

README h-lua-sdk-doc

快速开始

h-lua是一套完善的魔兽地图开发集成，更贴近程序开发者的使用套路。

摒弃手动物编，多余的excel物编，直接敲码绑定逻辑。

避免已知恶性问题，完善的马仔帮你调用API。从建图到上线，一波跑通，不浪费时间。

安装一个fork 或 git

[Fork - a fast and friendly git client for Mac and Windows \(git-fork.com\)](#)

[Git \(git-scm.com\)](#)

代码编辑器

编辑器推荐使用Jetbrain Ideac(社区版免费) 并按照插件【EmmyLua】。

相信得到编辑器 Buff加成 的你，会如虎添翼~

[Ideac \(Jetbrain\)](#)



克隆（下载）项目

```
git clone https://github.com/hunzsig-warcraft3/h-lua-sdk
```

[可使用码云加速：\(Gitee -> h-lua-sdk\)](#)

打开终端并进入工作区

```
cd ./h-lua-sdk
```

创建新项目

```
> sdk.exe new demo
```

修改地形 (非必要)

```
> sdk.exe we demo
```

测试

```
> sdk.exe test demo
```

拓展例子地图

拓展地图会编写一些业务，来引导你进行学习

- [helloWorld](#)
- [压力测试](#)
- [秘地探奇半成品](#)

完全沒试过lua作图怎么快速学习？

- 了解lua基本语法：[Runoob Lua教学](#)
- 了解YD引擎：其他学习 -> YDLua引擎
- 有魄力的人可以先略一遍 h-lua 的源码，即使你不用框架内实现，也是有所帮助的

目录结构

```
└── depend - 依赖的开发套件
    ├── h-lua - h-lua(v:latest, 随sdk更新的最新版)
    ├── w3x2lni - w3x2lni工具(v:2.7.2)
    └── YDWE - 马仔工具
└── projects - 用来放置你的地图项目目录, 如 h-lua-sdk-helloworld
    └── h-lua-sdk-helloworld
└── sdk.exe - sdk命令工具
```

项目结构

```
└── project_demo - 项目目录
└── hslk - 用来编写 hslk lua 物编配置, 不要把流程代码写里面
└── map - 地图文件
    ├── implant - 用来强制更新替换【DZUI布局、命令位置、平衡性常数、原生界面】参数
    └── resource - F12导入
        ├── hLua - h-lua需要的资源文件, 请不要乱删除
        └── interface - 冷却时间UI, 不需要可删除, 然后需要修改
implant/war3mapSkin.txt
    |   ├── ReplaceableTextures
    |   └── TerrainArt - 地形贴图, 不需要可直接删除
```

```
|   |   |   └── Cliff - 悬崖贴图, 不需要可直接删除  
|   |   └── CommandButtonsDisabled - 暗图标目录  
|   |   └── selection - 选择圈, 不需要可直接删除  
|   └── UI - 命令等系统图形的修改 (不包括dzui) 不需要可删除, 然后需要修改  
implant/war3mapSkin.txt  
|   ├── war3mapImported - 通用目录  
|   └── war3mapMap.blp - 小地图文件, 一般不会手动处理, 交给 -yd  
└── slk - ini式的物编  
    └── w3x - 地图ini  
└── scripts - lua脚本 (*此乃建议, 实际上你的lua只要在项目目录内, 一般都能按路径访问)  
    └── main.lua - 项目代码入口
```

命令行

新建项目

```
> sdk.exe new demo //新建一个地图项目, 名为demo
```

使用WE编辑地形

编辑完后在使用test时会自动将改动记录回你的项目中

```
> sdk.exe we demo //以WE打开项目demo, 主要用于编辑地形
```

模型批量查看

模型默认scale1.00, 方便对比

```
> sdk.exe model demo //查看项目demo的模型, 默认第0页  
> sdk.exe model demo 2 //查看项目demo的模型, 第2页  
> sdk.exe model demo ttg //查看项目demo的模型, 只要路径带有ttg的  
> sdk.exe model demo abc 1 //查看项目demo的模型, 第1页且只要路径带有abc的
```

