn	stomp	Piétinement de cendres → Cinder Stomp	quake	1.2	0.6	1.2	0	6	radius=6.2
elite_cinderhorn	gore	Coup de corne → Gore	melee	0.8	0.6	1	0	4	arc=1.8
elite_prismwarden	beam	Rayon prismatique → Prism Beam	beam	1.2	0.6	1.4	4	20	width=1.6
elite_prismwarden	shatter	Explosion d'éclats → Shard Burst	burst	1	0.6	0.8	0	20	speed=12, count=10
elite_prismwarden	quake	Séisme de cristal → Crystal Quake	quake	1.2	0.6	1.2	0	7	radius=7
elite_voidstalker	charge	Charge du vide → Void Charge	dash	1	0.6	1.5	5	16	speed=18, width=2.6
elite_voidstalker	quake	Séisme du vide → Void Quake	quake	1.2	0.6	1.2	0	6	radius=6.2
elite_voidstalker	shatter	Éclats de vide → Void Shards	burst	1	0.6	0.8	0	20	speed=12, count=10
elite_voidstalker	rend	Déchirure → Rend	melee	0.75	0.6	1	0	3.8	arc=1.9
elite_cryptwing	beam	Rayon spectral → Spectral Beam	beam	1.1	0.6	1.4	4	20	width=1.6
elite_cryptwing	nova	Nova spectrale → Spectral Nova	burst	1	0.6	0.8	0	20	speed=12, count=12
elite_cryptwing	dive	Piqué fantôme → Ghost Dive	dash	0.8	0.6	1.2	3	12	speed=17, width=2.2, weight=0.8
elite_hollowking	spray	Pluie d'épines → Thorn Spray	thorns	0.9	0.6	0.7	3	14	arc=1.4, speed=14, count=7
elite_hollowking	slam	Séisme du roi → King's Quake	quake	1.2	0.6	1.3	0	6.5	radius=6.5
elite_hollowking	swing	Coup de masse → Mace Blow	melee	0.85	0.6	1.1	0	4.2	arc=1.9
elite_blightalpha	pounce	Bondissement → Lunge	dash	0.9	0.6	1.3	4	13	speed=16, width=2.2
elite_blightalpha	rend	Déchiquetage → Rend	melee	0.75	0.6	1	0	3.6	arc=1.9
elite_blightalpha	bolt	Éclair de faille → Rift Bolt	bolt	0.9	0.6	1	5	14	
blightfish (借用 blightcrawler)	bite	Claquement de mandibules → Mandible Snap	melee	0.7	0.6	1	0	2.5	arc=1.6
blightfish (借用 blightcrawler)	lunge	Bond → Pounce	dash	0.6	0.6	1.1	3	6.5	speed=13, width=1.6, weight=0.6
blighttribal (借用 blightspitter)	spit	Glaire acide → Acid Glob	proj	0.9	0.6	1	3	13	speed=13
blighttribal (借用 blightspitter)	smack	Claque → Smack	melee	0.7	0.6	0.8	0	2.4	arc=1.5
blightmask (借用 blightbat)	dive	Piqué → Dive	dash	0.5	0.6	1	1.5	8	speed=14, width=1.5

1.8 精英 / Boss 汇总

id	名称 EN	位置 (x,z) → 群系	生成条件	重生	击退抗性	备注
elite_mossmatriarch	Thistle Matriarch	(-48, 12) → meadow	群系已解锁、玩家 85 m 内、eliteDown 已过期	900 s（fx = 900，	0.7	

id	名称 EN	位置 (x,z) → 群系	生成条件	重生	击退抗性	备注
				L27894；px L27942）		
elite_cinderhorn	Cinderhorn Alpha	(34, -38) → woods	同上	900 s	0.7	
elite_prismwarden	Prism Warden	(-30, -76) → wastes	同上	900 s	0.7	没有 ip 条目 → 永远程序化（八面体水晶核 + 8 根旋转水晶，L30009）
elite_voidstalker	Void Stalker	(112, -96) → rift	同上	900 s	0.7	
elite_cryptwing	Cryptwing	(12, 122) → rift	同上	900 s	0.7	
elite_hollowking	Hollow King	(-62, -118) → rift	同上	900 s	0.7	
elite_blightalpha	Blight Alpha	无固定点	世界事件 `elite_hunt`（玩家 ≥ Lv4，L48039），在玩家周围 28–56 m 随机点	事件结束（480 s）即移除	0.7	击败额外给 aether_dust×2、essence×4（L48062）
boss_aethon	Aethon, the Hollow Titan	竞技场 dA=(0,-90) r=15 → wastes	需 `world.beacon`，玩家进入竞技场（L47890–47895）	可无限重赛（rematches）	1.0	程序化（无 ip），见 §4.6

（群系由 `At(x,z)`（L25809）实算；坐标来自 `mp` L26367。）

2. 元素克制与伤害公式

2.1 元素表 `qe`（L23200）

key	FR	EN	颜色	图标	特效色 `x9`（L33811）
verdant	Verdoyant	Verdant	#68d391	🍃	0x7ae27a
ember	Braise	Ember	#fc8181	🔥	0xff792a
tide	Marée	Tide	#63b3ed	💧	0x5ab4ff
storm	Foudre	Storm	#f6e05e	⚡	0xfaf089
stone	Pierre	Stone	#a0aec0	⛰️	0xb0a090
arcane	Arcane	Arcane	#d6bcfa	🔮	0xc79bff

2.2 克制环（L23188–23199）

```
cd = { verdant: "stone", stone: "storm", storm: "tide", tide: "ember", ember: "verdant", arcane: null };
function ga(s, A) {                    // s = 攻击方元素, A = 防御方元素
  return cd[s] === A ? 1.5 : cd[A] === s ? .7 : 1
}
```

环：Verdant → Stone → Storm → Tide → Ember → Verdant（箭头 = 对其 ×1.5）。Arcane 不克也不被克（打谁、被谁打都 ×1）。

攻 \ 防	verdant	ember	tide	storm	stone	arcane
verdant	1	0.7	1	1	1.5	1
ember	1.5	1	0.7	1	1	1
tide	1	1.5	1	0.7	1	1
storm	1	1	1.5	1	0.7	1
stone	0.7	1	1	1.5	1	1

攻 \ 防	verdant	ember	tide	storm	stone	arcane
arcane	1	1	1	1	1	1

（非相邻对，例如 verdant↔tide、ember↔storm，均为 1。UI 里“强/弱于”列表 L49031–49032 也是用 `ga(...) > 1` 生成的。）

元素倍率适用范围（重要）：

- 单位→单位（`damageHittable` 非玩家目标，L34544）：`c = ga(this.element, a.element)`——队友打野怪、野怪打队友都有元素。
- 投射物：野生阵营打队友有元素（L46794）；队友阵营投射物打野怪有元素（L46803 `n.team === "ally" ? ga(...) : 1`）。
- 玩家本身没有元素：玩家近战 `Ys(A.atk, d.def, c, 1, …)`（L46339）、玩家弹弓、敌人打玩家（L34539）倍率恒为 1。

2.3 伤害公式 **`Ys`**（L27896）

```
function Ys(s, A, e, t, n, i = .08) {     // s=攻方ATK, A=守方DEF, e=招式power, t=元素倍率, n=rng, i=暴击率
  const r = n() < i,
        a = (s * e - A * .5) * t * (.9 + n() * .2) * (r ? 1.5 : 1);
  return { dmg: Math.max(1, Math.round(a)), crit: r }
}
```

即 dmg = max(1, round((ATK·power − 0.5·DEF) · 元素倍率 · U(0.9,1.1) · (暴击? 1.5:1)))。

后续修正（按调用链）：

1. 单位攻击：`damageHittable`（L34536）以 `Ys(..., 0)`（内部暴击率 0）计算，再单独掷 `critChance()`：`0.08 + critBonus`（`+0.04 sparkhopper/pricklehat`）（`+0.06 gloomwing 夜间`）（L34294），暴击 ×1.5。
2. 玩家近战（L46313）：三段连击 power `[1, 1.05, 1.55]`，暴击率 0.10；第三段击退 +4。玩家 ATK = `8 + 1.6(lv-1) + 武器`（`it()` L27273）。
3. Boss 受伤倍率 `N9`（L34955）：入场动画/无敌 → 0（显示“Immunisé”），`vuln`（核心发金光）→ ×1.5，其余 → ×0.75。
4. 护盾 `shield` 先吸收（L46501）。
5. 野生生物不会被非玩家来源杀死（L46506）：`A.kind === "creature" && t !== "player"` → `o = min(o, floor(hp) - 1)`——队友只能把野生生物打到 1 HP（方便捕获）。
6. 重击硬直：生物/普通敌人单次受到 ≥30% maxHp 伤害 → `stunned` 1.4 s（L46530）。
7. 击退（L46524）：`i × (暴击1.5) × (boss .2 | elite .6 | 1) × (1 + min(.6, 2·dmg/maxHp))`，再乘 `(1 − knockResist)`（L34345）。
8. 队友受伤 `damageAlly`（L46532）：护盾 → pebblepaw ×0.85 → `QUALITY.fix.ally`：`state.creatures[0]`（第一只获得的伙伴）只受 40% 伤害（`FIRST_DMG = .4`），且玩家 < Lv10 时每 40 s 可有一次“s'accroche !”锁 1 HP 不倒（L43786–43806）。

9. 玩家受伤 `hurtPlayer`（L46464）：Rift 日常词缀 `fury` ×1.25（`Ui().dmg`），护盾吸收，受伤无敌 0.6 s。

出招冷却（L34523）：

```
this.cd = i.recover + (this.edef ? this.edef.cd * .5 : (this.cdef?.ability.cd ?? 2) * .55)
```

即 敌人：`spec.recover(0.6) + uM.cd×0.5`；生物：`spec.recover(0.5) + ability.cd×0.55`；冲刺类结束后 `cd = recover + 0.9`（L34534）。

3. 捕获 / 驯服、队伍、经验

3.1 捕获合法性 `hx`（L27857）

1. 物种必须在 `Ge`（敌人/精英/Boss 一律“Cette créature ne peut pas être capturée”）。
2. 必须选中捕获器（`deviceTier`）。
3. `deviceTier ≥ sp(rarity)`：普通~稀有 1 档，史诗 2 档，神话 3 档。
4. 目标必须存活且未在被捕获。

目标选择 `captureTarget`（L47011）：锁定目标 16 m 内优先；否则 14 m 内、镜头前方锥形（桌面 `cos ≥ 0.35`，触屏 `-0.3`，距离 <2.5 m 不检查朝向），按 `dist × (2 − cosθ)` 取最小。收藏已满（`ls()`）时禁止投掷。

3.2 捕获概率 `ux`（L27849）

```
function ux(s) {
  const A = Ge[s.sp], e = NA[s.deviceId];
  if (!A || !e?.deviceMult) return 0;
  const t = 1 + (1 - ze(s.hpRatio, 0, 1)) * 1.8;                    // 血量越低越高，最高 ×2.8
  let n = A.capture * e.deviceMult * t * .62;
  return s.baitId && NA[s.baitId]?.bait === A.bait && (n += .18),     // 匹配诱饵 +18%
         s.stunned && (n += .1),                                      // 硬直 +10%
         s.asleep && (n += .15),                                      // 睡眠 +15%（全代码无处设置 asleep → 死分支）
         n += s.charmBonus ?? 0,                                      // 护符 + Rift 词缀
         ze(n, .02, .95)                                              // 夹到 [2%, 95%]
}
```

P = clamp(capture × deviceMult × (1 + 1.8·(1 − HP%)) × 0.62 + 0.18·[诱饵匹配] + 0.10·[stunned] + charmBonus, 0.02, 0.95)

调用处（L47119–47127）：`charmBonus = it(e).captureBonus + (At(x,z) === "rift" ? Ui().charm : 0)`；新手引导用的 starter Fluffern 强制 `chance = max(chance, 0.94)`（L47129）。

捕获器（L24499–24542）：

id	FR / EN（EN 表）	档位	deviceMult	投掷球颜色（L47160）
snare	Collet en fibres	1	1.0	0xa0aec0
cage	Cage en osier	2	1.5	0xc28a4a
orb	Orbe d’éther	3	2.2	0x9f7aea

id	FR / EN（EN 表）	档位	deviceMult	投掷球颜色（L47160）
lantern	Lanterne prismatique	4	3.2	0xf6ad55

诱饵（L24543–24582）：`sweet_bait / savory_bait / mineral_bait / arcane_bait`，对应物种 `bait` 字段（sweet/savory/mineral/arcane）。物品描述写“+20 %”，代码实际 +0.18（绝对值）。开启 `settings.autoBait` 时自动从背包挑匹配诱饵（`findBait` L47003）。投掷时捕获器与诱饵各消耗 1（L47150–47151）。

其他加成：`bond_charm`（Charme du Lieur）`captureBonus: .12`（L24755）；Rift 日常词缀 “Faille apaisée” `charm: .15`（L42420，仅在 Rift 群系内）。

示例（由公式计算）：

- Fluffern，满血，snare：0.62×1×1×0.62 = 38.4%；10% 血：×2.62 → 100.7% → 夹到 95%。
- Tempestail（mythic，需 orb），满血：0.12×2.2×0.62 = 16.4%；10% 血 + arcane 诱饵：0.164×2.62 + 0.18 = 60.9%。

3.3 捕获演出与结算（**startCapture** L47149 / **runCapture** L47178 / **dx** L27886）

```
function dx(s, A) {                    // 掷骰在投掷瞬间就决定
  const e = s() < A, t = e ? 3 : Math.floor(s() * 3);
  return { success: e, shakes: t }     // 成功固定晃 3 下；失败晃 0–2 下
}
```

- 阶段：`throw` 0.55 s 抛物线 → `wobble` 每晃 0.75 s → `result` 0.5 s；投掷冷却 `capCd = 1.2`。
- 成功：`Ac()` 生成个体（等级 = 野生等级，pot 随机，HP 满），`dex[sp].caught++`；玩家 `+10 + order×8` XP（L47209）；若是 starter 且队伍为空，自动入队。
- 失败（L47224）：`docile` → 朝反方向 `flee`；其他 → `provoked = 12`、锁定玩家、进入 `stunned`。

3.4 收藏 / 队伍 / 伙伴逻辑

- 收藏上限 `ls()`（L27248）：`min(66, 30 + 6 × 庇护所数量)`。
- 队伍上限 3（`XM` L27298）；正在基地工作的个体不能入队，队员也不能被分配工作（`ex` L27384）。
- 队员实例化为 `oc("ally", sp, lv, inst)`（`syncParty` L46852），属性取 `xt(inst)`。
- 队友 AI **`R9`**（L34812）：
  - 阵型槽 `P9 = [[-2.6,-2.4],[2.6,-2.4],[0,-4.6]]`（玩家朝向局部坐标，L34806）；跟随速度 `spd × clamp(0.6 + d/6, 0.6, 2)`。
  - 距玩家 > 30 m 或卡住 > 4 s → 传送回玩家身后 3 m。

  - 每 0.3 s 选目标：优先玩家锁定的敌对目标（< 22 m），否则玩家 17 m 内最近的敌对野怪（`isHostileToAllies`：敌人，或 aggressive/被激怒的生物）。
  - 攻击接近距离：近战/地震/冲刺 `max(1.4, max×0.7)`，其余 `max×0.7`；追击速度 `spd×1.15`；出招条件 `dist ≤ max×0.95 + 目标半径` 且 `≥ min×0.5`。
  - 跟随状态回血 0.4%/s（L34819）；twigwing/carrotail 每 4 s +1。
  - 支援型（主技能 heal/buff）：`QUALITY.fix.specs`（L43509）额外给一个 power 0.5 的“Coup léger”（heavy 原型为近战 2.2 m，其余为 9 m 投射物，recover 1.5）；只在自身 HP > 50% 且目标已在射程内才普攻（`F.eng` L43547），不追击。
  - 治疗 `D9`（L34875）：有队友 HP < 78% 或玩家 HP < 70% 且在射程内时施放；量 = `power × (1 + .085(lv-1))`，最伤者 100%、其余 50%；冷却 `recover + ability.cd`。
  - 护盾 buff：玩家 20 m 内有敌人才施放；队友护盾 = `maxHp × power × lvMul`，持续 8 s；玩家护盾 = `60 × power × lvMul × 1.5`。
- KO：HP 归 0 → `ko`，需 Tonique spirituel 或治疗站；不在队伍中的个体每秒回 maxHp 的 0.4%（有庇护所则 2%），恢复到 30% 自动解除 KO（`gl` L27443–27446）。
- 羁绊 bond（`qr` L27206 / `updateCcompanions` L45960）：队员每 60 s +1；玩家 HP < 30% 且战斗中，随机队员“守护”为玩家回 20% maxHp（冷却 150 s）并 +2；非战斗每 150–270 s 随机队员按群系“找到”1–3 个资源并 +2；击杀敌人时每个未倒队员 +1（精英 +3、Boss +8，`px` L27941）。注意：击杀野生生物不给羁绊和伙伴经验（`killActor` 生物分支不调用 `px`，L46555–46566）。

3.5 经验与等级曲线

```
const ml = 30, _o = s => Math.round(30 * Math.pow(s, 1.4));     // 生物：Lv→Lv+1 所需 XP，上限 30 级（L27241–27245）
const hd = 30, tr = s => Math.round(50 * Math.pow(s, 1.5));     // 玩家：上限 30 级（L25270–25271）
```

等级	生物升级所需 **_o**	生物累计	玩家升级所需 **tr**
1	30	30（到 Lv2）	50
2	79	109（到 Lv3）	141
3	140	249（到 Lv4）	260
4	209	458（到 Lv5）	400
5	286	744（到 Lv6）	559
10	754	3525（到 Lv11）	1581
15	1329	8983（到 Lv16）	2905
20	1989	17578（到 Lv21）	4472
25	2718	29683（到 Lv26）	6250
29	3346	42112（到 Lv30）	7808