测试

在test调试中, 你可以直接F10重启魔兽获得脚本的更新, 而无需再次使用test命令

重启脚本不包括hslk目录以及resource资源的变更, 只对框架脚本或项目脚本有效

```
> sdk.exe test demo //测试你的demo项目并测试  
> sdk.exe test demo ? //可以在后面加个问号, 从而生成地图而不进行测试, 一般用于检查物编的生成结果
```

打包

在build调试中，F10重启魔兽不能获得脚本的更新，必须再次使用build命令

build命令会引用打包流程的lua脚本，DEBUGGING会自动置为false

```
> sdk.exe build demo //打包你的demo项目并测试  
> sdk.exe build demo ? //可以在后面加个问号，从而生成地图而不进行测试，一般用于检查物编的生成结果
```

清理缓存

```
> sdk.exe clear demo //清理构建的临时文件
```

更新日志

项目当前由 xxqq9527 继续被动维护

2021/07 ~ 现在

- chore(*): 优化
- fix(*): 修复

2021/06

- feat(debug): 新增handle监听
- feat(attr): 新增吸魔 xxqq9527
- feat(ttg): 飘浮字优化 xxqq9527
- fix(monitor): 修复恢复监听器在满值无效的bug xxqq9527
- fix(event): 修复一些事件参数相关bug xxqq9527
- fix(skill): 修复bomb特效无效的问题 xxqq9527

2021/05

- fix(event): 修复hslk系列bugs
- fix(unit): 修复瞬间移动单位面向角度参数缺失bug
- fix(sync): 修复同步参数错位问题

2021/04

- feat(attr): 硬直开启判断及监听优化
- feat(sync): 加入同步
- feat(attr): 三围属性关联删除，现在支持大部分属性的成长关联 attr
- feat(table/hero/skill): 新增几个函数
- feat(hero): 修正经验计算
- feat(attr): 优化攻击间隔逻辑
- feat(attr): 删除xtras里面一些没什么用的事件支持

- feat(event): 玩家伤害数据记录修正
- feat(timer): 计时器新增暂停/恢复方法
- feat(sound): 新增调节bgm音量
- feat(sound): voiceCreate方法新增参数is3D
- feat(dzui): 删减一大批没什么必要的函数
- feat(hslk): 全面支持英雄技能,新增i18n可自定义部分文本
- feat(hslk): 新增hslk_item_fleeting函数, hslk_item增加_onItemGet、hslk_ability增加_onSkillStudy
- feat(math): 两位置角度限制范围返回[0-360]
- fix(sdk): 修复build无响应问题
- fix(attr): 修改伤害数据的获取模式
- fix(model): 暴击默认模型路径错误
- fix(damage): 修正没有伤害也触发暴击特效的问题
- fix(damage): 修复硬直恢复失衡
- chore(*): 删除implant, 不再主动支持war3mapMisc.txt
- chore(hslk): 默认单位杜绝恢复
- chore(hslk): 默认单位ID改为熊猫 (无技能1倍缩放255颜色)

2021/03

- 优化 | V1版本进入维护修bugs阶段, V2版本后成为SDK项目的子库
- feat(skill): damage嵌入原生伤害类型
- feat(texttag): model型漂浮字
- feat(ring): 调整ring的生效逻辑
- feat(japi): 完备JAPI
- fix(sdk): 修复_attr disabled无效果的问题
- fix(monitor/unit): 修复魔法恢复无效的bug
- chore(slk): 修正lua基准库的执行流程

2021/02

- chore(sdk): new hslk
- chore(sdk): new tools
- 致命 | onChat事件优化及重做
- 致命 | 镜头提升/下降指令改为-d, 使用为-d +100[提升100距离]
- 致命 | 淡化runtime
- 功能 | 新模块cache、monitor
- 功能 | 总属性在使用set时支持自动变换为绿字
- 优化 | 重做并优化满金币自动换木头的逻辑

2021/01

- 致命 | 删除了实际无用的三个属性: 冥想力meditative、幸运luck和救助力help
- 致命 | 删减player自带的黄金木头获取漂浮字, 新增资源变动事件, 交由作者自行编写
- 致命 | 删减unit自带的经验漂浮字, 新增单位获得经验事件, 交由作者自行编写
- 致命 | 参数high改为height
- 致命 | 新增黑色玩家颜色、敌人模块重做, 现在配置更简单并且支持多势力
- 致命 | loop方法统一改名为forEach, 并且回调都以(val, idx)的配置返回
- 致命 | includes、delete方法参数顺序改得与lua默认函数相似

- 致命 | 框架现在不处理护甲属性，全部交予原生处理
- 致命 | 新增总护甲属性defend（只读），现在护甲分白绿: defend_white/defend_green
- 致命 | 修复了一些有可能造成异步的潜在bug
- 功能 | xtras支持负动态映射数值
- 功能 | leap现在支持横纵偏移
- 功能 | 新模块物品池
- 功能 | 新增属性：攻击范围attack_range，攻击间隔attack_space
- 优化 | 重做单位RGBA颜色
- 优化 | 现在attr返回的攻击间隔不再是初始间隔而是根据攻速变幻的每秒间隔数据
- 优化 | 调整了丢弃物品触发的时机，事实上物品事件基本已重做
- 优化 | 遮罩mark增加一个空判断
- 优化 | slkHelper物品的叠加数现在下限起码是使用次数
- 优化 | 修复整数属性在小数扣除时有可能多出额外属性的bug
- 优化 | 优化shadow物品的名称

2020/12

- 致命 | slk与hslk分离
- 致命 | 自然属性参数全部删除
- 致命 | 彻底删除了自带的镜头模式
- 功能 | 新增附魔、附魔反应；属性系统与附魔连通
- 功能 | 新增商店模块
- 功能 | 新增buff模块
- 优化 | 英雄不再根据主属性赋予攻击伤害类型

2020/11

- 致命 | slkHelper的自定义键值统一为：以下划线开头小写作key
- 致命 | 伤害系统大改：删除了伤害种类，新增伤害来源等
- 致命 | 属性系统的特效重做，新设xtras
- 功能 | 新增n2i系列函数，名称获得ID
- 功能 | 事件event与新的xtras连通
- 功能 | 新新增新属性：总攻击、攻击间隔、力量、敏捷、智力、减伤（%）、自身暴击
- 优化 | 删除了“被/造成无法回避的伤害”事件

2020/10

- 致命 | 现在物品创建不再支持设定物品的栏目位置
- 致命 | 修复部分range没有改名为radius的问题
- 致命 | 修复attr对象类型添加时key获取错误的问题
- 优化 | 优化合成/拆分的执行效率
- 优化 | 光环目标判定新增enemy目标，与enemies同效果
- 优化 | slkHelper信使的自动技能改为局部参数

2020/09

- 致命 | 为辨明范围的距离，涉及的参数名或回调数据不再使用“range”改为“radius/diameter”半径或直径来处理
- 致命 | [hero]修复设置玩家英雄最大数无效的bug

- 致命 | 修复物品拆分bugs
- 致命 | ID_ARRAY参数改名
- 功能 | slkHelper信使技能拓展，新增四个框架信使行为事件:(闪烁,范围拾取,拆分,批量传送)
- 功能 | event新增移动捕捉的相关事件
- 功能 | 新增matcher
- 功能 | slkHelper[光环]技能的完善 (相关bug可看项目历史)
- 功能 | is模块部分判断函数名改为更贴近原生j,新增sapper判断自爆工兵
- 优化 | 物品拆分等功能继续完善
- 优化 | 整合并优化物品/技能属性计算