- 生物升级（`xp()` L27286）：可一次连升多级，每次升级（若有）回复 25% maxHp；满级后 XP 归零。

- 伙伴 XP 来源：击杀敌人（px L27905）时每个未 KO 队员获得 `ceil(敌人XP × 0.5)`（玩家拿全额）；训练所（training）里工作时 `0.6 XP/s`（L27485–27490）。
- 玩家 XP 来源：击杀敌人 `uM.xp`（Boss 重赛 ×0.5）；击杀野生生物 `4 + 2·lv + 4·order`（L46558）；捕获 `10 + 8·order`；发现地点 +25（巢穴/竞技场 +15）；首次进入群系 +25；Rift 词缀 bounty 额外 +60%。
- 金币：`pc(xp, kind) = max(1, round(xp × (boss 1.2 | elite .55 | 其他 .32)))`（L41988）。
- 敌人掉落（px）：每项独立按 chance 掷骰，数量 `min + floor(rand·(max−min+1))`，Boss 掉落 ×(1 + 0.5·rematches)。野生生物被击杀固定 60% essence、50% meat（稀有度 ≥ rare 时数量 +1，L46565）。

4. 敌人 / 野生生物 AI 与刷新规则

4.1 单位初始化差异（**oc** 构造，L34262–34280）

项	野生/队友生物	普通敌人	精英	Boss
kind	creature	enemy	elite	boss
aggro（侦测半径 m）	aggressive = 11，其他 = 0	`uM.aggro`	`uM.aggro`	`uM.aggro`（40）
leash（拴绳 m）	30	`uM.leash`	`uM.leash`	60（实际被限制在竞技场内）
specs	`QUALITY.fix.specs(i, y9(i))`	`B9[id] ?? wkEspec(id)`	`B9[id]`	动态生成（§4.6）
飞行 flies	flyer / support / 带 hover 的 elemental	仅 swooper	否	否
knockResist	cindertusk/mossback 0.6，其余 0	0	0.7	1
AI	`S9`（野生）/ `R9`（队友）	`S9`	`S9`	`z9`

低等级玩家仇恨缩减 `QUALITY.fix.ag`（L43771–43780）：对普通敌人与野生生物（不含精英/Boss/事件单位），玩家等级 < 3 → 侦测半径 0（只在被打后反击）；3–4 级 → 半径 ×0.5；≥ 5 → 原值。

4.2 野生 AI 状态机 **S9**（L34705–34804）

每帧公共处理：`stateT += dt; cd -= dt; provoked -= dt`；单位间分离推开；玩家 15 m 内首次出现 → 图鉴“发现”。 `n`（是否主动敌对）= 非生物 ∨ aggressive ∨ provoked > 0。

```
          ┌──────── 受击(C9) ────────┐
idle ⇄ patrol ──(侦测)──► detect ──0.55s/0.4s──► chase ──(chooseAttack & cd≤0)──► attack ──► recover ──► chase/idle
  ▲                                          │  │                                  (dash: updateDash)
  │                                          │  └─(HP低)──► flee ──4s或>22m──► return
  └──────── return ◄──(超出 leash / 丢失视线 5s / 目标远)──┘
stunned（受重击或捕获失败）1.4s ──► chase（被激怒/非生物）或 idle
dead：淡出（普通 0.5s 后 0.9s，Boss 2.2s 后 2s）
```

状态	行为（代码事实）
idle	减速；每秒回 3% maxHp；若 `n` 且 `scan`（每 0.1 s 一次，按 id 错峰，L44847）通过，用 `bl()` 在 `ag(s) + (provoked?16:0)` 半径内找最近目标（敌对野怪的目标集合含玩家与队友，需视线，水体/屏障阻挡 `Vp=["water","barrier"]`）→ `detect`。`stateT > wanderT` 时在出生点 8 m 内随机取点 → `patrol`。

状态	行为（代码事实）
patrol	以 spd × 0.45 走向巡逻点；到达（<1.2 m）或 7 s 超时 → idle（等待 1.5–4.5 s）；途中同样侦测。
detect	转向目标，头顶冒 “!”，播放 attack 语音（敌人播放 boss_roar）；生物 0.55 s、敌人 0.4 s 后 → chase。
chase	① 目标失效时在 aggro + 8 内重找（不需视线），找不到 → return。② 计算 LOS（20 m 内每帧，远处 15 Hz 缓存，L44859）。③ 脱战条件：离出生点 > leash，或丢失视线 > 5 s，或目标距离 > leash×1.4 + 12 → return。④ 逃跑：生物且非 aggressive 且 HP < 18%（只一次），或任何生物 HP < 12% 时 50% 概率 → flee。⑤ chooseAttack(dist)（L34391）：在 [min,max] 覆盖当前距离的非 heal/buff 招式中按 weight 加权随机。⑥ 移动：若全部招式 min > 1.5（纯远程，_sv），距离 < specs[0].min + 0.8 时以 spd×0.8 后退风筝，距离 > specs[0].max × 0.8 才靠近；否则距离 > 最短 max×0.85 时以 spd（生物 ×1.05）追击，进入范围站定转向。⑦ cd ≤ 0 且（非 proj/bolt/beam 或有视线）→ beginAttack。
attack	beginAttack（L34401）：速度 ×0.2，野生方显示地面预警（melee/thorns 扇形、quake/burst 圆、dash/beam/bolt/proj 线）。updateAttack（L34508）：前摇期间以 3.2/s 速率追踪目标方向（quake/burst 不追踪），前摇结束 fire()；dash 转入冲刺（速度 spec.speed ?? 14，时长 min(0.9, max/speed)，碰墙即停，每目标只命中一次，击退 7）。
recover	减速，stateT > spec.recover（默认 .6）→ 有目标/被激怒则 chase，否则 idle。
return	以 spd×0.95 回出生点，每秒回 8% maxHp，清除激怒；到达（<2 m）或 14 s → idle，敌人直接回满血。
flee	背离威胁点以 spd×1.25 跑；4 s 或距离 > 22 m → return。
stunned	速度 ×0.8 衰减，1.4 s。

受击反应 C9（L34694）：生物 docile 被打 → 立刻 flee（取消预警、清除激怒）；neutral/aggressive 被打 → provoked = 14 s、锁定攻击者、从 idle/patrol/return/detect 直接进 chase。敌人被打 → 锁定攻击者并 chase。

性格总结：docile 永不主动攻击、被打即逃、捕获失败也逃；neutral 不主动攻击，被打后激怒 14 s（侦测半径 +16）；aggressive 侦测半径 11 m 主动攻击，且不会因 18% 血线逃跑（仍有 12% 血线 50% 的逃跑）。

招式命中判定（fire L34549）：melee 扇形 |Δθ| ≤ arc/2 + 0.25 或贴身（< r+0.8）；quake 圆形 ≤ radius + r/2；proj/thorns 发射实体投射物（thorns 按 count、arc 扇形散布）；bolt 自动锁定前方 ±0.5 rad、max+2 内最近目标；beam 线段宽 width/2 + r；burst 环形 count 发弹幕（交替偏移 0.3 rad）。

模拟 LOD（QUALITY.logic.tick L44818）：攻击/侦测/硬直/死亡/受击/冲刺/Boss/事件单位每帧思考；其余按距离档（MEDIUM：<20 m 每帧、<50 m 30 Hz、<100 m 10 Hz、更远 4 Hz；屏幕外再降一档），位置用速度外推。

4.3 精英

精英走同一个 S9 状态机，区别全在数据：更大的 aggro/leash、3–4 个招式、knockResist 0.7、受击击退 ×0.6、不受玩家等级仇恨缩减、无等级缩放；击杀后 eliteDown[id] = clock + 900，并给出 banner 与稀有掉落。没有额外阶段逻辑（阶段机制只属于 Boss）。elite_blightalpha 的 leash = 60（事件怪，追得更远）。

4.4 刷新规则（**updateSpawner** L46887）

节奏与上限：每 0.5 s 运行一次；Boss 战中（bossLocked）停止。上限 QUALITY.actorCap() = 画质档 actors（LOW 19 / MEDIUM 27 / HIGH 35 / ULTRA 40；移动端 14 / 20 / 27）×actorMul，最少 6；栖息地在玩家 48 m 内时可刷到满上限，更远只到 60%。

清理：距玩家 > 100 m 且不在 chase/attack 的单位被移除（Boss、事件单位、starter、捕获中除外）；生物另有一个 `&& !1` 的被关闭分支（L46898，原本会在 28 m 外移除未激怒生物——已禁用）。

栖息地遍历：取 `d ≤ r + 75` 的栖息地，按距离排序；群系未解锁则跳过；每个栖息地有 `max` 个槽位：

- 槽位里的单位死亡 → 重生计时：营地 150 s，否则 28–68 s（`28 + rand×40`）；被清理（未死）→ 8 s。
- 生成点：栖息地圆内均匀随机（`sqrt(rand)·r·0.95`），距玩家 ≥ 16 m（blight）/18 m（生物），blight 不在主营地 RA=(0,58) 24 m 内（营地除外），不在水下/陡坡/障碍/竞技场旁，最多试 10 次（`findSpawnPos` L46864）。
- 按权重随机物种；模型未加载完的物种跳过（`QUALITY.assets.ready`）。
- 营地（`camp`）：槽位全部被击杀 → “Camp nettoyé !”，`campDown = clock + 900` s，期间不刷新。

等级 `levelFor`（L46878）：

```
const t = A === "meadow" ? [1, 4] : A === "woods" ? [5, 9] : A === "rift" ? [14, 20] : [9, 14];
return ze(t[0] + Math.floor(Math.random() * (t[1] - t[0] + 1)) + (e >= 3 ? 2 : 0), 1,
        40)   // epic/mythic 生物 +2
```

群系（At L25809）：`max(|x|,|z|) > 96` 且离竞技场 > 29 m → rift（外环）；否则按 z 与两条正弦边界：`z > Br(x)`（≈ −10）→ meadow；`z > Er(x)`（≈ −58）→ woods；其余 → wastes。

栖息地生成（世界生成 `TM` 内，L26625–26770，确定性 RNG）：

群系	生物栖息地	物种权重
meadow（固定 5 个，max 3）	主营地周边	① fluffern 3 / twigwing 2 / puddlepup 1 / carrotail 2；② pebblepaw 2 / sparkhopper 2 / fluffern 1；③ puddlepup 2 / emberkit 1 / bloomcap 1；④ (-38,46) 池塘 puddlepup 3 / dewdrake 2；⑤ (40,36) dewdrake 1 / puddlepup 2 / bloomcap 1
meadow（随机 6 个）	按物种 `biomes` 自动	权重 = `He.spawn`（common 1, uncommon .6, rare .3, epic .14, mythic .05）
woods（随机 9 个）	同上	同上
wastes（随机 8 个）	同上	epic ×2、mythic ×0.6
wastes 特殊 “garden” (-52,-66) r12 max2	—	prismoth / aetherling / tempestail / pyrelope：mythic 权重 1，其余 2
rift（随机 8 个，max 3）	同上	epic ×1.6、mythic ×0.7、rare ×1.3、common/uncommon ×0.6

Blight 栖息地	位置	权重
meadow 固定 7 处（max 3）	主营地周围、(20,18)、(-14,24)、(38,4)、(-58,38)、(8,2) 等	以 crawler/leaper/thornling/bat/spore/spitter/fish/wisp/brute/tribal 小组合为主（详见 L26707–26721）
woods 最多 9 处（r12, max3）	随机	crawler 2, spitter 2, bat 2, brute 1, leaper 2, wisp 1, thornling 1, drake 0.4, fish 2, tribal 1, mask 1
wastes 最多 9 处	随机	spitter 2, bat 2, brute 2, wisp 2, drake 1, leaper 1, crawler 1, tribal 2, mask 2, fish 1

Blight 栖息地	位置	权重
各 camp 地点（max 4，营地）	En 中 type=camp	crawler 2, spitter 1, brute 1, thornling 1
wastes 营地 (46,-66)	营地	spitter 2, bat 2, brute 1, wisp 1
rift 16 处（max 4）	随机	brute 2, drake 2, wisp 2, leaper 2, spitter 1, thornling 1, spore 1
rift 营地 (-108,80)（max 5）	营地	brute 2, drake 1, wisp 2, leaper 1

其他刷新来源：

- Starter（`ensureStarter` L46012）：玩家还没有 starter 时，在主营地 60 m 内、镜头前 10–15 m 刷一只 2 级 Fluffern，HP 45%，捕获率 ≥ 94%。
- 世界事件（`updateWorldEvents` L48031）：玩家 ≥ Lv2 才触发；类型 `migration`（同群系 uncommon~epic、`!night` 的物种刷 3 只）、`rare_resource`（4 个临时水晶节点）、`elite_hunt`（≥ Lv4，Blight Alpha）；持续 360 s（elite_hunt 480 s），间隔 420–660 s。
- 发现系统（`this.disc.spawn` L42283）：在已解锁群系随机放置 plant/stash/ore/creature；creature 优先选图鉴未见过的物种。
- Boss 召唤：见 §4.6。

4.5 昼夜

- 时间：一天 = `Wd = 720` 秒（L42464；`hour += dt × 24/720`）。天空/光照有视觉夜晚（`atmo.night`，L41516）。
- 但游戏逻辑上的 **`night`** 每帧被强制写成 **`false`**：`updateGame` L45919 `…t.day++), this.night = !1, …`，以及睡觉跳时 L47644 同样 `this.night = !1`。结果：
  - 刷新规则不区分昼夜（`updateSpawner` 完全不读 night）；
  - gloomwing 夜间暴击、音乐 “night”、夜晚提示“les créatures nocturnes apparaissent…”（L48123）全部不会触发；
  - `nocturnal` 字段无读取；`night` 字段只用于把 aetherling 排除出迁徙事件与发现系统的候选。
  - 【推断】昼夜玩法是被刻意关闭的（开发者在补丁层强制置 false），提示文本是旧版本残留。

4.6 Boss：Aethon, le Titan Creux（**`z9`** L34964、**`L9`** L35037、**`V9`** L35072）

- 进场：`intro = 2.8 s` 不动、不受伤（`N9 → 0`），2.6 s 时咆哮震屏。被限制在竞技场圆内（`dA.r − 3`）。
- 重赛缩放（L47916）：`maxHp × (1 + 0.35n)`，`atk × (1 + 0.15n)`。
- 阶段：`l = hp ≤ 33% ? 3 : hp ≤ 66% ? 2 : 1`；进入新阶段：无敌 2.6 s、取消当前招式、把玩家击飞（速度 14）、banner “Phase II / Phase III：Enragé”。
- 招式序列 `Sd`（L34941）循环：
  - P1：slam, barrage, charge, barrage, slam, charge
  - P2：slam, pillars, barrage, charge, summon, pillars, slam, charge

    - P3：barrage, sweep, pillars, slam, charge, summon, sweep, barrage, charge
- 出招间隔 `cdt`：P1 1.2 s / P2 0.8 s / P3 0.4 s；距离 > 9 m 时以 `spd`（P3 ×1.3）逼近。
- 招式参数：
    - `slam`（quake）：windup 1.5（P3 1.2），radius `7.5 + 阶段`，power 1.25 → 之后 vuln 2.6 s（P3 1.8）。
    - `barrage`（burst）：12 发（P3 16），速度 10，射程 24，power 0.75。
    - `charge`（dash）：windup 1.3（P3 1.0），长 26、速度 20、宽 4.4、power 1.5 → 冲刺结束 vuln 3.2 s（P3 2.2）。
    - `pillars`：5 个（P3 7）半径 3.2 的落雷圈，第一个落在玩家脚下，1.5 s 后结算 power 1。
    - `summon`：召唤 `min(3（P3 4）, 6 − 场上敌人数)` 只 12 级 blightbat / blightcrawler 交替，强制 `provoked = 99` 追玩家 → vuln 1.8 s。
    - `sweep`：从玩家方向 −0.9 rad 起扫 1.8 rad 的 32 m 光束，时长 2.2 s（P3 2.6），每 0.2 s 判定 power 0.55 → vuln 2.2 s。
- 受伤：vuln 期间 ×1.5，其余 ×0.75，入场/阶段无敌 ×0；核心 `glow[0]` 颜色：vuln 金 0xffe08a、无敌 白、平时 0xd6a8ff（L34976–34977）。
- 胜利：首次击败 → `victory`，否则 `rematches++`；奖励 XP 1500（重赛 ×0.5）、掉落 ×(1 + 0.5·rematches)。

5. 动画：GLB 剪辑与程序化驱动

5.1 GLB 家族与剪辑（**`glb_summary.json`** + **`iM`** L22169）

全部 GLB：单材质 “Atlas”（例外 `flying_Dragon` 用 “Material”），无内嵌贴图（共用外部 `atlas.png`，`rM` L22232），glTF-Transform v4.5.1 处理。每个文件恰好 4 个剪辑：

家族	idle	walk/move	attack	hit	骨骼规模	文件（使用者）
big_	`Idle`	`Walk`	`Punch`	`HitReact`	43 joints / 1 skin；Demon、MushroomKing、Orc 为 2 skins / 86 joints	Alien（blightcrawler）、BlueDemon（elite_blightalpha）、Cactoro（elite_mossmatriarch）、Demon（ashmaw, elite_voidstalker）、Dino（cindertusk）、Frog（sparkhopper, blightleaper）、MushroomKing（mossback, elite_hollowking）、Orc（blightbrute, elite_cinderhorn）、Yeti（shardhide）；⚠️缺失：Bunny（carrotail）、Birb（spikebeak）、Fish（blightfish）、Tribal（blighttribal）
blob_	`Idle`	`Walk`	`Bite_Front`	`HitRecieve`（原文拼写）	4 joints；Alien、GreenSpikyBlob、PinkBlob、Wizard 为 2 skins / 8；Mushnub_Evolved 3 skins / 12	Alien（blightspitter）、Cat（emberkit）、Chicken（rainstrider）、Dog（puddlepup）、GreenSpikyBlob（thornstalker, blightthornling）、Mushnub（bloomcap, blightspore）、Mushnub_Evolved（sporelord）、PinkBlob（fluffern）、Wizard（npc_merchant）、Yeti（pebblepaw）；⚠️缺失：Cactoro（pricklehat）
flying_	`Flying_Idle`	`Fast_Flying`	`Headbutt`	`HitReact`	13 joints；Ghost / Ghost_Skull / Hywirl 38；Squidle 43；Dragon_Evolved 46	Alpaking_Evolved（aurelpaca）、Armabee（blightbat）、Armabee_Evolved（hexabuzz）、Dragon（voltcrest, blightdrake）、Dragon_Evolved（pyrelope, elite_cryptwing）、Ghost（gloomwing, blightwisp）、Ghost_Skull（aetherling）、Glub（dewdrake）、Glub_Evolved（maelglub）、Goleling_Evolved（golemurk）、Hywirl（prismoth）、Pigeon