2020/08

- 致命 | 修复设置魔法百分比却设置到生命值的bug
- 致命 | 删除了h-lua的变身技能
- 致命 | 修复event删除事件的bug
- 致命 | 修复event一个BJ误用错误，现在改回CJ
- 致命 | 修复event[Esc]事件错误
- 致命 | 修复生命魔法的恢复器有可能出错的问题
- 致命 | 修复print_mbr部分打印没有正确显示中文的问题
- 致命 | 修复硬直已经补满了依然增加的bug
- 功能 | slkHelper现在支持一些简易的自定义设定，如文本颜色等
- 功能 | shadow物品 (默认开启)
- 功能 | 物品合成、拆分 (相关bug可看项目历史)
- 功能 | 强化物品使用事件，现在物品技能如有可能，可以获取释放的目标坐标/目标单位
- 优化 | runtime的精简和优化
- 优化 | hevent.POOL现在可以归nil
- 优化 | 英雄重生方法增加一个如果已经复活则跳过的逻辑
- 优化 | 修正超重提示的小数点溢出
- 优化 | 现在 print_r 会显示数值类型

2020/07

- 致命 | 修复onBeAttack的triggerUnit数据错误
- 功能 | 增加一个0秒无敌的致命抵抗，减少生命技能的性能负担
- 优化 | 删除Dzapi一行多余的代码

2020/06

- 致命 | 修复硬直不会自动补充的bug，修复被击硬直计算不实时的问题 (顺便优化并修正了硬直的判断准则)
- 致命 | 修复hero.levelUp无效的bug
- 功能 | 增加his两个判断：背向对与正面对
- 功能 | 新增[可破坏物被破坏]事件
- 功能 | slkHelper新增[光环]技能
- 功能 | 新增matchUsed方法，使物品使用绑定更快更优雅
- 功能 | 新增[暂停]、[恢复]单位方法
- 优化 | leap和crashFly的地图边界问题
- 优化 | 优化slkHelper的item

- 优化 | hunit创建现在可以带一个attr设定，顺便设置属性

2020/05/31

- 功能 | 去掉了格式化里W（万）这个数值单位

2020/05/30

- 致命 | 修复一个runtime空值bug

2020/05/26

- 优化 | print_mb现在和print一样支持多个参数

2020/05/25

- 优化 | 单位的初始化，英雄的初始化以及判断的自动化

2020/05/19

- 功能 | hero模块构建酒馆内实现增强

2020/05/18

- 优化 | 重做了dzapi服务器清理server.clear

2020/05/17

- 致命 | slkhelper增加内置方法与hRuntime结合构建物编对象
- 优化 | 修改了math.numberFormat的单位

2020/05/14

- 优化 | F9提示优化，并增加all参数，一key全配置
- 优化 | table.merge方法支持动态参数(多table简单合并)

2020/05/09

- 致命 | 更多地框架内实现，group不再使用原生方法进行选取单位，可以选取到hunit.create的所有单位
(如果原来就使用框架内create方法，几乎对现有功能无影响)

2020/05/08

- 致命 | event优化pool的处理，优化掉了框架拓展trigger的引用
- 优化 | 初始化合并,优化dzapi关闭时的提示，优化掉了框架拓展trigger的引用

2020/05/07

- 优化 | 新增item模块瞬逝物fleeting方法

2020/05/06

- 功能 | 新增多频对单位伤害技能 hskill.damageStep
- 优化 | 修改bgm的播放逻辑流程(改善了bgm的播放顺序效率)

2020/05/05

- 致命 | 修复quest离奇大小写导致任务全部是"未发现"的bug
- 致命 | 修复眩晕技能，计时器参数调用错误的bug(会导致多次眩晕直接报错)
- 功能 | sound模块删除了绑定点音效，新增绑定区域音效
- 功能 | 新增方法 his.inRect 判断是否在区域内
- 优化 | 删除了h-lua默认的F9任务 (需要的可自行调用)