家族	idle	walk/move	attack	hit	骨骼规模	文件（使用者）
						（twigwing）、Squidle（tempestail）；⚠️缺失：Tribal（blightmask）

```
// L22169 iM：按 src 的前缀（big/blob/flying）选正则
blob:   { idle: /^Idle$/,        walk: /^Walk$/,        atk: /^Bite_Front$/, hit: /^HitRecieve$/ },
big:    { idle: /^Idle$/,        walk: /^Walk$/,        atk: /^Punch$/,      hit: /^HitReact$/ },
flying: { idle: /^Flying_Idle$/, walk: /^Fast_Flying$/, atk: /^Headbutt$/,   hit: /^HitReact$/ }
```

注意家族由 `ip.src` 的目录决定而不是物种原型：例如 `golemurk`（heavy）用 `flying/Goleling_Evolved`、`hover .9`，所以走飞行剪辑并悬浮。

加载顺序（`QUALITY.assets` L44198）：关键模型 `CRIT = ["blob/PinkBlob","flying/Pigeon","big/Alien","big/Dino","blob/Wizard"]` 先同步加载（starter、鸟、第一个敌人、shader 预热、商人），其余按 meadow(0) → blight(10) → blightalpha(15) → woods(20) → wastes(30) → 其他(40/50) → 精英(80) 排序后台加载，战斗/Boss 中暂停解析。

5.2 GLB 混合驱动（**`l0`** L22264–22326）

1. 克隆场景（`Ap = SkeletonUtils.clone`），替换为共享图集材质（带 `tint`），按包围盒缩放到 `h`，脚底贴地。
2. 外层包一个 `Gt(h + hover, radius×1.1, hover)` 程序化 rig，GLB 节点挂在 `rig.bob` 下——程序化的 bob/前倾/受击后仰都叠加在骨骼动画之上。
3. 所有 4 个 action 同时 `play()`，只调权重：

```
// L22312–22322（每帧，_ = 归一化速度 |v|/spd ∈[0,1.4]，z = 前摇进度 windup ∈[0,1]）
T = _r(T, Math.min(1, _ * 1.4), 8, N);                          // 速度混合因子（指数平滑，速率 8）
p.setEffectiveWeight(1 - T);                                     // Idle
g.setEffectiveWeight(T), g.timeScale = Z ? .9 + _ * .4 : Math.max(.5, _ * 1.1);  // Walk：飞行 .9+.4v，地面 max(.5,1.1v)
// 攻击剪辑：LoopOnce + clamp，timeScale=0，手动"刷时间轴"
z > 0 ? (B = _r(B, 1, 20, N), E = z * .5 * $, x = 0)             // 前摇：权重→1，时间 = 进度 × 前半段
      : y > .55 && (x = 1, E = .5 * $),                          // 前摇刚结束（上一帧 >.55）：开始出手
x && (E = Math.min($ - .001, E + N * $ * 2.2), E >= $ - .01 && (x = 0)),  // 后半段以 2.2×速率播完
z <= 0 && !x && (B = _r(B, 0, 10, N)), v.time = E, v.setEffectiveWeight(B)
// 受击剪辑：rig.hit > .4 时触发，播放 90% 时长，权重→.85（速率 25），结束后衰减（速率 12）
b && (l.hit > .4 && I <= 0 && (I = b.getClip().duration * .9), I > 0 ? (I -= N, D = _r(D, .85, 25, N), b.time = …) : D = _r(D, 0, 12, N), b.setEffectiveWeight(D))
```

即攻击动画与游戏逻辑的前摇严格同步：前摇 0→100% 对应剪辑前 50%，`fire()` 之后以 2.2 倍速播放后 50%（"出拳"瞬间）。受击剪辑由 `rig.hit = min(1, flash×6)` 驱动，`flash` 在受伤时设为 0.14（L46506），所以每次受击都触发一次。

动画 LOD（`QUALITY.logic.animSkip` L44882）：屏幕外不更新；野生单位按距离每 1/2/3/4 帧更新一次（MEDIUM 22/40/70 m）；队友、Boss、攻击/受击/冲刺/捕获中的单位始终每帧更新。

5.3 程序化 rig **`Gt(height, radius, hover)`**（L21906–21976）

`update(f, m, p)`：`f` = 时间（`syncVisual` 传入 `t + actor.id` 做相位错开，L34379），`m` = 归一化速度，`p` = 前摇进度。可挂载的部件数组：`legs / wings / arms / spin / glow / ears / tailSegs / breath / blink`，以及 `head`、`tail`、`hop`、`flap`、`custom`。

通道	公式（代码原文）	说明
步频相位	`r += g * (this.hop ? 7.5 : 5.5 + v * 5)`，`a = Hr(a, m, 10, g)`，`v = min(1.2, a)`	速度平滑后推进步伐相位
跳跃型 bob（hop，fluffern / sparkhopper）	`y = e +`	sin r
悬浮 bob（hover>0）	`y = e + sin(f·2.2)·0.12`	飞行单位上下漂浮
地面 bob	`y =`	sin r
腿	`hip.x = sin(r+phase)·amp·min(1,1.3a)`；`knee.x = max(0,cos(r+phase))·amp·.35·…`；`ankle.x = −(hip+knee)·.75`	带膝/踝的两骨骼 IK 近似（`On()` L28473 生成）
翅膀	`wing.z = side·(sin(f·flap+.4)·.75 + .2)`；`tip.z = side·sin(f·flap−.9)·.5`	`flap` 默认 14；`ec` 鸟 16、蛾 9（L28785）；`Cx` 6（L29177）；blightbat 18（L29866）
呼吸	`scale = (1+sin(1.9f)·.012, 1+sin(1.9f)·.02, …)`	`breath[]`
待机摇摆	`bob.z = sin(.7f)·.025·R + sin(r)·.03v`；`bob.y = sin(.5r)·.04v`（R = 1 − min(1, 2a)，非 hop 非悬浮）	
尾巴	`seg.y = sin(f(2.4+3v) − .8i)·(.25+.1i)·(.6+.6v)`；单段尾 `sin(3f)·.35`	鞭状传播
耳朵抽动	`D = max(0, sin(.9f + 2.1·side))^24`；`ear.z = side·(.04 + .5D + .25v) + sin(3f+side)·.03`	24 次幂 → 偶发尖峰式抖耳
旋转件	`spin.y = f·1.4`	光环、水晶环
眨眼	`E = (f·.27 + (look>9?0:.37)) % 1 < .03 ? .1 : 1`；`eye.scale.y = Hr(eye.scale.y, E, 40, g)`	周期 1/.27 ≈ 3.7 s，闭眼约 0.11 s；`cn()` 生成的眼睛都进 `blink[]`
注视	`l = Hr(l, clamp(look,−1,1), 5)`；`h = Hr(h, sin(.37f)·.35·R, 2)`；`head.y = .85l + h`	`look` 在 `syncVisual` 中设：玩家 16 m 内（<256）时 = 相对朝向/1.2（L34368–34377）
前摇	`bob.x = −.28p`；`bob.z −= .14p`；`bob.y −= .06p`；`head.x = .35p`；`arm.x = −.75π·p`	后仰蓄力
出手	前摇从 >.55 归零时 `c = 1`，以 4.2/s 衰减；`y = sin(cπ)`；`bob.x += .38y`；`bob.z += .55y`；`head.x −= .5y`	前扑
受击	`u = Hr(u, hit, 24)`；`bob.x −= .35u`；`bob.z −= .22u`；`scale = (1+.06u, 1−.08u−.05p, …)`；`head.x += .5u`	后仰压扁
自定义	`this.custom?.(f, m)`	`Cx`（dewdrake/prismoth）尾链漂浮 L29215；`Sx` 元素体尾部摆动 L29330 等

`syncVisual`（L34363）另外叠加：受击时整体放大 `scale × (1 + flash×.35)`；ko 侧倒（`rotation.z → 1.4`）；dead 旋转 0.6、上浮 ≤0.4、缩小淡出。`glow[]` 仅被 Boss 用于核心变色/脉动（L34976–34978），其余单位的 glow 只是发光材质。

程序化构建器（GLB 不可用时）：

构建器	物种	部件
`qt`（L28511）四足	fluffern、puddlepup、sparkhopper、emberkit、thornstalker、ashmaw、rainstrider、cindertusk、mossback；以及无 case 的 sporelord、	4 条带膝踝腿、耳朵、尾链、呼吸躯干、眨眼眼睛；

构建器	物种	部件
	hexabuzz、maelglub、golemurk、aurelpaca、carrotail、spikebeak、pricklehat（default，L29716）	fluffern/sparkhopper `hop = true`
`ec`（L28778）鸟/蛾	twigwing、gloomwing（moth）、voltcrest（crest）	双翼（带翼尖）、腿、耳羽、hover
`Tx`（L28885）	bloomcap	菌盖、手臂 `arms[]`
`bd`（L28986）	pebblepaw、shardhide（带水晶）	岩甲四足
`Cx`（L29091）	dewdrake、prismoth	悬浮、翅 flap 6、旋转环 + custom 尾链
`Sx`（L29222）	pyrelope、tempestail、aetherling	元素核 + 旋转环 + custom
`Px` case（L29748–30183）	blightcrawler、blightspitter、blightbat（flap 18）、elite_blightalpha/mossmatriarch（共用）、elite_cinderhorn、elite_prismwarden、boss_aethon	其余敌人无 case → default 单个 blob（L30179）

Boss 程序化模型：`Gt(6.5, 2.3)`，双腿、躯干、晶体肩甲、头部、发光核心 `glow[0]`、两条 `arms[]`（参与前摇抬臂/出手挥落）、10 颗环绕旋转水晶（`spin[]`）。

6. 发现的不一致 / 死代码（汇总）

1. `QUALITY.fix.PROC` + `F.prune` 定义了“18 种改回程序化”，但从未调用 → 实际仍是 GLB。
2. 游戏逻辑 `night` 恒为 false → 夜行/夜间加成/夜间刷新全部失效；`nocturnal` 字段无读取。
3. `uM.range`、`uM.windup` 未被读取；敌人行为完全由 `B9` 决定。
4. 大量被动描述未实现（见 §1.4）；tempestail 的实际效果（每次连锁闪电）与描述（雷系 ATK +15%）不符；mossback 实际是抗击退。
5. 诱饵描述 “+20 %”，代码 +0.18；`asleep` 加成（+0.15）无任何设置处。
6. 野生生物的“28 m 外移除”分支被 `&& !1` 关闭。
7. 新增 6 个 GLB（Bunny/Birb/Cactoro(blob)/Fish/Tribal×2）不在提供的资源包中；若缺失，`QUALITY.assets.ready` 会因加载失败（`bad` 集合）回退为程序化 default 模型。

WILDKEEP 世界 / 地图 / 场景系统逆向报告

> 来源：/workspace/work/wk/game.pretty.js（行号均指此文件），英文译名取自 assets/en-AVvZq18N.js。 复现脚本与采样数据：/workspace/work/wk/report/world_data/（worldlib.js 为从 bundle 原样抽出的 25663–26929 行 + 依赖函数，可在 node 中直接 require 运行 TM()/MA()/At()）。 俯视图：/workspace/work/wk/report/world_map_s.jpg（由真实高度函数 MA 和生物群系函数 At 采样生成，0.25 单位/像素）。 标注【推断】的内容为根据代码行为得出的结论，非代码中直接写明。

---

0. 符号速查表（混淆名 → 含义）

名称	行	含义
`Cn = 96`	25775	内陆半宽（
`XA = 140`	25776	世界可玩半宽（外侧为山墙）
`yM = 34`	25777	建造半径（距基地中心）
`dA = {x:0,z:-90,r:15}`	25778	最终 BOSS 竞技场「棱镜圣所」中心
`RA = {x:0,z:58}`	24089	玩家基地「炉石要塞」中心
`gt = RA+(3,13)`	42465	商人（巫师 NPC）位置 (3,71)
`bo = (0,-73)`	41745	竞技场入口光门
`pr = -0.5`	25853	水面高度
`ho/ba/ai`	25784–25805	2D 整数哈希 / 值噪声 / fBm
`Br/Er`	25806–25807	草甸↔灰烬林、灰烬林↔荒原 分界曲线
`At(x,z)`	25809	生物群系判定
`MA(x,z)`	25845	地形高度函数
`cs(x,z)`	25855	坡度（中心差分）
`iu(x,z)`	26443	到道路折线 `EM` 的距离
`TM(seed)`	26455	世界内容生成（资源节点、道具、植被、栖息地）
`ve(seed)`	23907	PRNG（splitmix 风格）
`pd`	25663	2D 空间哈希网格（碰撞/占用）
`Ny`	41055	地形网格构建 + 顶点着色
`zy`	41180	池塘水面
`es / Gy / Hy`	41277 / 41372 / 41396	昼夜关键帧表 / 插值 / 大气（天空、太阳、雾、云）
`M`	21778	程序化几何缓存工厂
`Sn`	21593	多部件合并 + 顶点色烘焙（AO/渐变/噪声/风权重）
`c0 / FA`	21794 / 21820	自定义着色器补丁 / 材质库
`Cy / Jp`	39284 / 39300	资源节点模型注册 / 场景植被模型注册
`fu`	39360	实例化渲染器（3 级 LOD + 视锥/距离剔除）
`hc / Yp`	39669 / 40474	建筑模型 / 世界道具模型

---

1. 世界尺寸、坐标系、地形、水体、生物群系

1.1 尺寸与坐标系

- 右手系 three.js：X 向东、Y 向上、Z 向南。北 = −Z。证据：圣所位于 z=−90，英文文本称其为 "far north of the Crystal Wastes"；游戏地图绘制 `(u+XA)*i` 也是 −Z 在上。
- 原点 (0,0) 位于「守卫之门」(0,−5) 附近；玩家基地在 (0,58)（南部），游戏流程是一路向北推进。
- 尺度：1 单位 ≈ 1 米（教学任务 `ae("travel",15,"Marcher ou courir sur 15 mètres")`，24875 行）。
- 三个嵌套的正方形：
  - `|x|,|z| ≤ 96`（`Cn`）：内陆（草甸 / 灰烬林 / 荒原）。
  - `96 < max(|x|,|z|) ≤ 140`（`XA`）：外圈「破碎之地 Shattered Lands」（代码里叫 `rift`）。
  - `> 134`：山墙抬升；地形网格一直延伸到 ±164（`(XA+24)*2 = 328` 单位见方）。
- 可玩区域约为 280 × 280 单位；内陆约 192 × 192。

```
// 25775
const Cn = 96, XA = 140, yM = 34, dA = { x: 0, z: -90, r: 15 };
```

1.2 噪声与随机数

```
// 25784 整数哈希
function ho(s, A) { let e = Math.imul(s, 374761393) ^ Math.imul(A, 668265263);
  e = Math.imul(e ^ e >>> 13, 1274126177); e ^= e >>> 16; return (e >>> 0) / 4294967296 }
// 25790 双线性值噪声 + smoothstep 插值
function ba(s, A) { ... return _n(_n(ho(e,t),ho(e+1,t),r), _n(ho(e,t+1),ho(e+1,t+1),r), a) }
// 25800 fBm：默认 4 个八度，增益 0.5，频率倍数 2.03
function ai(s, A, e = 4) { let t=.5,n=1,i=0; for(...) i += t*ba(s*n,A*n), t*=.5, n*=2.03; return i }
```

- PRNG `ve(seed)`（23907）：`A += 0x6D2B79F5` 后做 mulberry32 式混合。
- 世界种子：存档里的 `world.seed = 20260404`（24171；28150 行加载时限制在 1–4e9 之间）。地形本身不依赖种子：`MA` 只用固定哈希，所以所有玩家的地形完全一样。种子只影响 `TM()` 中的物件撒点。
- `TM` 里的派生子种子：`s ^ 1912062174`（裂隙区节点）、`s ^ 236915466`（基地附近额外矿脉）、`s ^ 1592594996`（场景植被）。地表装饰用 `ve(777)`，萤火虫等粒子用 `ve(4242)`，星空 `ve(5)`，云 `ve(9)`。
- 模型使用另一套 3D 哈希 `Ei/AM/tp`（21562–21587，频率倍数 2.1），只用于模型顶点扰动和顶点色。

1.3 高度函数 `MA(x,z)`（25845）

```
function MA(s, A) {
  const e = we(-4, 4, A - Br(s)),           // 草甸→灰烬林 混合权重
        t = we(-4, 4, A - Er(s)),           // 灰烬林→荒原 混合权重
        n = (ai(s*.025, A*.025) - .5)*7 + 1,              // 草甸：缓坡丘陵 ±3.5
        i = (ai(s*.035+40, A*.035) - .5)*11 + 1.5,        // 灰烬林：更起伏 ±5.5
        r = 1 - Math.abs(ai(s*.03-20, A*.03+9, 3)*2 - 1), // 脊状噪声
```

```
      a = r*r*11 - 3 + (ai(s*.08, A*.08) - .5)*3;       // 荒原：水晶山脊
  let c = _n(a, _n(i, n, e), t);
  { // 外圈裂隙：另一组脊状噪声，圣所附近不处理
    const l = Math.max(Math.abs(s), Math.abs(A)),
          u = we(Cn-8, Cn+6, l) * we(dA.r+8, dA.r+24, Math.hypot(s-dA.x, A-dA.z));
    if (u > 0) { const h = 1 - Math.abs(ai(s*.04+70, A*.04-30, 3)*2-1),
                         d = h*h*12 - 1.5 + (ai(s*.09, A*.09)-.5)*3.5; c = _n(c, d, u) } }
  for (const l of BM) c = _n(l.h, c, we(l.r0, l.r1, hypot(...)));   // 平台压平
  for (const l of ps) c -= 2.4*(1 - we(l.r*.2, l.r*1.25, hypot(...))); // 池塘下挖
  const o = Math.max(Math.abs(s), Math.abs(A));
  return c += we(XA-6, XA+14, o) * (14 + 10*ba(s*.12, A*.12)), c      // 边界山墙
}
```

- we 即 smoothstep(a,b,x)，_n 即 lerp（23921）。
- 压平区 BM（25832）：基地 (0,58) 内半径 17、外半径 28，高 0.7；圣所 (0,−90) 18/26，高 1.5；守卫之门 (0,−5) 6/12，高 0.5；棱镜之门 (0,−56) 6/12，高 1.0。
- 在 node 中实测（328² 范围，1 单位步长）：高度在 −3.40 至 +34.0 之间；山墙在 |x|或|z|>140 后抬升 14–24。
- 坡度 cs(x,z)：间隔 0.8 的中心差分（25855）。用途：资源撒点要求 cs ≤ 0.75，植被 ≤ 0.8，地表装饰 ≤ 0.62；建造时坡度过大会提示「地形太陡」（26973）。

1.4 水体

- 没有河流。 EM（26399）其实是道路折线，不是河。共 10 条，全部从基地 (0,56) 或主路节点出发，连到各 POI。主路依次经过 (0,56)→(0,−5) 守卫之门→(−10,−34)→(0,−56) 棱镜之门→(0,−73) 竞技场门。
  - iu(x,z) = 到最近线段的距离 + (ba(x*.35,z*.35)-.5)*1.6 抖动（26443）。
  - 地形着色：iu < 0.9~2.5 渐变成路面色；iu 在 2.2–3.8 范围的路肩压暗约 10%。
  - 场景植被要求 iu ≥ 3.4 + 半径，地表装饰要求 iu ≥ 1.5。
- 4 个池塘 ps（25812）：

id	x	z	r	说明
pond1	−38	40	9	镜湖 Mirror Pond
pond2	42	30	7	草甸东侧
pond3	−32	−34	8	灰烬林
pond4	32	−72	7	荒原，「水晶水」配色（uCrystal=1）

- 水面 zy()（41180）：每个池塘一个径向网格（22 圈 × 56 段，半径 1.35r），放在 y = pr = −0.5。逐顶点属性 aDepth = (pr − MA)/2.2。
  - 着色器：两层值噪声法线、菲涅尔天空反射、高光指数 120、浅水泡沫（smoothstep 0.11→0）、深浅水颜色混合。aDepth < −0.02（水面低于地形）的片元会被 discard。
  - 顶点做小幅 sin 波（振幅约 0.035），只在 aDepth>0 的区域生效。

- 池塘在碰撞网格里登记为 `kind:"water"` 的圆，半径 0.82r（26597）。

1.5 生物群系与分界（**At**，25809）

```
const Br = s => -10 + 5*Math.sin(s*.07) + 3*Math.sin(s*.19 + 1),
      Er = s => -58 + 6*Math.sin(s*.05 + 2) + 3*Math.sin(s*.17);
function At(s, A) {
  return Math.max(Math.abs(s), Math.abs(A)) > Cn && Math.hypot(s-dA.x, A-dA.z) > dA.r+14 ? "rift"
       : A > Br(s) ? "meadow" : A > Er(s) ? "woods" : "wastes" }
```

内部 id	法文名 / 英文名	中文	范围（实测）	面积占比*
meadow	Prairie d'Émeraude / Emerald Meadow	翡翠草甸	z > Br(x)，Br 的取值约在 −17.2 至 −2.2 之间（x=0 时 −7.5），向南到 z=96	26%
woods	Bois Cendrés / Ashen Woods	灰烬林	Er(x) < z < Br(x)，Er 的取值约在 −66.3 至 −50.9 之间（x=0 时 −52.5）	12%
wastes	Étendues de Cristal / Crystal Wastes	水晶荒原	z < Er(x)，向北到 z=−96，并包含圣所外扩区域（r<29）	10%
rift	Terres Brisées / Shattered Lands	破碎之地	96 < max(	x

* 按 |x|,|z|<140 网格采样计算。

- 解锁与屏障：存档初始 `unlocked:["meadow"]`（24138）。任务奖励 `unlockBiome:"woods"`（25020）和 `"wastes"`（25061）分别解锁后两个区域；`bossDefeated` 后自动解锁 `rift`（47873）。
  - 未解锁时，`CM()`（26909）沿 Br/Er 曲线每 2.2 单位放一个碰撞圆（r=1.5）。rift 的屏障用 `gp()`：沿 |x|,|z|=97 的方框布置，并在圣所处绕出一个圆弧缺口（26877）。
  - 可视化是 `fc` 类（41700）：一条半透明加色帘幕（woods 橙 0xFF7A45，wastes 0x9F7AEA，rift 0xFF4D6D；高度 6 或 9），每隔 5 个点放一根 `ward` 柱。
- 生物群系还会改变天空、地平线和雾的色调（`Hy.update`，41517）：rift 0xD06A90、wastes 0xB79AF0、woods 0x9A8F78，混合强度分别为 0.42 / 0.35 / 0.22。
- 注意：`loc_prismgate` 在数据里标为 `biome:"woods"`，但 (0,−56) 处 `At` 实际返回 `wastes`（Er(0)=−52.5）。这是数据与函数之间的小不一致，【推断】不影响游戏性。

1.6 地形网格（**Ny**，41055）

- `PlaneGeometry(328, 328, seg, seg)`，其中 `seg = round(画质 terrain * 164/120)`：
  - LOW：150 → 205 段（约 1.6 单位/格）
  - MED：210 → 287 段
  - HIGH：270 → 369 段
  - ULTRA：320 → 437 段（约 0.75 单位/格）
- 顶点 Y = `MA(x,z)`。网格被切成 8×8 = 64 个子网格，共享同一个 position buffer，各自独立 index 和包围球，用于视锥剔除。

- 顶点色按层叠加（41070–41110）：
  1. 草甸：深绿 / 浅绿 / 黄绿 / 湿地 / 土，由多层 ai、ba 噪声驱动混合。灰烬林：灰 / 橄榄 / 炭色。荒原：沙 / 玫瑰 / 石 / 尘。在 Br/Er 两侧 ±5 单位内平滑过渡。
  2. 按坡度 `D = 1 - normal.y` 叠加岩石色；在草甸缓坡上加苔藓。
  3. 道路（iu）、基地周围 6–14 单位的泥土、池塘边湿沙。
  4. 裂隙区 rA/rB/rC 紫红色；山墙用岩石色，`y > 15–21` 时变雪。
  5. 每个树和石头脚下烘焙一个高斯 AO 暗斑（函数 `m()`）。
- 片元着色器（41140）：在 30–120 m 距离内做细节值噪声调亮/调暗、大尺度冷暖色调变化，以及一层交叉条纹（st）模拟草纹。

---

2. 资源节点

2.1 节点类型表 **mr**（25856–26090）

kind	法文 / 英文【译名为推断】	产出	工具	等级	次数	刷新 (s)	半径 r	阻挡
tree	Arbre / Tree	wood×5	axe	1	4	300	0.7	✔
ash_tree	Arbre cendré / Ash Tree	wood×6, resin×1	axe	1	5	360	0.7	✔
branch	Branche tombée / Fallen Branch	wood×2	none	0	1	200	0.4	✘
loose_stone	Pierres éparses / Loose Stones	stone×2	none	0	1	200	0.4	✘
boulder	Rocher / Boulder	stone×6	pick	1	4	320	1.1	✔
ore_vein	Filon de minerai / Ore Vein	ore×5	pick	1	5	240	1.0	✔
berry_bush	Buisson de baies	berry×3	none	0	1	240	0.6	✘
fiber_tuft	Herbe fibreuse	fiber×2	none	0	1	180	0.3	✘
herb_patch	Herbes médicinales	herb×2	none	0	1	280	0.4	✘
sprout	Pousse sauvage	fiber×1 + 种子池 [sunroot, sunroot, glowberry, starmelon]	none	0	1	300	0.4	✘
ember_vent	Fumerolle de braise / Ember Vent	ember_core×1, charcoal×2	pick	2	5	480	1.0	✔
crystal_cluster	Amas de cristal	crystal×3	pick	2	5	420	1.0	✔
prism_spire	Aiguille prismatique	prism_shard×2	pick	2	7	600	1.1	✔
aether_bloom	Fleur d'éther	aether_dust×1	none	0	1	520	0.5	✘

- 每个节点实例会随机缩放 `s = 0.85~1.35`。碰撞半径为 `r·s`，树木再乘 0.55，其余阻挡类节点乘 0.9（26522）。
- 采完后写入 `world.depleted[id] =` 到期时钟。`Wy.sync` 每秒把未到期的节点在实例渲染器里隐藏（41646）。

2.2 撒点算法（**TM**，26455）

- 所有节点共用两个空间网格：`e = new pd(8)` 碰撞网格、`t = new pd(6)` 占用网格。
- 合法性检查 `a(x,z,r)`：
  - 在边界内：|x|,|z| < 137；

    - 避开每个 POI 的禁区（arena 24、ruin 9、garden 12、landmark 5、其余 6）、宝箱（3）、池塘（r+2）、基地（15）；
    - 坡度 `cs ≤ 0.75`；
    - 占用网格未被阻挡。
- 拒绝采样函数 `u(kind, 数量, 生物群系谓词, [xmin,xmax,zmin,zmax])`：最多尝试 `数量×40` 次。
- 基地周围的环形撒点 `h(kind, n, rmin, rmax)`：围绕 RA。

```
// 26553 基地环带（入门资源）
h("branch",10,16,28); h("loose_stone",10,16,28); h("fiber_tuft",8,16,28); h("berry_bush",4,18,30);
h("sprout",4,18,32); h("herb_patch",3,20,32); h("tree",8,20,34); h("boulder",3,24,38); h("ore_vein",3,26,40);
// 26557 草甸 [-92,92]×[-4,92]：tree55 branch14 loose_stone14 boulder14 ore9 berry12 fiber14 herb8 sprout8
// 灰烬林 [-92,92]×[-62,-6]：ash_tree95 boulder12 ore14 herb12 berry9 branch10 ember_vent10 sprout8 fiber12
// 荒原 [-92,92]×[-92,-48]：crystal34 prism12 aether12 boulder12 ore8
// 26742 裂隙区（子种子 s^1912062174，范围 ±136）：crystal46 prism22 aether22 ember16 ore14 boulder14 ash_tree40
        herb6
// 26773 基地额外矿脉（子种子 s^236915466）：ore_vein×7，距离 RA 20–48
```

- 默认种子下实测结果：共 648 个节点。tree 63、ash_tree 135、crystal_cluster 80、boulder 55、ore_vein 55、prism_spire 34、aether_bloom 34、branch 34、fiber_tuft 34、herb_patch 29、ember_vent 26、berry_bush 25、loose_stone 24、sprout 20。完整坐标见 `world_data/world.json` 中的 `nodes` 字段。
- 场景植被（不可采集，子种子 `s^1592594996`）：先用低频噪声 `Z = ai(x*.042+11, z*.042−7, 3)` 划出「林分」，阈值 0.5 / 0.46 以下的区域只保留约 4–5%。
    - 草甸：oak 68% / birch，共 190 次尝试；另有 5 棵固定的大橡树。灌木/蕨 240，花 110，蘑菇/树桩/原木 40，岩石 52。
    - 灰烬林：gnarled 55% / pine 150；枯灌木/暗蕨 150；荧光蘑菇等 60；岩石 50。
    - 荒原：crystal_tree 48，rock_c 70。
    - 裂隙：crystal_tree 90，rock_c 110。
    - 植被不能离基地小于 34、离道路小于 3.4，也不能放在栖息地圆内。
    - 实测共 1333 个（oak 115、crystal_tree 138、fern 139、flowers 110、bush 101 等）。
- 地表装饰 `_y(veg)`（41551，`ve(777)`）：数量 = `9500·veg·142²/98²`，ULTRA 下约 2.5 万株。
    - 草甸：grass_m 66 / flowers 16 / fern 5 / rock 4 / mushroom 1；
    - 灰烬林：grass_ash 40 / grass_dry 24 / fern_dark 14 / …；
    - 荒原及裂隙：grass_crystal 56 / rock 10 / grass_ash 6。
- 世界道具 `props`：
    - 基地 9 段环形断墙（半径 12.5，缺第 2、6 段作为出入口），2 根柱子，1 面旗；
    - 每个 ruin 5 个残墙或柱子；每个 waystone 一个传送石；
    - 每个 camp 3 顶帐篷 + 1 个营火；
    - 灰烬尖塔一根 spire；水晶花园 6 个 crystal_big；方尖碑 1 个；

    - 每个宝箱一个 cache_pad；圣所周围 16 根竞技场柱（半径 17，北侧 ±0.4 rad 留出入口）。
- 栖息地 `habitats`（83 个，实测）：固定点 + 拒绝采样生成的生物栖息地与 Blight（腐化怪）刷怪点。各圈都有 `max` 上限；camp 类栖息地额外带 `camp:true`。

2.3 随机「发现」点（**AB**，42166）

- 玩家每走 `_d=70` 单位，在前方 22–34 单位处随机生成一个发现点，类型为 plant / stash / ore / creature 之一，并有 18 m 高的光柱标记。
- 名称表 `Gd`（42096）：月光花 / 余烬苔 / 棱镜花、探险者藏匿处、稀有矿床。
- 战利品表 `Hd` 按生物群系区分（例如荒原藏匿处：crystal3、prism_shard1、aether_dust2，外加 70 金）。

---

3. 程序化场景模型

3.1 几何工厂 **M**（21778）

所有几何体按参数做 key 缓存（`Yt`）：

- `box / cyl / cone / sph / ico / oct / tor / cap`：three.js 原生几何体。
- `blob(r, detail, amp, seed, freq)` → `eM`：二十面体细分后投影到球面，再按 `tp` 3D fBm 径向扰动 `1 + (fbm−.5)·2·amp`。用于树冠和石块。
- `tube(points, radii)` → `np`：Catmull-Rom 曲线加 Frenet 标架扫掠，半径沿曲线插值，两端封口。用于树干和枝条。
- `leaf` → `tM`：弯曲的双面带状叶片，可选两种外形。
- `tier` → `nM`：圆锥底部按正弦做波浪裙边，用于松树的层叠针叶。
- `lathe`：旋转体。

3.2 合并与顶点色烘焙 **Sn**（21593）

- 输入是部件数组 `{g, p, r, s, c, c2, a, w, f}`。每个部件应用 TRS 后转成非索引几何，`f:true` 时重算法线得到平直着色。最后合并成一个 BufferGeometry。
- 逐顶点颜色 = `c`，若有 `c2` 则按部件内高度做渐变，再乘以：
    - 半球光项 `1 − hemi + hemi·(n.y·.5+.5)·1.15`；
    - 自底向上的 AO（`a`，默认 0.32，在整体高度的 62% 处达到饱和）；
    - 3D 噪声抖动 ±7%。
- 额外写入 `aw` 属性（风权重）= `w · (归一化高度)^1.3`，因此树干底部不摆动、树冠摆动最大。

3.3 材质与着色器（**c0** 21794，**FA** 21820，**Kt** 21535）

- 全部基于 MeshStandardMaterial（vertexColors），通过 onBeforeCompile 打补丁。共享 uniform Kt： uTime, uCam, uPly, uWind(=1), uRimCol(0x9FC4FF), uRimK, uFade, uSunDir。
- 风（wind:true）：

```
float ph = _w.x*0.31 + _w.z*0.23; float sw = sin(uTime*1.5+ph) + 0.5*sin(uTime*2.7+ph*1.9);
transformed.x += sw*aw*0.10*uWind; transformed.z += cos(uTime*1.2+ph*1.3)*aw*0.07*uWind;
```

  相位取自世界坐标，因此实例化渲染时每棵树的摆动不同步。
- 边缘光：rimF = pow(1−N·V, 2.6)，乘以 uRimCol·rim·uRimK，再按朝上程度加权，最后乘 albedo。
- 遮挡淡出（fade:true）：在相机到玩家的连线上、高于玩家头部 0.5 以上、半径 2.4 内的片元，用 Interleaved-Gradient 抖动做 discard。效果是玩家被树挡住时，树冠局部变得「透明」。
- 自发光：totalEmissiveRadiance += albedo·emit。

材质	roughness	metalness	rim	emit	wind / fade
vcol（solid）	0.86	0	0.35	–	–
foliage	0.82	0	0.55	–	✔ / ✔
bark	0.92	0	0.25	–	✔ / ✔
vcolGlow	0.45	0	0.6	0.75	–
crystal	0.16	0.12	1.0	0.45	–
metal	0.35	0.85	0.4	–	–
terrain	0.96	0	0.18	另有细节噪声补丁	–

3.4 树木（38183–38500）

- LOD 参数表 Ra（38190）：

LOD	管道径向段 rad	轴向段 seg	blob 细分 bd
0	8	8	2
1	5	4	1
2	4	2	0

  枝条和叶团数量通过 un(LOD, a, b, c) 按 LOD 递减。

- 橡树 **Op**（38214）：
  - 4 点 tube 树干，半径 0.68→0.22，高约 3.1；
  - 3/2/0 根主枝；LOD0 额外有 3 条板根；
  - 树冠 = 1 个大 blob（r≈1.95，位于 y≈4.55）+ 7/4/2 个小 blob；
  - 配色 cy[s%3]，例如 0x2C6A3C → 0x86BE4C 的竖向渐变；
  - 整体用 foliage 材质，仅 LOD0 投影。
- 白桦 **Wp**：细长 tube（高 6.2），浅色树皮；LOD0 用 7 个黑色小球做树皮斑。
- 松树 **Xp**：高 8.2 的锥形树干，加 7/5/3 层 M.tier 波浪锥。

- 扭曲树 **Zp**（灰烬林）：弯曲树干，4/3/2 根枝条（LOD0 有二级分枝），扁平叶团；LOD0 带 3 颗 glow 材质的余烬发光球。

- 水晶树 **hy**：crystal 材质。

3.5 石头 / 晶体 / 其他

- oi(shape, seed, biome, lod)（38819）：由多个 blob 组合（形状有 boulder / slab / pillar / cluster / flat）。
  - 配色来自生物群系调色板 my（38815）：
    - meadow [0x5C5853, 0xAAA398, 0x4A7A30]（第二项原码写作 11183e3）
    - woods [0x2E2A2A, 0x6A5A52, 0x4A4A30]
    - wastes [0x6A5F8A, 0xCBBCEC, 0x8A7AB0]
  - 非荒原的石头顶部会加一层苔藓 blob（foliage 材质，带风权重）；灰烬林石头额外再加一层。
- 晶簇 vy（38920）：底座 blob + 6~7 根「六棱柱 + 锥尖」晶体。晶体用 crystal 材质（自发光 0.45、rim 1.0）。
- 模型注册：Cy（39284，14 种资源节点，me(n, fn, lod) 生成 n 个变体，树木和石头带 lod1/lod2）；Jp（39300，24 种场景植被，带 collide 碰撞半径）。