2020/05/04

- 致命 | 修复bgm停止后可能无法再次播放的bug

2020/05/01

- 致命 | 复活的方法从unit转到了hero模块

2020/04/30

- 致命 | 修复了一个神奇的heroPick事件调用错误
- 功能 | 新增三围主属性对白字攻击的影响
- 功能 | 现在遮罩自带一些贴图并支持RGB颜色
- 优化 | 修改了hero的底层判断方式 (不影响his.hero的使用)

2020/04/29

- 致命 | 现在time模块不再支持原生计时器对象的获取时间了，全部改为框架自实现
- 致命 | 修复timer索引有误的bug (可能导致计时器无法删除)
- 致命 | 修复胜利失败不会自动结束游戏的问题
- 致命 | 修复物品事件无法注册的bug
- 优化 | 继续完善英雄选择

2020/04/26

- 优化 | 调整skill的damage，现在伤害动作可以没有伤害来源

2020/04/25

- 致命 | 修复is模块的时间判断bug，修复天气模块的一些bug
- 致命 | 删除mark模块 (遮罩) 归并到新增的texture模块
- 优化 | 修复了一些技能边界问题

2020/04/23 - 24

- 致命 | message模块删除，echo方法现在是全局方法，同时合并实现echo00等方法
- 优化 | hero模块重构
- 优化 | env模块重构 (迷雾方法现在归属在env模块)

2020/04/22

- 优化 | 更优质的事件顺序编排

2020/04/20

- 致命 | 不再内置BJ文件，如需使用，自行补充和引用

2020/04/15 - 20

- 致命 | 删除了event的onSummon (被召唤) 事件
- 致命 | 修复击飞目标参数错误的bug
- 功能 | 新增被反伤事件
- 优化 | 删除了上下左右和选择单位对apm的影响
- 优化 | event充分利用pool，重构大部分的方法

2020/04/14

- 致命 | - 修复event区域进入离开参数错误的bug

2020/03/31

- 功能 | - 整合随机数

2020/03/30

- 致命 | - 修复一个物品拾取事件变量[pickup]命名丢失错误

2020/03/28

- 优化 | - 利用event注册池，动态注册事件动态分配，极大降低框架默认注册崩溃的可能性

2020/03/23

- 致命 | - 修复unit.del持续时间为0的注册单位变量无法释放的问题

2020/03/22

- 优化 | - 重实现整理DZapi

2020/03/16

- 功能 | - 增加一个商城作弊法
- 优化 | - strpos现在找不到返回false而不是-1

2020/03/06

- 功能 | - string新增一个strpos方法
- 优化 | - 优化isOwnItem,现在同时支持int和string类型的id

2020/02/29

- 优化 | - 整合区间距离计算

2020/02/24 ~ 27

- 优化 | - 计时器与窗口分离，去除计时器调用时的闭包泄漏
- 优化 | - 改变filter的使用，用来解决group的闭包泄漏
- 优化 | - 重做物品拾取方法，改成框架自实现，不再使用jass方法，用来解决引擎闭包泄漏

2020/02/22

- 致命 | - 弃用pairs

2020/02/19

- 优化 | - time模块新增一个池，用于计时器handle的复用，经测试计时器即使删除内存也不释放，此举解决了内存溢出的问题
- 功能 | - time模块delDialog方法删除，现在调用delTimer会自动删除对话框，而回调也不再提供dialog的handle

2020/02/18

- 功能 | - 现在debug文件里自带调试开启，并且进行了常规错误的捕获，更方便突发测试，自动调试在打包时会自动删除

2020/02/17

- 优化 | - texttag设置了一个limit值作用限制上限，解决lua调用jass函数无效却占用资源丢失跟踪无法释放内存的问题

2020/02/15

- 功能 | - 新增打包package工具用于上线,复制exampleini,双击package打包
- 优化 | - 统合group

2020/02/14

- 致命 | - 重构dzapi！现在有个plugin文件复制到触发开启dzapi
- 功能 | - 新增一个玩家离开游戏的事件

2020/02/13

- 功能 | - 新增is两个判断：单位是否已被删除,单位是否正在受伤

2020/02/12

- 致命 | - 优化加载步骤，不再需要构建初始化触发器执行作弊码指令
- 致命 | - 删除了区域注册的初始化过程，转移至hunit(以后不通过hunit来创建单位就不会被系统识别)
- 功能 | - hunit.create新增register,isOpenSolt参数，可以控制是否注册和注册物品栏
- 功能 | - 新增获取排行榜名次玩家的3个方法
- 功能 | - 新增玩家名称处理的2个方法