3.6 建筑与道具

- 建筑调色板 oA（39510）：timber 0x6B4426、plaster 0xE8DCC0、stone 0x8F8B84、shingle 0xB8553A、thatch 0xD8B869、iron、gold、canvas、glass 等。
- PA(w,h,d,col,pos)：盒子部件的快捷写法。
- hc(type)（39669）构建 10 种可建造建筑：campfire、chest、workshop、shelter、farm、station、training、healing、advanced、beacon。
  - 例如营火：11 块 blob 石头围成一圈，5 根倾斜木柴，3 个树桩凳；另用 ico 半透明粒子模拟烟雾。
  - 建筑数据表 nu（25475）：cost / radius / max / unlockLevel / workerSlots / upgradeCost。
  - 只能建在草甸，且距基地中心不超过 yM=34（26963）。
- Yp(kind)（40474）构建世界道具：ruin_wall、ruin_pillar、banner、waystone（底座 + 拉长 blob 石碑 + 发光符文）、tent、camp_fire、spire、crystal_big、obelisk、cache_pad、arena_pillar、ward（屏障柱）。
- 道具在 Xy 中逐个 clone，之后由 QUALITY.render.mergeProps 合并以减少 draw call。
- 商人 Vd()（41990）：程序化摊位（条纹顶棚 + 柜台 + 货物球），NPC 使用 GLB blob/Wizard（22091 行 npc_merchant: {src:"blob/Wizard"}）。

3.7 实例化与 LOD（`fu`，39360）

- 物件按 `kind|变体` 分组。每组为 LOD0/1/2 各建一个 `InstancedMesh`（每种材质一个），带 `instanceColor`：每个实例有 0.86–1.12 的亮度随机，红蓝通道再各加 ±4%。
- `repack(group, fx, fz)`，以玩家（焦点）位置为中心：
  - 距离超过 `min(cull, far)` → 剔除；
  - 相机前向点积 < −0.25 且距离超过 `minDist`（开阴影时 50，否则 14）→ 剔除（背向剔除）；
  - LOD 选择：距离 > `near·2.3` 用 LOD2，> `near` 用 LOD1，否则 LOD0。
- 只有焦点移动超过 4 单位（或超过 0.5 且距上次 0.6 s 以上）、或相机方向变化（点积 < 0.94）时，才标记脏并重排。
- 被采集的节点用 squash 动画（`compose` 中的缩放和倾斜，持续 0.25 s）。
- 三套实例渲染器：
  - 资源节点 `Wy`：大件 cull 190，小件 110；
  - 场景植被 `Oy`：非碰撞物 cull = far·0.55；
  - 地表装饰 `_y`：cull 70。
- 画质档位（42530 起，`near / far / dNear / dFar / 节点 / 地形 / 雾`）：

档位	near	far	dNear	dFar	nodes	terrain	fog	阴影
LOW	26	110	16	34	90	150	60–150	无
MEDIUM	34	160	–	–	–	210	80–190	1024
HIGH	46	210	–	–	–	270	100–250	2048
ULTRA	54	250	–	–	–	320	120–300	4096

  植被密度系数：LOW 0.25 / MEDIUM 0.6 / HIGH 1 / ULTRA 1.25。移动端另有降级配置。

---

4. 昼夜循环、光照、雾、天气

4.1 游戏时钟

```
// 42464
const Wd = 720,
// 45919
t.clock += A, t.hour += A * (24 / Wd), t.hour >= 24 && (t.hour -= 24, t.day++)
```

- 1 个游戏日 = 720 实际秒（12 分钟），即 1 游戏小时 = 30 秒。新存档从 8:00、第 1 天开始。
- 在营火休息：白天（6–18 点）跳到 20:00，夜间跳到 6:30；跳过的时间会按 5 秒步长补跑模拟（47637）。

4.2 关键帧 `es`（41277）与插值 `Gy`

h	太阳强度 sunI	半球光 hemiI	雾色	说明
0 / 24	0.28	0.55	0x121A38	深夜
4.5	0.25	0.50	0x1C2548	黎明前
6	0.70	0.70	0xE7B58F	日出
8	1.25	0.95	0xCFE5F3	上午
13	1.40	1.00	0xD7EAF7	正午
17	1.15	0.90	0xE8D9BC	下午
19	0.70	0.65	0xE5895A	日落
20.5	0.30	0.55	0x3A3A62	黄昏

- 每个关键帧还带天顶色 top、地平线色 hor、太阳色 sun、半球光天/地色 hemiS/hemiG。
- 太阳角度：`r = (h−6)/12·π`，方向 `(cos r·0.9, sin r, 0.35)`。
- 夜间因子：`night = 1 − clamp(sin r·4+0.5)`。当 `sin r < 0.05` 时改用月光（反向方向，强度 `0.35·night+0.12`）。
- 阴影相机：正交 ±48，以玩家为中心，位置按 texel 对齐防止抖动。
- 天空球（半径 380）：着色器内含两层程序化远山剪影（1D 值噪声脊线）、太阳光晕，500 颗星点，14 朵精灵云（以 1.2 单位/秒向 +x 漂移，超出 ±320 后循环）。
- 雾：`Fog(0xCFE5F3, near, far)`，near/far 由画质档决定（见 3.7 表）。

4.3 【重要】当前版本昼夜视觉被固定

```
// 45892 frame()
this.atmo.update(10, r, this.mode === "menu" ? "meadow" : this.lastBiome, A)
// 45919 updatePlaying()
... t.day++), this.night = !1, ...
```

- 大气渲染写死在 10:00。`game.night` 每帧被强制设为 `false`。
- 因此在这个构建中天空、光照、雾都不会随时间变化。`state.hour` 仍会推进，只在休息逻辑、时钟 UI、存档里使用。
- 依赖 `night` 的功能也就永远不会触发：`gloomwing` 夜间加成、夜间音乐和环境声、夜行生物的筛选。
- 【推断】这是开发者有意关闭昼夜表现（可能为了性能或可读性），`es` 表属于残留代码。

4.4 天气

- 没有天气系统：搜索 rain / storm / weather / pluie，只命中元素属性（storm 元素）和技能文本。灰烬尖塔描述里「即使在雨中也在阴燃」只是文案。
- 环境变化只有以下几种：
  - 生物群系色调（天空、雾的混色，见 1.5）；
  - 云朵漂移；
  - 环境漂浮粒子 `motes`（`Jy`，按生物群系变色，萤火虫式）；
  - 世界事件（`updateWorldEvents`，48031）：玩家 2 级后，每隔 `420 + rand·240` 秒在随机位置触发一次。类型有 `migration`（迁徙兽群）、`rare`

_resource（临时稀有节点）、4 级后加入 elite_hunt（精英猎杀）；持续 360 秒，精英猎杀为 480 秒；地图上用 ❗ 标出。

---

5. 地图、快速旅行、POI、BOSS、基地与商人

5.1 世界地图 UI

- buildMapBase()（49127）：在 XA·2·2 = 560² 的画布上，每像素调用 At 取生物群系纯色，并按 MA 高度调亮度。生物群系配色：meadow (88,160,92)、woods (124,92,66)、wastes (112,98,176)、rift (120,70,100)。池塘画成蓝色圆盘。
- 打开地图时，drawWorldMap() 缩放到 440² 画布：
  - 未解锁的生物群系压暗到 35%；
  - 只显示已发现的 POI，图标为 🏰 keep / ⛺ camp / ☠ lair / ⛩ waystone / 🔮 arena / ✦ 其他；
  - 玩家建筑画成黄色方块；
  - 世界事件 ❗；
  - 玩家位置用带朝向的箭头表示。
- 没有常驻小地图（minimap）：搜索 minimap / radar / compass 均无结果。地图仅是 UI 面板。
- POI 发现：进入 POI 半径的 0.75 倍范围即记为已发现，同时给 25 XP（lair 和 arena 给 15）并显示横幅（updateLocations，48065）。

5.2 快速旅行（**travel**，47658）

```
pl = ["loc_keep", "loc_wardgate", "loc_prismgate"]   // 26398
travel(A){ if(!locations.includes(A) || !pl.includes(A)) ...; if (inCombat||bossLocked) ...;
  this.pr.x = t.x + 2.5, this.pr.z = t.z + 3 ... }
```

- 只有 3 个传送点：基地 (0,58)、守卫之门 (0,−5)、棱镜之门 (0,−56)。必须先发现，战斗中不可使用。传送后落点偏移 (+2.5, +3)。

5.3 POI 全表 **En**（26092 起）

「实测高度」一列为在 node 中用 MA 计算的结果。

id	英文名	中文	x	z	r	群系	类型	实测高度
loc_keep	Hearthstone Keep	炉石要塞（玩家基地）	0	58	22	meadow	keep	0.70
loc_tower	Old Watchtower	旧瞭望塔	−52	72	10	meadow	ruin	0.29
loc_pond	Mirror Pond	镜湖	−38	40	14	meadow	landmark	−1.71
loc_shrine	Sunken Shrine	沉没神龛	52	50	10	meadow	ruin	−0.36
loc_camp1	Scavenger Camp	拾荒者营地	58	16	12	meadow	camp	−0.40
loc_matriarch	Thornwall Hollow	荆墙洼地	−48	12	14	meadow	lair	1.62
loc_wardgate	Ward Gate	守卫之门（传送）	0	−5	10	meadow	waystone	0.50
loc_cinder	Cinder Hollow	余烬洼地	34	−38	16	woods	lair	−0.28
loc_hunters	Hunters' Camp	猎人营地	−46	−22	12	woods	camp	1.83
loc_ashstone	Ashen Spire	灰烬尖塔	−10	−34	10	woods	landmark	2.53

id	英文名	中文	x	z	r	群系	类型	实测高度
loc_prismgate	Prism Gate	棱镜之门（传送）	0	−56	10	woods*	waystone	1.00
loc_warden	Warden's Perch	守望者栖所	−30	−76	14	wastes	lair	6.84
loc_garden	Crystal Garden	水晶花园	−52	−66	14	wastes	landmark	6.39
loc_obelisk	Shattered Obelisk	破碎方尖碑	52	−80	10	wastes	ruin	3.63
loc_rift_pillar	Breach Pillar	裂隙之柱	108	0	10	rift	landmark	6.69
loc_rift_citadel	Fallen Citadel	陷落城堡	−112	−40	12	rift	ruin	5.64
loc_rift_glass	Glass Lake	玻璃湖	110	62	14	rift	landmark	10.46
loc_rift_lair	Echo Den	回声巢穴	112	−96	14	rift	lair	1.74
loc_rift_camp	Fallen Camp	堕落者营地	−108	80	12	rift	camp	2.97
loc_rift_pit	Void Pit	虚空深坑	12	122	12	rift	ruin	−0.09
loc_rift_altar	Altar of Dead Stars	死星祭坛	−62	−118	12	rift	ruin	2.47
loc_sanctum	Prism Sanctum	棱镜圣所	0	−90	20	wastes	arena	1.50

* 见 1.5 末尾的不一致说明。

- 宝箱 **`gr`**（26290 起，共 14 个，均在对应 POI 旁偏移约 2 单位）：
  - tier1：tower、shrine、camp1；
  - tier2：hunters、ashen；
  - tier3：garden、obelisk、warden，以及 6 个裂隙宝箱。
- 另有一个荒原营地刷怪点 `camp_wastes`，位于 (46,−66)，不是 POI。

5.4 精英 BOSS（**`mp`**，26374）与最终 BOSS

精英	位置	所在 POI
elite_mossmatriarch（Thistle Matriarch）	(−48, 12)	荆墙洼地
elite_cinderhorn（Cinderhorn Alpha）	(34, −38)	余烬洼地
elite_prismwarden（Prism Warden）	(−30, −76)	守望者栖所
elite_voidstalker（Void Stalker）	(112, −96)	回声巢穴
elite_cryptwing（Cryptwing）	(12, 122)	虚空深坑
elite_hollowking（Hollow King）	(−62, −118)	死星祭坛

- 最终 BOSS 竞技场：圈心 `dA = (0,−90)`，r = 15，周围 16 根竞技场柱位于半径 17 处。
  - 入口光门 `bo = (0, −73)`：一块 13×9 的半透明面板，点亮 Beacon（`world.beacon`）后打开。
  - 玩家进入 r−3 范围且 `z < −76` 时触发 `startBossFight()`（47897）：在 z=−73 生成 8 个封印碰撞体，在 (0,−96) 生成 `boss_aethon`。
  - 重赛时 HP 乘 `1 + 0.35n`、攻击乘 `1 + 0.15n`。
- 「地牢」：代码中没有独立的地牢场景，所有内容都在同一个开放世界里。最接近地牢的是：
  - 外圈 Shattered Lands（rift）：击败 Aethon 后开放；受伤倍率等受「每日裂隙」词缀影响。
  - 每日裂隙 **`Mn / Ui`**（42400–42463）：每天由日期种子 `ve(zo(day)*40503)` 从 4 种词缀中选一种——Opulent（金币和 XP ×1.6）、Furious（受伤 ×1.25，掉 essence）、Calm（捕捉率 +15%）、Mineral（矿石

+40%）。目标 `Ln = 10`【推断为击杀数】，奖励 `Zo`：250 金 + essence×6 + orb×1 + aether_dust×4。

5.5 玩家基地与巫师商人

- 基地：炉石要塞，`RA = (0, 58)`，位于草甸南部。
  - 地形在内半径 17、外半径 28 范围压平到 0.7；
  - 9 段断墙环（半径 12.5，缺口朝向角度 2/9·2π 与 6/9·2π）；
  - 出生点 / 复活点在 (0, 64)，初始朝向 π（面北）；
  - 建造范围为距中心 34 以内的草甸。
- 巫师商人：`gt = {x: RA.x+3, z: RA.z+13, rot: π}` → (3, 71)，在基地南侧。
  - 模型为程序化摊位 + GLB `blob_Wizard`；碰撞半径 1.5；
  - 玩家距离 < 80 时才更新动画，距离 < 9 时会转头看玩家；
  - 商店内容是外观（帽子、披风，例如 `hat_wizard`「巫师帽」、`hat_crown` 1200 金），见 41853 起。

---

6. 俯视图说明（world_map_s.jpg）

- 采样范围 $[-164,164]^2$，分辨率 1312^2 像素（0.25 单位/像素），北（−Z）在上。
- 底色：按生物群系着色，叠加高度调亮度、西北光源的阴影，以及山墙和积雪。
- 叠加层：
  - 道路：`iu < 1.5` 处为浅棕色；
  - 池塘：`MA < pr` 处为蓝色；
  - 方框：世界边界 140（白）、内陆边界 96（粉）；
  - 屏障线：Br（橙）、Er（青）、裂隙墙 `gp`（红）；
  - 648 个资源节点（按种类着色，见图例）；
  - 14 个宝箱；22 个 POI（圆圈为发现半径，标注中英文名和坐标）；
  - 商人与竞技场门位置。
- 重新生成方法：

```
cd /workspace/work/wk/report/world_data
node dump.js && python3 render.py
```

WILDKEEP 玩家与核心玩法系统逆向分析

> 分析对象：`/workspace/work/wk/game.pretty.js`（共 49,771 行；第 1–21500 行左右是 three.js r170，游戏逻辑在后面） 字符串翻译：`/workspace/work/wk/assets/en-AVvZq18N.js`（法语 → 英语字典，源码里的文案都是法语） 配套数据（由本次分析用 node 直接对源码切片求值导出，不是手抄）：
>
> - `/workspace/work/wk/report/data/tables.json`：物品 `NA`、配方 `Wo`、建筑 `nu`、主线/支线任务 `ui/fp`、作物 `va/hp`、信标献祭 `Go`
> - `/workspace/work/wk/report/data/tables_readable.txt`：上面这些表的可读版本（带英文名）
>
> 记号说明：`Lxxxx` 表示 game.pretty.js 里的行号。标【推断】的地方是从代码行为推出来的，不是代码直接写明的。 补充一点：bundle 里有一大段手写、未压缩、带法语注释的补丁模块 `QUALITY`（L42470–45448，包含 `QUALITY.fix`、`QUALITY.save`、`QUALITY.logic`、`QUALITY.assets` 等），它改写了原版的很多行为，比如移动端锁定、首个同伴减伤、自动存档。下文提到时会注明。

---

0. 主要代码地图

模块	位置	说明
`class iB`（游戏主类）	L45449–48305	持有 state/scene/camera/input；包含 `loop/frame/updatePlaying/updatePlayer/meleeAttack/...`
`class G9`（输入）	L35141–35328	键盘、鼠标、Pointer Lock、滚轮
`class $9`（触屏 UI）	L35870–36038	虚拟摇杆、相机拖动区、按钮
`class hB`（DOM UI）	L48736 起	面板、商店 `shopHTML` L49181、帮助 `helpHTML` L49230
`function dl()`（新存档模板）	L24106	state 结构
`NA`（物品表）	L24184–24757	用 `JA(ZA({...}))` 注册，共 56 个
`va / hp`（作物）	L24758–24779	
`ui / fp`（主线/支线任务）	L24871 / L25130	`class fd` 任务管理器 L25394
`it(s)`（玩家属性推导）	L25273	
`nu`（建筑表）	L25475	`ut = byId`
`class pd`（空间哈希碰撞网格）	L25663	
`mr`（采集节点表）	L25892	
`bp / PM / RM / DM / kM`（放置/建造/拆除/升级）	L26941–27131	
`gl`（建筑/工人模拟 tick）	L27433	
`Wo`（配方表）	L27637	`Sa = byId`
`Ys`（伤害公式）	L27896	
`vx`（读档校验/清洗）	L28008	`Bx`（存档器）L28287
主角模型 `Cd → M9`，失败时回退 `Rx`	L33579 / L33587 / L30205	
`uM / rp`（敌人表）	L23272–23905	
Y8 SDK 封装 `yn`	L47–145	

模块	位置	说明
启动入口 `dB()`	L49724	

---

1. 操作、玩家控制器与相机

1.1 键盘和鼠标绑定（**`G9.kd/md`**，L35185–35246）

```
// L35199-35209  UI 键 → onUi(...)，在 input.blocked 时也有效
const t = { KeyI:"inventory", KeyC:"creatures", KeyM:"map", KeyJ:"quests", KeyB:"build",
            Escape:"pause", KeyP:"pause", KeyH:"help", KeyK:"craft" };
// L35217-35244  游戏动作（边沿触发 edge），只在 !blocked 时生效
Space→"jump"  KeyE→"interact"  KeyF→"capture"  KeyX→"heal"  KeyQ→"device"
KeyR→"rotate" Tab→"target"     Enter→"confirm"
// L35189  鼠标：左键 → edge+held "attack"；右键 → edge "sling"
```

动作（y8 页面上的名称）	按键	内部名	备注
Move	W/A/S/D、方向键（帮助里写 ZQSD，因为代码按 `code` 物理键位识别，AZERTY 键盘上同样可用）	`axis()`	L35272：前进/后退累加到 `z`，左右累加到 `x`，向量长度超过 1 时归一化
Look Around	鼠标移动	`lookX` / `lookY`	只有在 Pointer Lock 状态或按住任意鼠标键时才累加（L35151）。第一次点画布会请求 Pointer Lock（L35187）
Attack	鼠标左键	`attack`	按住时，只要 `atkCd<=0` 且处于锁定/触屏状态就会自动连击（L46261）
Sling	鼠标右键	`sling`	
Sprint	左/右 Shift	`axis().sprint`	触屏：» 按钮，或摇杆推到 >0.92 自动冲刺（L35277）
Jump	Space	`jump`	
Interact	E	`interact`	对资源节点按住 E 会按 0.38 s 的冷却重复采集（L47341）
Throw（扔捕获装置）	F	`capture`	
Change Device	Q	`device`	建造时 Q 和 R 都能旋转 45°（L47752）
Rotate while building	R	`rotate`	
Healing	X	`heal`	建造时 X 改为取消放置（L47752）
Lock Target	Tab	`target`	默认行为被 `preventDefault` 屏蔽
Inventory / Creatures / Craft / Build / Journal / Map	I / C / K / B / J / M		
Pause	Esc / P		丢失 Pointer Lock 时也会自动打开暂停（L45586 `onLockLost`）
Help	H		
Fullscreen	不在 G9 里，是 UI 按钮【推断：DOM 工具栏 `#tools`】		
相机缩放	滚轮		`cam.dist ± 0.8`，限制在 [3.5, 12]（L45586）

窗口失焦（blur）时会清空所有按键（L35146）。

1.2 触屏操作（**$9**，L35870；补丁 **QUALITY.fix**，L43493–43968）

- 判定为触屏：`matchMedia("(pointer: coarse)")`，也可以用 URL `?touch=1` 强制开启、`?touch=0` 强制关闭（L35143–35146）。
- 左侧 **`.joy-zone`**：浮动摇杆，按下的位置就是摇杆中心，半径 52 px，死区 0.12（L36003–36010）。
- 右侧 **`.cam-zone`**：拖动转视角，`addLook(dx*1.6, dy*1.6)`（L35932）。
- 圆形按钮：⚔️ 攻击（按住会连击）、🪃 弹弓、⤒ 跳跃（补丁 CSS 里被隐藏：`body.touch .mb-jump{display:none}`）、E 交互、🕸️ 捕获、🩹 治疗、🔁 换装置、🎯 锁定、» 冲刺（按住生效）。按钮角标显示装置/石子/治疗品的数量（`setContext`，L36022）。
- 菜单行：🐾 🎒 🔨 📜 🗺️ ☰。
- 双指捏合缩放：`F.pinch`（L43571），范围 `zoomMin 4 – zoomMax 12.5`，存到 `localStorage.wk_camzoom`（L44926–44942）。另外还有 `+` / `−` 按钮，长按 380 ms 后每 110 ms 重复一次。
- 点地移动模式（设置项 `tapMove`）：点击判定为 `TAP {ms:320, px:16}`（L43898）。开启后隐藏摇杆、左侧区域也变成转视角（`F.tm`）。`updateTapMove`（L46162）把移动写入 `tapAxis`，距离大于 18 m 时自动冲刺（L46207，有滞回：已在冲刺时阈值降到 14）。
- 触屏专属的战斗辅助：
  - `F.alock`（L43708）：攻击或射弹弓时如果没有锁定目标，自动锁定最近的敌人（不需要朝向）。
  - `F.tapSel`（L43725）：点屏幕上的敌人（52 px 半径内）锁定它，再点一次取消。
  - `F.lunge/F.mag`（L43755–43776）：锁定目标超出攻击距离时，向它冲刺，速度 16 m/s，最长 0.22 s。
  - `F.rb`：对已锁定的目标攻击距离加 1 m。
  - `F.cone`：捕获的朝向锥要求从 0.35 放宽到 -0.3，几乎不用瞄准。
  - 同伴/敌人的 aggro 也会被等级修正：`F.AGGRO_LV=[3,5]`，玩家低于 3 级时普通野怪完全被动，3–4 级时侦测半径减半（L43683）。

1.3 玩家控制器（**updatePlayer**，L46217–46293）

玩家状态对象 `pr`（L45476）：`radius:.45, yaw:π, grounded, vx/vy/vz, atkCd, combo, comboT, swingT, stamDelay, shield, hurtT, inCombatT...`

速度和加速

```
// L46249-46253
t.sprinting = c.sprint && g && n.player.stam > 3;
const v = 5.2 * i.speedMult * (t.sprinting ? 1.62 : 1) * (t.swingT > .12 ? .75 : 1),
...
R = Math.min(1, (t.grounded ? 14 : 4) * A);   // 指数逼近：地面 14/s，空中 4/s
t.vx += (b - t.vx) * R, t.vz += (w - t.vz) * R
```

参数	值
步行速度	5.2 m/s × `speedMult`（迅捷护符为 1.12）

参数	值
冲刺速度	×1.62，即 8.42 m/s
挥砍期间（swingT>0.12）	×0.75
移动方向	相对相机 yaw；角色朝向按 `16·dt` 的速率转向移动方向（L46262）
世界边界	`XA=140`，玩家被限制在 ±138（L46273）
Boss 竞技场	被锁在圆心 `dA={x:0,z:-90}`、半径 `r:15–1.2` 的圆内（L46274）

跳跃和重力

```
// L46289
r.pressed("jump") && t.grounded && !this.placing && (t.vy = 8.4, t.grounded = !1,
   n.player.stam = Math.max(0, n.player.stam - 6), ... ), t.vy -= 24 * A, t.y += t.vy * A,
```

- 起跳速度 8.4 m/s，重力 24 m/s²：跳跃最高约 1.47 m，滞空约 0.70 s（由公式计算）。跳一次消耗 6 点体力。体力为 0 也能跳（只是被钳到 0）。
- 落地判定：`t.y <= MA(x,z)` 时吸附到地面。如果原本在地面、`vy<=0` 且离地不到 0.55 m，也会贴地（下坡不会“飞”出去）。以 `vy<-6` 落地时有尘土和脚步声。

地形坡度阻挡（不是物理引擎，是逐轴测试）

```
// L46263-46270
const x = (Y,_,z,N) => { const $=Math.hypot(z-Y,N-_); if($<1e-4) return !0;
   const J = MA(z,N); return t.y > J - .5 || (J - MA(Y,_)) / $ < 1.5 };
x(B,y,T,y) && (B = T), x(B,y,B,E) && (y = E);   // X、Z 分别测试，可贴墙滑动
```

当坡度（高差/水平距离）≥ 1.5，且玩家没有高出目标地面 0.5 m 时，这一轴的位移被拒绝。

碰撞：之后调用空间哈希网格 `world.grid.resolve(x, z, 0.45)`（`class pd`，L25737）。所有障碍都是 2D 圆，迭代 3 次，按穿透深度把玩家推出去：

```
// L25745-25750
const d = o.r + t; if (h < d) { const f = (d - h) / h; A += l * f, e += u * f, ... }
```

网格单元大小 8 m（默认构造参数），`kind` 有 `water / barrier / ground / building / prop` 等。玩家不按 kind 过滤，碰撞所有类型。

体力

```
// L46289 (后半)
t.sprinting && g ? (stam -= 20*A, t.stamDelay = .9)
 : t.stamDelay > 0 ? t.stamDelay -= A : stam = Math.min(i.maxStam, stam + 26*A)
```

- 最大体力 `100 + 3·(lv−1)`（L25281）。
- 冲刺消耗 20/s；需要体力 > 3 才能开始冲刺。
- 停止冲刺 0.9 s 后开始恢复，速度 26/s。

脱战回血：没有 `inCombatT` 且 `night===false` 时，每秒回复 `0.4% maxHp`（L46292）。`updatePlaying` 每帧都会写 `this.night=!1`（L45918），所以这个条件恒成立【推断：夜晚机制被关掉了，或者是残留代码】。

1.4 相机（**updateCamera**，L46415–46455）

第三人称轨道相机，完全手写，没有用 OrbitControls：

```
// L46419-46421
r = .0026 * n.sens;
e.yaw -= i.dx * r, e.pitch = ze(e.pitch + i.dy * r * (n.invertY ? -1 : 1), -.12, 1.3)
```

参数	值
初始值	`cam = {yaw:π, pitch:.38, dist:7.5}`（L45504）；触屏 pitch 0.56，dist 为 8（竖屏 8.6）或读取保存的缩放值
灵敏度	0.0026 rad/px × `settings.sens`（读档时限制在 0.2–3）
俯仰范围	−0.12 到 1.3 rad；支持 Y 轴反转
注视点	玩家 y+1.55，高度方向平滑：差值大于 2 m 时系数 40，否则 7（L46424）
FOV	62°（触屏竖屏 66，桌面竖屏 74），冲刺 +5°，`fovKick·9`（捕获/命中特效），按 4/s 逼近
防穿墙	从玩家向相机方向采样 8 个点，用 `grid.blocked(...,0.25,["water","barrier","ground","building"])` 检测，碰到就把距离缩到 `max(1.8, dist·(R−1.4)/8)`。拉近的平滑系数 14，拉远 3
地面约束	相机高度不低于地形 + 0.7
震屏	在 `frame()` 里给相机位置加 `fx.shake·0.35` 的随机偏移（关闭"减少动态"时）；触屏的 `shakeScale` 为 0.55

锁定目标（Lock-on）（`updateLock`，L46990）：

- 按 Tab 时，把 26 m 内的有效目标按距离排序，依次切换；切到列表末尾后变成 null（取消锁定）。
- 目标距离超过 34 m、死亡或正在被捕获时自动解锁。
- 锁定不会转动相机（代码里没有任何根据 lock 修改 `cam.yaw` 的地方）。锁定只影响：`autoAim` 直接返回锁定目标、捕获目标优先、同伴 AI 的目标（`R9(..., this.lock)`，L46842）、脚下光环 `lockRing` 的显示（L48137，creature 为黄色、敌人为红色）。

自动瞄准 **`autoAim(range, yaw, cone)`**（L46295）：如果锁定目标在 `range+4` 内就直接用它；否则在 `|角度差|≤cone` 的扇形里找最近的敌人（贴身 1.2 m 内忽略角度）。近战用 `range = 攻击距离+1.2`、`cone = 0.9`；弹弓用 24 m、0.35。

---

2. 主角模型与动画

2.1 模型来源（**Cd → M9**，L33579–33818）

- 模型不是程序化拼出来的。它是一个内嵌的 FBX 二进制：`d9` 是 base64 字符串（L32784，解码后 167,212 字节，文件头 `Kaydara FBX Binary`），由 `e9`（FBXLoader，L31246）解析。
- 骨骼节点（从 FBX 里提取）：`Root, characterMedium, Hips, Spine, Chest, UpperChest, Neck, Head, Left/RightShoulder, Arm, ForeArm,`

Hand, UpLeg, Leg, Foot, Toes，另有 IK 控制器 LeftFootIK / KneeCtrl / HeelRoll ...。【推断：characterMedium 这个命名和 Kenney 的 “Animated Characters” 系列一致】
- 缩放：w9 / 376，w9 = 1.8，所以模型身高是 1.8 m（逻辑高度写成 1.98，L33658）。
- 材质：kn（MeshStandard）加 c0 包装出 rim 光（{rim:.5, key:"hero"}）。贴图从 Lp 里取，Lp 是 4 张内嵌 PNG（L33318）。外观 **outfit** 用 **Math.abs(s) % 3** 选 3 张贴图里的一张（L33596）。hair 参数在 M9 里没有被使用，只在程序化回退模型 Rx 里用来选发色（wd[A % wd.length]，L30212）。
- 回退模型：M9 抛异常时调用 Rx(s, A)（L30205），它是纯程序化拼的低模（qA 合并 box/sph/tube 等图元，颜色取自 Xo=[12858671, 14256430, 14246042]，L30187）。

2.2 装饰品：帽子 / 披风 / 染色（**setCosmetics**，L33704）

- 帽子挂在 Head 骨骼上的锚点 E；披风挂在 UpperChest 的锚点 B。
- 帽子 p9(id)（L33320）：程序化图元（M.sph / M.cyl / M.box / M.cone / M.tor），每顶帽子写在一个 switch 分支里。例如 hat_cap 由压扁的球、圆柱和一块倾斜的帽檐 box 组成。
- 披风 g9(id)（L33478）：一个 torus 领圈，加一块 0.46×0.78×0.02 的 box 布片（e.panel）。颜色来自 m9[id]。capeUpdate 根据移动速度把布片的 rotation.x 平滑到 0.1 + mv·0.55（L33717）。
- 染色 tint：v9(png, hueShift)（L33506）。在 canvas 上逐像素做 HSV 色相旋转：跳过饱和度 < 0.28、明度 < 0.2 的像素，并跳过肤色区间（色相 6–42°、饱和度 < 0.72、明度 > 0.45）。偏移量 Bd = {tint_forest:110, tint_dusk:230, tint_ember:35, tint_ocean:170}（L33541）。
- 装饰品数据 ra（L41828），在商店里用金币购买，见 §5.5。
- 武器模型 Sp(id)（L30772）：程序化图元（tube、box、cone），挂在 RightHand 的子节点 weaponMount 上（位置在 RightHandIndex1 的 0.6 倍处，朝向偏移 Euler(1.25,0,0)）。

2.3 动画逻辑（**w.update(time, moveSpeedNorm, attackProgress)**，L33735）

基础层用 AnimationMixer 播放 3 个内嵌片段 f9：idle（1.067 s）、run（0.667 s）、jump（0.5 s，只播一次并停在最后一帧）。

```
// L33737-33739
I.run = qi(.06,.45,tA)            // smoothstep 混合 idle↔run
u.run.setEffectiveTimeScale(Co(tA*.95,.35,1.6))
u.idle.setEffectiveWeight((1-I.run)*(1-I.air)); u.run...(I.run*(1-I.air)); u.jump...(I.air)
```

离地时重置并播放 jump；落地后有 0.22 s 的 landT 下蹲。

叠加层是程序化骨骼覆盖：先把 Spine / UpperChest / Head / 两臂的 mixer 结果存下来，再在 UpperChest 空间里用 slerp 叠加手臂姿态（`_()` 辅助函数，L33733）。输入是 `$ = 1 − swingT/0.28`（攻击进度 0→1），分成三段：`C = qi(0,.32)` 抬手、`L = qi(.3,.62)` 挥出、`eA = qi(.62,1)` 收招。

rig.ctx.act	触发	姿态特点（L33760–33775）
"melee" combo 0 / 2	第 1、3 刀	右臂从 −2.65（第 3 刀 −3.0）高举后下劈，躯干先后仰再前压。第 3 刀前压更深（`ee = .5+.3`）
"melee" combo 1	第 2 刀	横扫：`$A`（躯干扭转）从 −1 到 +1.05
"ranged"	弹弓	右臂抬到 −2.9 后回放 1.5，叠加 `sin(Q·5π)` 抖动，表现甩绳；左臂前伸
"gather"	采集	双臂同步高举后砸下（斧/镐动作）
interact>0	按 E	0.5 s 的 sin 曲线伸手
受击 hurtT 上升		`I.hit=1`，按 3.2/s 衰减，躯干后仰、手臂缩回
死亡		`I.die → 1`，身体前倾、双臂下垂。此外 `rig.root.rotation.z` 按 `(deathT−.35)·2.6` 倒地，最大 1.45 rad（L46229）

投掷（F 扔捕获装置）没有单独的 act：`startCapture` 只设置 `pr.swingT = .25` 并转向目标（L47176），沿用上一次的 **`rigAct`**（melee/ranged/gather）【推断：投掷看起来就是复用了上一次的挥砍或甩弓动作】。装置本身沿一条抛物线飞行（见 §3.6）。

受伤后的闪烁：`hurtT > 0` 时按 24 Hz 显隐，`hurtT > 0.45` 或刚出生的 `noBlinkT` 期间不闪（L46292）。

---

3. 战斗

3.1 玩家属性（**`it(s)`**，L25273）

```
maxHp: 100 + 12*(lv-1) + charm.maxHp,
maxStam: 100 + 3*(lv-1),
atk: 8 + 1.6*(lv-1) + weapon.atk,
def: .8*lv + armor.def,
speedMult: 1 + charm.speed,  captureBonus: charm.captureBonus
```

3.2 伤害公式（**`Ys`**，L27896）

```
function Ys(atk, def, mult, elem, rnd, critChance = .08) {
  const crit = rnd() < critChance,
        a = (atk*mult - def*.5) * elem * (.9 + rnd()*.2) * (crit ? 1.5 : 1);
  return { dmg: Math.max(1, Math.round(a)), crit } }
```

- 元素克制 `ga`（L23197）：克制 ×1.5，被抵抗 ×0.7。克制环：Verdant > Stone > Storm > Tide > Ember > Verdant，Arcane 为中立（帮助文本 L49232 有说明）。
- 玩家近战和弹弓不吃元素加成（elem 固定传 1，L46340）。元素只作用于同伴和敌人之间。

3.3 近战与连击（**meleeAttack**，L46313）

参数	值
攻击距离 r	2.7 m；ash_spear 和 crystal_glaive 为 3.5 m
冷却 atkCd	默认 0.38；iron_blade 0.33；crystal_glaive 0.40；ash_spear 0.42
连击	`combo = comboT>0 ? (combo+1)%3 : 0`，连击窗口 `comboT=.95 s`。倍率 `[1, 1.05, 1.55]`
挥砍时长	`swingT=.28`，期间移动速度 ×0.75
命中判定	距离 ≤ `r + 目标半径 (+1 触屏锁定)`，且相对朝向的 `
暴击	10%（`Ys(..., .1)`），×1.5
击退	4.5，第 3 刀 +4；暴击 ×1.5；对精英 ×0.6、Boss ×0.2；再乘 `1 + min(.6, dmg/maxHp·2)`（L46525）
向前冲步	桌面每刀 `vx += sin(yaw)·2.2`（`F.lunge`，L43755）
顿帧	命中时 `hitStopT`：普通 0.03、大伤害（≥20% maxHp）0.05、暴击 0.08。期间 `timeScale = 0.05`（L46517）
击晕	一次伤害 ≥ 30% maxHp 时，普通敌人和野生 creature 被打断前摇并进入 `stunned`（L46530）