2020/02/10

- 功能 | - player模块新增一个隐藏所有单位方法

2020/02/09

- 致命 | - 修复缴械对象获取错误的bug
- 致命 | - 修复is.get错误的bug

2020/02/08

- 致命 | - 修复了之前改动的眩晕以致有时无效的bug
- 功能 | - string 新增方法 vkey,attr调用此方法，提升set的效率大概20倍

- 优化 | - 反伤数值机制精度为0.01, 小于0.01无效

2020/02/06

- 优化 | - 物品属性系统的完成及内存优化

2020/02/05

- 优化 | - 部分attr的slk方法转移至slk-helper

2020/02/04

- 致命 | - math删除了两个已经无用的函数 oddsItem coverAttrEffectVal
- 功能 | - 新增一个quest模块
- 功能 | - debug新增rem方法, 用于检测执行时间
- 优化 | - 修复hluaF9事件错误的bug

2020/02/02

- 功能 | - 新增一个方法判断是否隐身
- 优化 | - 无敌的作用机制改为技能的Avul
- 优化 | - 删除了源的框架属性设定
- 优化 | - 优化SLK预读

2020/02/01

- 优化 | - 删除了全局注册这个事件
- 优化 | - 排行榜现在自动根据数值降序

2020/01/31

- 功能 | - item新增方法pick

2020/01/30

- 致命 | - 修复attr没有处理护甲和魔抗在伤害中比率的问题

2020/01/29

- 优化 | - runtime的gc
- 优化 | - 黄金木材资源的ratio生效机制更改, 现只需要调用add方法会自动结合比率计算, 如果add方法设定了单位, 还会自动展示漂浮字
- 优化 | - 漂浮字的动画和呈现的效率

2020/01/27

- 致命 | - 修复attr-effect不能正确删除的bug
- 功能 | - 新增一个蝗虫的is判断
- 优化 | - 超过漂浮字上限的优化处理

2020/01/26

- 优化 | - 新增一个group方法loop, 新增一个技能, 矩形打击

2020/01/25

- 优化 | - 伤害增幅的key改为damage_extent

2020/01/24

- 鼠年快乐!

2020/01/23

- 致命 | - 修复了attr对于string数值处理不妥的bug
- 功能 | - 新增技能leapReflex

2020/01/22

- 致命 | - 修复小数缓存的微妙差值bug
- 功能 | - 新增属性：神/神抗
- 优化 | - 修复swim技能damageType获取错误对象的bug

2020/01/20

- 功能 | - 新增unit方法，设置单位的动画速度

2020/01/19

- 功能 | - 新增string方法：子串统计
- 功能 | - 反伤增加无敌判定(无敌不会被反伤)

2020/01/18

- 致命 | - 修复event的handle对象错误bug
- 功能 | - 新增一个string方法，用于重复字符串

2020/01/17

- 重构 | - 增加了一个常规伤害的设定
- 致命 | - 修复一部分属性小数精度丢失的bug
- 功能 | - 分裂、物理暴击、魔法暴击现在不属于常规属性，改为effect类型

2020/01/16

- 功能 | - 属性系统的伤害动作全部与skill的合并，现在只需要使用hskill.damage,而hattr.huntUnit则被删除

2020/01/15

- 功能 | - 新增leap拓展技能leapRange
- 功能 | - 新增技能：隐身/现形，优化hskill及hattr
- 功能 | - 修复effect一些关于更新迭代的bug