特殊规则：野生 creature 只要不是被玩家打的，伤害会被限制成至少留 1 HP（L46505）。也就是说同伴不会打死可捕获的生物，只有玩家能打死。

3.4 弹弓（**slingShot**，L46344；**updateProjectiles**，L46770）

- 前提：背包里有 `sling`（工具，不占武器槽），并且消耗 1 个 `pebble`。冷却 0.55 s。
- 弹丸参数（L46360）：`speed 28, range 26, radius .5, atk = 玩家 atk·0.8 + 3, power 1, crit .1, element "arcane"`（只用于特效，伤害不吃元素）。
- 弹道没有重力，是直线：每帧 `x += dx·speed·dt`。弹丸高度 = 地形高度 + 出生时的离地高度 `h`（约 1.3 m），也就是贴着地形起伏飞（L46775）。
- 碰到 `water / barrier / ground` 网格圆、超出射程或越出世界边界时消失。命中判定是 2D 距离 < `radius + 目标radius`，且高度 `y < 目标y + height + 0.8`。
- 自动瞄准：24 m、±0.35 rad 扇形，没有目标就沿相机 yaw 射出。

3.5 受伤、无敌帧、护盾（**hurtPlayer**，L46464）

- 进入条件：`!alive || hurtT > 0 || victoryT > 0` 时直接 return。所以无敌帧就是 **`hurtT`**：被击中后 0.6 s；伤害被护盾完全吸收时 0.3 s；复活后 2 s；读档后 1.5 s（同时 `noBlinkT 2.5`）。
- 裂隙（rift）里伤害乘以 `Ui().dmg`。护盾 `pr.shield`（同伴技能提供，`shieldT` 计时）先吸收伤害。
- 玩家受到的击退：`v += dir·n·1.1`；`n > 6` 时还会被击飞（`vy=4`）。之后脱战计时 `inCombatT = 8 s`（玩家打中别人时为 6 s）。
- 敌人打玩家的伤害：`Ys(enemy.atk, player.def, spec.power, 1, rnd, 0)`，暴击另由 `critChance()` 判定（L34538）。敌人等级缩放：普通 blight `×(1 + .085·(lv−1))`，精英和 Boss 固定数值（L34275）。

3.6 捕获（F）

- 选目标 `captureTarget`（L47011）：锁定的 creature 在 16 m 内就用它；否则取 14 m 内、朝向点积 ≥ 0.35（贴身 2.5 m 内不看朝向）的目标，按 `距离·(2−点积)` 打分取最小。
- 装置等级要求 `sp(rarity)`（L23269）：普通到稀有需要 1 级，史诗需要 2 级，神话需要 3 级。
- 捕获概率 `ux`（L27848）：

```
t = 1 + (1 - hpRatio) * 1.8;           // 血越少越容易抓
n = species.capture * device.deviceMult * t * .62;
+ .18（对口诱饵） + .1（击晕） + .15（睡眠） + charmBonus;   clamp [.02, .95]
```

初始伙伴 `starter` 的概率至少为 0.94。

- 结果 `dx`（L27886）：成功固定晃 3 次；失败随机晃 0–2 次。
- 表现（`runCapture`，L47178）：先是 `throw` 阶段 0.55 s，从手部 lerp 到目标，叠加 `sin(n·π)·1.8` 的抛物线高度；然后 `wobble` 每次 0.75 s；最后 `result` 0.5 s。成功：慢动作 0.8 s、FOV 冲击、顿帧 0.12、XP `10 + order·8`。失败：温顺的生物会逃跑，其余的被激怒 12 s 并进入 stunned。

3.7 治疗

- X 快速治疗 `quickHeal`（L46380）：在所有 `heal>0` 且非复活类的物品里，按 heal 从小到大排序，选第一个 heal ≥ 缺失血量·0.8 的；没有就用最大的。冷却 0.8 s。体力低于一半时也允许使用。
- 治疗道具：berry 8、meat 5、sunroot 14、glowberry 10（+40 体力）、starmelon 25、roast 45、stew 80（+60 体力）、bandage 35、salve 75。完整数值见 tables_readable.txt。
- 营火休息 `restAtCamp`（L47632）：站在营火 6 m 内、附近没有在追击的敌人时可用。时间跳到下一个 6:30 或 20:00，并按 5 s 为步长模拟这段时间的建筑产出。HP 和体力回满，队伍至少回到 70% HP、复活倒地成员、能量 +30。
- 治疗站 `ax`（L27590）：给队伍回满 HP。没有工人时每个成员消耗 1 个 herb。冷却：有工人 20 s，没有工人 45 s。

3.8 死亡与复活

- `die()`（L46653）：清除锁定、捕获序列和建造放置；统计 `deaths++`；Boss 战中会重置竞技场。
- `updatePlayer` 里 `deathT > 3.2 s` 时调用 `respawn()`（L46230）。
- `respawn()`（L46733）：
  - 传送到要塞 `RA=(0,58)` 的 z+6 处，yaw 设为 π。
  - HP = 60% maxHp，体力满。
  - 队伍里倒地的成员复活到 35% HP，其余成员至少 50%。
  - 所有在追击的野怪回到 `return` 状态。

    - 立即存档，并尝试播放 Y8 插页广告 `yn.midgame`（冷却 180 s）。
    - 代码里没有任何物品或金币惩罚【推断：死亡唯一的代价是跑图】。

3.9 击杀奖励

- 野生 creature：XP 和金币基数都是 `4 + 2·lv + 4·rarityOrder`；金币再乘 0.32（`pc`，L41966）。另有 60% 概率掉 essence、50% 概率掉 meat。
- 敌人：`px()`（L27905）按掉落表 roll，XP 用表里的 `xp` 字段。Boss 二刷时 XP ×0.5、掉落 ×(1 + 0.5·rematches)。队伍成员各得 50% XP。金币 `pc(xp, kind)`：Boss ×1.2、精英 ×0.55、其它 ×0.32。精英被击杀后 900 s（游戏时钟）重生。
- 敌人数值表（hp / atk / def / spd / xp / 元素 / 掉落）可在 `rp` 里查到，例如 `blightcrawler 45/9/2/3.6/14`、`elite_cinderhorn 520/24/9/4.0/260`、`boss_aethon 2600/38/14/3/1500`。完整导出方法见 tables 生成脚本（L23272–23905 加上补丁 `wkAddEn` L44076 新增的 blightfish / blighttribal / blightmask）。

---

4. 背包、物品、制作、采集、建造

4.1 背包（**`class vM`**，L24782）

- 格子数 `su = 24 + 12 × 箱子数`（`gM=24, dp=12`，L24780 和 L26937）。箱子最多 4 个，所以最多 72 格。
- 堆叠上限由物品的 `stack` 决定（工具和装备为 1，大部分资源为 99）。`add` 先填已有堆叠再开新格；`remove` 从最后一格往前扣。
- 背包满时溢出的物品进入 `state.pending`（待领取），之后自动 `flushPending`（L25461）。
- 装备槽：`weapon / armor / charm`。工具和弹弓不用装备，只要在背包里就生效（`au()` 取背包里最高等级的工具，L27133）。

4.2 物品表（56 个，**`NA`**，L24184–24757）

类别	物品 id（关键属性）
资源 resource	wood, stone, fiber, ore, crystal, essence, herb, resin, ember_core, prism_shard, aether_dust, charcoal
材料 material	plank, brick, rope, ingot, lens, aether_core
食物 food	berry(h8/s15), meat(h5), sunroot(h14/s20), glowberry(h10/s40), starmelon(h25/s30/energy25), roast(h45/s20), stew(h80/s60), treat(同伴治疗 60 / 能量 45)
种子 seed	sunroot_seed, glowberry_seed, starmelon_seed
治疗 healing	bandage(35), salve(75), tonic(复活 + 同伴治疗 80)
捕获装置 device	snare(T1 ×1.0), cage(T2 ×1.5), orb(T3 ×2.2), lantern(T4 ×3.2)
诱饵 bait	sweet / savory / mineral / arcane
工具 tool	stone_hatchet(axe T1), stone_pickaxe(pick T1), iron_hatchet(axe T2), iron_pickaxe(pick T2)
武器 weapon	wood_club +4, iron_blade +10, ash_spear +15, crystal_glaive +22, sling（远程，atk 0）
弹药	pebble
护甲 armor	leaf_vest +2, ash_mail +6, crystal_plate +11

类别	物品 id（关键属性）
护符 charm	vital_charm +30 HP, swift_charm +12% 速度, bond_charm +12% 捕获

4.3 制作（**Wo**，L27637；校验 **Tp**，L27801；执行 **lx**，L27832）

- 工作台判定 `stationsNear()`（L47472）：`hand` 永远可用；13 m 内有 campfire / workshop / advanced 建筑才算可用。
- 等级门槛 `unlock.level`。`stack:1` 的物品已经拥有时不能重复制作。
- 折扣：工坊或高级工坊有工人时，`gd()` 给出折扣 `min(0.3, 工人效率·0.08)`。只对数量 ≥ 3 的原料生效：`ceil(n·(1−折扣))`（L27789）。
- 制作是原子操作：先做快照，扣料，产物放不下就回滚。每次制作玩家获得 3 XP（L47495）。

完整 38 条配方（输入 → 输出 @工作台 [等级]）：

id	输入	输出	工作台	等级
r_hatchet	2 wood, 2 stone, 1 fiber	1 stone_hatchet	hand	
r_pickaxe	2 wood, 3 stone, 1 fiber	1 stone_pickaxe	hand	
r_club	4 wood, 1 fiber	1 wood_club	hand	
r_sling	1 wood, 3 fiber	1 sling	hand	
r_pebble	1 stone	6 pebble	hand	
r_snare	3 fiber, 1 wood	3 snare	hand	
r_bandage	2 fiber, 1 herb	2 bandage	hand	
r_vest	8 fiber, 1 herb	1 leaf_vest	hand	
r_sweet	2 berry, 1 fiber	2 sweet_bait	hand	
r_plank	3 wood	2 plank	hand	
r_rope	3 fiber	1 rope	hand	
r_brick	3 stone	2 brick	hand	
r_treat	2 berry, 1 fiber	2 treat	hand	
r_roast	1 meat	1 roast	campfire	
r_stew	1 meat, 1 herb, 2 berry	1 stew	campfire	
r_savory	1 meat, 1 herb	2 savory_bait	campfire	
r_charcoal	3 wood	2 charcoal	campfire	
r_treat2	1 sunroot, 1 berry	3 treat	campfire	
r_cage	2 plank, 2 rope, 1 essence	2 cage	workshop	
r_ingot	2 ore, 1 charcoal	1 ingot	workshop	
r_iron_hatchet	2 ingot, 2 plank, 1 rope	1 iron_hatchet	workshop	
r_iron_pick	2 ingot, 2 plank, 1 rope	1 iron_pickaxe	workshop	
r_iron_blade	2 ingot, 1 plank, 1 rope	1 iron_blade	workshop	
r_salve	3 herb, 1 fiber	2 salve	workshop	
r_mineral	2 stone, 1 crystal	2 mineral_bait	workshop	
r_tonic	3 herb, 2 essence	1 tonic	workshop	
r_vital	4 essence, 3 herb, 1 rope	1 vital_charm	workshop	
r_swift	4 essence, 2 resin, 1 rope	1 swift_charm	workshop	4
r_ashmail	4 plank, 4 resin, 2 ingot	1 ash_mail	workshop	4
r_ashspear	3 plank, 3 resin, 3 ingot, 1 ember_core	1 ash_spear	workshop	5
r_orb	1 ingot, 3 essence, 2 crystal	2 orb	advanced	5
r_lens	3 crystal, 1 ingot	1 lens	advanced	5
r_lantern	1 lens, 1 ingot, 3 essence, 1 aether_dust	2 lantern	advanced	6
r_arcane_bait	2 essence, 1 crystal	2 arcane_bait	advanced	
r_core	3 prism_shard, 2 aether_dust, 1 lens	1 aether_core	advanced	6

id	输入	输出	工作台	等级
r_glaive	2 lens, 3 ingot, 3 prism_shard, 2 plank	1 crystal_glaive	advanced	6
r_plate	2 lens, 4 ingot, 4 prism_shard, 2 rope	1 crystal_plate	advanced	6
r_bond	6 essence, 2 prism_shard, 1 lens	1 bond_charm	advanced	6

4.4 采集与工具等级（**mr**，L25892；**Ho / NM / IM**，L27148–27198；**gatherNode**，L47369）

节点	产出	工具 / 等级	基础次数	重生（游戏秒）
tree	5 wood	axe T1	4	300
ash_tree	6 wood + 1 resin	axe T1	5	360
branch	2 wood	徒手	1	200
loose_stone	2 stone	徒手	1	200
boulder	6 stone	pick T1	4	320
ore_vein	5 ore	pick T1	5	240
berry_bush	3 berry	徒手	1	240
fiber_tuft	2 fiber	徒手	1	180
herb_patch	2 herb	徒手	1	280
sprout	1 fiber + 随机种子（sunroot×2 / glowberry / starmelon 权重）	徒手	1	300
ember_vent	1 ember_core + 2 charcoal	pick T2	5	480
crystal_cluster	3 crystal	pick T2	5	420
prism_spire	2 prism_shard	pick T2	7	600
aether_bloom	1 aether_dust	徒手	1	520

```
// L27159  需要的敲击次数
const IM = (s, A) => s.tool === "none" ? 1 : Math.max(2, Math.ceil(s.hits / (1 + .45 * (A - 1))));
// L27166-27170  产量：工具每超过要求一级 +50%，再乘节点自身的随机系数 (.8 + e.s*.2)
c = 1 + .5 * (a - r.tier); n = Math.max(1, Math.round(u.n * c * (.8 + e.s * .2)))
```

- 每次敲击冷却 0.38 s；“剩余次数”记录 5 s 后过期（`a.t = 5`）。
- 节点采空后写入 `world.depleted[id] = clock + respawn`。
- 完成一次采集：徒手 +1 XP，用工具 +3 XP。

4.5 建筑与放置（**nu**，L25475；**startPlacement / updatePlacement**，L47675–47768；**bp / PM**，L26941–27031）

id	造价	半径	上限	解锁等级	工人位	最高等级 / 升级费用（×当前等级）
campfire	4 wood, 4 stone	1.6	2	1	1	3 / plank 4, brick 3
chest	6 wood, 2 fiber	1.2	4	1	0	1
workshop	8 wood, 6 stone, 1 rope	2.2	1	2	1	1
shelter	4 plank, 6 fiber, 4 stone	2.0	4	2	0	1
farm	4 wood, 4 fiber	2.0	3	2	2	2 / plank 4, brick 2
station	4 plank, 4 brick, 2 rope	2.0	2	3	2	3 / ingot 3, brick 4
training	5 plank, 3 rope, 6 stone	2.4	1	3	2	1
healing	4 plank, 6 herb, 3 brick	1.8	1	3	1	1
advanced	8 plank, 4 ingot, 6 brick, 3 rope	2.4	1	5	1	1
beacon	8 brick, 4 ingot, 2 lens, 6 crystal	1.8	1	6	0	1（不返还材料）

放置算法（不是格子拼接式建造，没有吸附到其他建筑的功能，而是“圆形占地 + 0.5 m 网格量化”）：

```
// L47738-47740  幽灵模型位置：玩家前方（沿相机 yaw）radius+3.2 处，量化到 0.5 m
r = n.radius + 3.2,
a = Math.round((this.pr.x + Math.sin(this.cam.yaw) * r) * 2) / 2,
c = Math.round((this.pr.z + Math.cos(this.cam.yaw) * r) * 2) / 2,
// L47752  旋转：R 或 Q 每次 +π/4
(i.pressed("rotate") || i.pressed("device")) && (e.rot += Math.PI / 4, ...)
```

合法性检查 `bp()`（L26941），按顺序：

1. 等级达到要求，数量未到上限。
2. 位置在 `|x|, |z| ≤ XA−4 = 136` 以内。
3. 离要塞 `RA` 不超过 `yM = 34` m。
4. 所在生物群系必须是 `meadow`。
5. 中心点和四个方向半径处的坡度 `cs()` 都不超过 0.45。
6. `world.occ.blocked || grid.blocked(..., ["ground"])` 都不阻挡。
7. 与已有建筑不重叠，间距要求 `≥ r1 + r2 + 0.4`。
8. 玩家没站在占地范围里，要求距离 `≥ r + 0.7`。

确认放置用 Enter、鼠标左键或 E；按住 Shift 确认可以连续放置（L47765）；X、B 或 Esc 取消。幽灵模型是半透明（opacity 0.55）的建筑模型，外加一圈地面环，合法时绿色 `0x9DFFB0`，不合法时红色。落地时坐标再取整到 0.5 m（`PM`，L27006）。建成后玩家获得 6 XP。

拆除 `DM`：返还 `floor(造价·0.75)`，外加建筑内未取走的产出和农田里的种子；已点亮的信标不能拆。

建筑模型 `hc(type)`（L39669）全部是程序化图元。

---

5. 农业、任务、等级、经济、商人

5.1 农业（**`va / hp`**，L24758；**`sx`** 种植 L27525，**`rx`** 收获 L27557，**`gl`** 模拟 L27433）

作物	成熟时间（秒）	收获	返还种子
sunroot	60	2 sunroot	1 sunroot_seed
glowberry	100	3 glowberry	1 glowberry_seed
starmelon	180	2 starmelon	1 starmelon_seed

- 地块数 `ru(lv) = 4 + 2·(lv−1)`：1 级 4 块，2 级 6 块（L26938）。
- 生长速度 `1 + 0.5·工人效率`（`+0.1`，如果工人是 `dewdrake`）（L27455）。
- 有工人时自动收获：产物放进建筑的 `out`（每种物品最多 40 个，`cu=40`）。如果 `out` 里有对应种子，会自动补种。
- 没有工人时需要手动按 E 收获，每次 +4 XP。

其它工作建筑（`gl`）：

- campfire：有工人时每 30 秒“进度单位”产 1 charcoal（emberkit 工人效率 ×1.2，每升一级 +25%）。
- station：stone 20、ore 45；2 级以上解锁 crystal 120。雷属性工人 ×1.4。

- training：工人每秒获得 0.6 XP。

离线产出 `ox`（L27624）：读档时如果距上次存档超过 20 s，按 `min(离线秒数, 1800) × 0.5` 模拟，步长 5 s（L45690–45697 会提示“限 30 分钟、半速”）。

5.2 任务 / 日志（**`ui`**、**`fp`**，L24871–25267；管理器 **`fd`**，L25394）

主线是严格的线性顺序（完成一个后自动激活下一个未完成的，L25447）。进度按“激活那一刻的统计快照 `quests.base[id]`”做差计算（`tu`，L25365）。

#	id	英文标题	目标	奖励
1	q01_steps	First Steps in Aetherwild	移动 15 m	20 XP, 3 fiber
2	q04_sighting	First Sighting	发现 1 种生物	30 XP, 3 snare
3	q05_capture	Friend of the Wild	捕获 1 只	80 XP, 3 snare, 2 sweet_bait
4	q06_party	Teamwork	队伍里有 1 只	40 XP, 3 treat
5	q02_gather	Scavenger	采集 5 wood、4 stone、3 fiber	40 XP, 4 berry
6	q03_tools	Hands to Tools	制作 stone_hatchet 和 wood_club	40 XP, 6 wood, 4 stone
7	q07_defend	Hold the Line	击杀 3 个 blight	80 XP, 3 bandage, 12 pebble
8	q08_fire	A Warm Hearth	建 campfire	60 XP, 2 meat, 2 charcoal
9	q09_storage	Room to Grow	建 chest	60 XP, 4 plank
10	q10_worker	Helping Paws	指派 1 个工人	80 XP, 4 treat, 2 rope
11	q11_farm	Green Thumb	建 farm，种 2 次，收 2 次	100 XP, 4 sunroot_seed, 2 glowberry_seed
12	q12_forge	Forged for the Wild	建 workshop，造 iron_blade 和 iron_pickaxe	120 XP, 2 ingot, 4 herb，解锁 woods
13	q13_woods	Into the Ashen Woods	进入 woods	100 XP, 3 savory_bait, 2 cage
14	q14_rare	Rare Find	捕获 1 只稀有及以上	150 XP, 3 cage, 4 essence
15	q15_elite	Slay the Alpha	击杀 elite_cinderhorn	300 XP, 2 orb, 3 salve，解锁 wastes
16	q16_wastes	Crystal Calling	进入 wastes	150 XP, 3 crystal
17	q17_prism	Prism Harvest	持有 8 crystal、6 prism_shard、2 aether_dust	200 XP, 4 ingot, 1 lens
18	q18_beacon	Light the Beacon	建造并点亮信标（献祭 `Go = 1 aether_core, 3 prism_shard, 4 crystal`）	300 XP, 2 aether_dust
19	q19_prepare	Prepare the Expedition	制作 crystal_plate，等级达到 7	200 XP, 4 salve, 2 tonic, 2 stew
20	q20_final	The Final Expedition	击败 boss_aethon	1000 XP, 10 essence

支线（`optional:true`，可同时进行，按全局统计判断）：

id	英文标题	目标	奖励
c_explorer	Explorer	发现 6 个地点	120 XP, 5 essence
c_treasure	Treasure Hunter	开 3 个宝箱	120 XP, 2 cage, 2 salve
c_collector	Collector	图鉴 10 种，收藏 8 只	200 XP, 1 orb, 5 treat
c_epic	Epic Bond	捕获史诗	250 XP, 2 orb, 6 essence
c_mythic	Mythic Bond	捕获神话	500 XP, 2 lantern, 10 essence
c_farmer	Master Farmer	收获 15 次	150 XP, 4 starmelon_seed, 5 treat
c_hunter	Blight Hunter	击杀 25 个 blight	200 XP, 2 tonic
c_builder	Master Builder	4 个 shelter	200 XP, 10 plank, 10 brick
c_elites	Elite Slayer	击杀 3 个指定精英	400 XP, 3 aether_dust, 10 essence
c_veteran	Veteran	等级 12	300 XP, 12 essence

击败 Boss 后 `unlockFrontier()` 会加入 `rift`（L47931）。rift 有每日挑战和计分（`Mn/Ui/tB`）。

5.3 玩家等级（**`Js / tr / hd`**，L25270–25300）

```
hd = 30,                                    // 等级上限
tr = s => Math.round(50 * Math.pow(s, 1.5));  // 从 lv 升到 lv+1 需要的 XP
```

升级时 HP 和体力回满；满级后 xp 被限制在 `tr(30)-1`。部分数值：

升级	所需 XP	累计
1→2	50	50
2→3	141	191
4→5	400	851
5→6	559	1410
6→7	735	2145
9→10	1350	5552
11→12	1824	8957
19→20	4141	33567
29→30	7808	94514

XP 来源：击杀、捕获（`10 + 8·order`）、任务、制作 3、采集 1 或 3、建造 6、手动收获 4。生物自己也有等级：上限 30，`_o = round(30·lv^1.4)`；训练场工人每秒 0.6 XP。

5.4 金币经济

- 金币存放在 `state.stats.gold`，不是一个独立字段（L46670）。所以它会随 stats 一起存档，读档清洗时要求非负、有限。
- 收入：
  - 击杀（`pc`）。
  - 每日登录（7 天循环 `op`，L23942）：60 → 3 snare → 100 + 2 essence → 3 aether_dust + 3 herb → 160 + 2 cage → 80 + 3 crystal + 2 essence → 300 + 2 orb + 5 essence。
  - 每日 3 个挑战（`hl`，用日期种子 `$l` 生成，L24030）：每个 40–70 金币加小物品；全部完成额外奖励 `Lo` = 150 金币、3 essence、2 snare。
  - 出售资源。
  - 看 Y8 激励广告 +150（L46723）。
- 出售价 `Yy`（L41955）：wood/stone/fiber/berry 2，rope/charcoal 3，herb/plank/brick 4，ore/resin/meat 5，essence 6，crystal 9，ember_core/ingot 14，aether_dust 16，prism_shard 20，lens 22，aether_core 45；其它 food 默认 3，再其它默认 0（不能卖）。已装备的物品不能卖。

5.5 巫师商人（Wizard Merchant）商店

- 模型：`Vd()`（L41990）是程序化搭的摊位（box 加三个球表示货物），摊主是 `l0("npc_merchant")`，即 `blob/Wizard` GLB（L22091）。

- 位置固定，不会游走：`gt = {x: RA.x+3, z: RA.z+13, rot: π}`（L42465），就在出生营地里，是 `CRIT` 预加载模型之一。“Traveling” 只是外观设定【推断】。玩家 80 m 内才更新它的动画。
- 交互标签“Parler au marchand”（L47318），打开 `shopHTML()`（L49181），共 5 个标签页：

帽子 / 披风 / 外观（**`ra`**，L41828）

id	英文名	槽位	价格（金币）
hat_cap	Scout's Cap	hat	90
hat_chef	Chef's Hat	hat	140
hat_bunny	Bunny Ears	hat	220
hat_viking	Horned Helmet	hat	320
hat_wizard	Wizard Hat	hat	450
hat_top	Top Hat	hat	600
hat_crown	Golden Crown	hat	1200
cape_red / cape_blue / cape_green	Scarlet / Night / Forest Cape	cape	各 260
cape_gold	Royal Mantle	cape	900
tint_forest / dusk / ember / ocean	Forest / Dusk / Ember / Ocean Outfit	tint	各 380

拥有状态记为 `stats["own."+id] = 1`；装备状态记为 `stats["eq."+slot] = ra` 中的下标 `+ 1`（L46679–46698）。

生物（**`Kp`**，L41920）：fluffern Lv3 150、puddlepup Lv3 160、twigwing Lv4 180、emberkit Lv5 320、mossback Lv8 900、prismoth Lv10 1500、sporelord Lv10 2400、hexabuzz Lv10 2600、maelglub Lv12 2800、golemurk Lv12 3000、aurelpaca Lv12 3400。收藏满（`ls = min(66, 30 + 6·shelter)`）时不能买。

出售：见 §5.4。

---

6. 存档、自动存档、游戏循环、状态机

6.1 存储后端（L28209–28358）

- `Mx()`：优先使用 IndexedDB（数据库 `"wildkeep"`，版本 1，objectStore `"kv"`），失败时回退到 `localStorage`（`bx`）。
- IndexedDB 模式下，`wk_main` 会同时镜像写到 localStorage（L28243）。
- 存档键：`wk_main`（主档）、`wk_bak`（上一份好档）、`wk_tmp`（写入中的临时档）、`wk_corrupt`（开新游戏时把读不出来的旧主档备份到这里）。
- 存档格式 `yx()`（L28269）：

```
JSON.stringify({ v: 2, sum: FNV1a32(dataStr), data: JSON.stringify(state) })
// ap = FNV-1a 32-bit (L23915): h=2166136261; h^=c; h=Math.imul(h,16777619)
```

data 就是 dl() 生成的 state 树：v, created, saved, clock, hour, day, player{x,z,yaw,hp,stam,lv,xp,equip}, inv[], pending[], creatures[], party[≤3], dex{}, buildings[], quests{active,done,base,seq}, stats{...gold, own.*, eq.*}, world{unlocked, visited, locations, caches, depleted, eliteDown, bossDefeated, beacon, victory, rematches, ev, campDown, tutorial, uid, seed, daily{...rift}}, settings{}。

- 原子写入 Bx.save()（L28324）：写 tmp → 读回校验 → 如果当前 main 有效就复制到 bak → 写 main → 读回校验 → 删除 tmp。
- 读取 Bx.load()：按 main（状态 ok）→ bak（recovered）→ tmp（recovered）的顺序尝试。三个都存在但全部损坏时，locked=true，之后停止自动存档，直到玩家开新游戏（L48148 会弹提示）。
- 读档清洗 vx()（L28008）：每个字段都经过 KA 钳制，例如 lv 1–30、坐标 ±140、生物等级 1–30、pot 0.8–1.5、建筑数量不超过上限、工人必须符合物种工作类型、农田作物必须在白名单里。gx() 负责 v1→v2 迁移（inv 从对象改为数组）。

  【推断：可能是 bug】settings 只保留白名单里的键（master/music/sfx/quality/sens/invertY/reduceMotion/autoBait/tapMove/lang），但新游戏时把 outfit/hair 写进了 state.settings（L49474）。所以刷新页面重新读档后，主角外观会回到 outfit 0。装饰品存在 stats 里，不受影响。

6.2 自动存档节奏（**markDirty**，L45848；补丁 **QUALITY.save.step**，L44572）

```
markDirty(A=false){ ... this.saveDirty = A ? .6 : Math.max(this.saveDirty, 4) }   // 普通改动 4 s 去抖，关键改动 0.6 s
// QUALITY.save.step：去抖到 0 时存档；如果一直被 markDirty 续命，最多等 12 s 强制存档
// 另有安全存档：每 30 s 检查一次，有改动或移动超过 6 m 就存，否则最多 120 s 存一次
```

另外这些时机会立即调用 saveNow()：页面隐藏（visibilitychange hidden）、pagehide、复活、退出到菜单、暂停菜单里的“立即保存”。saveNow 有重入保护：保存进行中再次请求会设置 saveAgain 标记，当前保存完成后再存一次（L48146）。

6.3 Y8 SDK（**yn**，L47–145）

- 初始化：等待 y8sdk.ready 事件（超时 3 s），然后 y8.sdk().init({appId:"6ac9de676da5b1f7cb9fa002", autoLogin:true}, {gameId:"287019", preloadAdBreaks:"auto"})。
- 只用于广告：midgame() 在复活时播插页广告（名称 "respawn"，间隔至少 180 s）；rewarded() 是金币激励广告（名称 "gold-boost"，+150）。广告期间 window.__y8Ad=true，会暂停游戏（L45886）并静音。
- 没有云存档。submitScore() 返回 false，leaderboard 也是 false，属于桩函数【推断：这个构建剥离了排行榜和云存档】。存档完全在本地（IndexedDB 和 localStorage）。

6.4 localStorage 键一览

键	用途	位置
`wk_main`	主存档的镜像（IndexedDB 不可用时就是主存储，`wk_bak/wk_tmp/wk_corrupt` 也同样落在这里）	L28237
`wk_settings`	设置 JSON。启动时只合并默认值里存在、且类型一致的键（L49740）	L49243
`wk_lang`	语言（`fr` / `en` / `es`，对应的字典 chunk 动态加载）	L212、L221
`wk_q3`, `wk_q4`	一次性的画质迁移标记：第一次运行时把画质从 low 改成 auto	L49750
`wk_forceLow`	WebGL 上下文恢复后强制使用低画质	L45636、L49747
`wk_trk`	移动端任务追踪器的折叠状态	L43885
`wk_camzoom`	触屏相机缩放	L44930
`wk_qhist`	画质 / 性能历史（QUALITY 模块）	L42835

6.5 游戏循环（**`iB.loop / frame / updatePlaying`**，L45855–45920）

```
// L45855-45858  rAF 驱动，可变步长（没有 fixed timestep），dt 上限 50 ms
loop = A => { requestAnimationFrame(this.loop); if (!this.running || QUALITY.skip(A, this)) return;
  const e = (A - this.last) / 1e3; let t = Math.min(.05, e); ... this.frame(t) }
// L45886-45888  时间缩放：慢动作 0.3；顿帧 0.05
this.slowmoT > 0 ? (..., this.timeScale = .3) : this.timeScale = 1,
this.hitStopT > 0 && (..., this.timeScale = Math.min(this.timeScale, .05));
const n = A * this.timeScale;
this.mode === "menu" ? this.menuCamera(e) : t ? this.updatePlaying(n, e) : this.updateRenderOnly(n, e);
```

- `QUALITY.skip`（L43398）：测量屏幕刷新率（取 60 个 rAF 间隔的 15% 分位数，吸附到 60/72/…/240 Hz），超过目标帧率时跳帧限帧。移动端打开面板或闲置 45 s 后降到 30 fps（`QUALITY.vis.low`，L44694）。
- 远处 AI 按距离 LOD 降频，“思考”间隔为每帧 / 30 Hz / 10 Hz / 4 Hz（`QUALITY.logic`，L44741）。
- **`updatePlaying`** 的更新顺序（L45917）：
  1. 游戏时钟：`clock += dt`，`hour += dt·24/Wd`（`Wd = 720`，一个游戏日等于 12 分钟现实时间），`QM` 清理已过期的 depleted 记录，每秒同步一次采集节点状态。
  2. `updatePlayer`
  3. `updateCamera`
  4. `updateActors`
  5. `updateProjectiles`
  6. `updateSpawner`（每 0.5 s 执行一次）
  7. `updateCapture`
  8. `updateInteract`
  9. `updatePlacement`
  10. `gl()`（建筑和工人模拟）
  11. `updateBuildingViews`
  12. `updateLock`
  13. `updateWorldEvents`

14. updateLocations
15. disc.update
16. ensureStarter
17. updateCcompanions
18. updateDaily
19. updateArena
20. updateTemp
21. updateAudioContext
22. updateMobileContext
23. updateHints
24. QUALITY.save.step
25. 把 pr 的位置和朝向写回 state.player

- frame() 之后（无论在哪个模式）：大气、水面、屏障、shader uniform、萤火虫粒子、节点渲染、商人、风景装饰、特效、奖励飞字、animateWorld、震屏、阴影 tick，然后后处理渲染（EffectComposer：RenderPass → UnrealBloom → 调色（含受伤红屏 hurt uniform）→ OutputPass，L35842），最后 ui.update。

- 输入是边沿触发的：pressed(x) 读一次就清掉。updatePlacement 结尾会 clearEdges()。

6.6 主状态机

资源加载阶段（QUALITY.assets.mark，L44257）：boot → core（预加载 CRIT 里的 5 个 GLB：PinkBlob、Pigeon、Alien、Dino、Wizard）→ menu → zone（开始新游戏或继续游戏）→ secondary（后台按“出生群系 → 敌人 → 后续群系 → 精英”的顺序逐个加载其余 GLB，并会避开战斗时段）。可以在控制台用 window.__wkBoot() 查看当前阶段。

启动流程 dB()（L49724）：

1. 检测 WebGL。
2. 加载语言 om()，挂载 UI um()，初始化 Y8 SDK。
3. new iB(canvas)，new hB(game, ui)。
4. 合并 wk_settings，处理 wk_forceLow 和 wk_q4。
5. iB.init()：创建 renderer 和场景、开始 rAF 循环（此时 mode = "menu"；menuHold 期间画布透明度为 0）。
6. openSaves()，得到存档状态 ok / recovered / corrupt / none。
7. ui.initMenu(status, hasSave)。
8. cM() 预热模型，然后 releaseMenu()（超时 20 s 也会释放），标记 menu，开始后台加载。

运行时模式：

mode	modal	行为
"menu"	—	menuCamera 绕要塞旋转（半径 24，角速度 0.06 rad/s），主角隐藏，input.blocked = true
"playing"	false，且没有在播广告	updatePlaying（完整模拟）
"playing"	true（任意面板打开）或 __y8Ad	updateRenderOnly：只更新矩阵和姿态，游戏时间暂停；setModal 时释放 Pointer Lock 并清空按键（L45938）

- 进入游戏：doNew()（L49471）调用 startNew → afterStateLoaded(true)；继续游戏调用 loadState → ox 离线产出 → afterStateLoaded(false)。两条路径最后都会设置 mode = "playing"、setModal(false)，并播放探索音乐。
- 退出：exitToMenu() 先存档，再 cleanupWorld()，然后 mode = "menu"。
- 玩家子状态：pr.alive 为 false 时进入死亡动画，deathT > 3.2 后复活。placing 状态下禁用攻击、弹弓、跳跃和治疗，X 改为取消放置。capSeq 期间每帧调用 runCapture。bossLocked 时锁在竞技场里、不能快速旅行。victoryT 期间对玩家伤害无效。
- 音乐状态 currentMusic()（L45940）：menu / victory / boss / combat / night / explore。

附：结论可信度

- 所有数值表（物品、配方、建筑、任务、节点、敌人）都是用 node 直接对源码切片求值导出的。可以用 report/data/dump.js 和 print.js 复现，结果保存在 report/data/。
- 标【推断】的点：模型来源归属（Kenney）、“traveling”商人其实固定不动、投掷动作复用上一次的 rigAct、夜晚回血门槛恒为 true、死亡无惩罚、Y8 构建剥离了云存档和排行榜、outfit 读档后被清掉。其中 outfit 读档丢失可以实测验证：选一个非 0 的 outfit，刷新后点继续游戏。