2020/01/14

- 致命 | - 修复start.lua错误判断蝗虫存在的bug
- 致命 | - 修复math的bug
- 功能 | - 无敌和群体无敌技能，增加一个可以设置特效的参数

- 功能 | - leap技能重做，修复math库的bug
- 功能 | - 新增leap拓展技能pow

2020/01/13

- 功能 | - 新增dzapi封装、unit方法
- 功能 | - 新增技能，剑刃风暴

2020/01/12

- 重构 | - damage函数的优化
- 致命 | - 修复眩晕技能持续时间过短时，无法准确删除buff的bug
- 功能 | - 新增玩家方法get/set称号
- 优化 | - 修复属性系统临时增加的物品栏没被删除的bug
- 优化 | - 增加一个魔爆特效
- 优化 | - 修复新版事件触发技能事件有可能获取数据错误的bug

2020/01/11

- 重构 | - hevent模块重做，影响获取回调数据的方式

2020/01/10

- 功能 | - 新增一个修改单位颜色的方法
- 优化 | - 排行榜的数据设定法修改(改成和多面板差不多)

2020/01/09

- 致命 | - 修复了闪电效果的无效偏移bug及技能闪电链的一些bug
- 功能 | - group新增获取最近单位方法，skill闪电链优化单位选取(最近的敌人)

2020/01/08

- 重构 | - hlogic模块直接删除，转为基本库的math
- 重构 | - mark模块重做，同时修复一些BJ问题
- 致命 | - 修复属性系统视野，绿字攻击，攻速，护甲的减除bug
- 致命 | - 修复slk攻速技能数值错误的问题
- 致命 | - 修复负护甲负魔抗失效的bug
- 优化 | - 调节多面板的图标占位
- 优化 | - 修复BJ、player bug

2020/01/07

- 功能 | - 新增table模块，而system被拆分为table和string，不复存在
- 功能 | - 新增string基本库，原来的idchar转换方法从system删除，内置于string
- 功能 | - 新增多面板，排行榜增加两个设定参数的方法,修复一个skill的bug
- 功能 | - group新增坐标法，优化原有的点、单位创建
- 功能 | - unit增加一个方法获取单位名字
- 功能 | - 增加了一个技能【范围眩晕】
- 优化 | - 修改effect、buff的特效作用顺序
- 优化 | - 优化单位、物品的id问题

2020/01/06

- 重构 | - skill大更新，部分技能改为options参数模式,眩晕等技能现在自动对抵抗作出反应
- 优化 | - 修复了新版table层次不正确的bug
- 优化 | - 属性系统的实现，MD5的调用
- 优化 | - 修正buff的删除

2020/01/05

- 重构 | - 属性系统大更新

2020/01/04

- 功能 | - 现在创建物品时如果是给予单位，可选物品的位置
- 优化 | - 修复了删除技能必须要设置during的不人性化问题

2020/01/03

- 致命 | - 修复了hattr力敏智的计算出错的bug
- 功能 | - 自然属性新增“虫”insect属性

2020/01/01

- 致命 | - 修复属性系统的特性bug
- 功能 | - 现在hEvent的GetTriggerSkill返回的不再是handleID而是charID
- 功能 | - award模块与unit连通(主要是exp的改动)
- 优化 | - 修复buff/debuff无效的bug (key错误)
- 优化 | - 修复了一些unit模块的bug

2019/12/31

- 优化 | - 修复了hColor亮紫色颜色错误的bug

2019/12/30

- 优化 | - 修改item获取slk的部分方法, getAvatar改为getArt (获取图标) ,getModel改为getFile (获取模型路径)

2019/12/29

- 优化 | - 修复一个关于玩家胜利失败的BJbug

2019/12/25

- 功能 | - dialog现在支持table的buttons，同时会自动识别设置的key指，可能的话作为热键
- 功能 | - 优化物品，修复自动物品不会触发使用事件的bug
- 功能 | - 优化system的错误提示
- 功能 | - 新增hhero方法：设置英雄当前等级

2019/12/24

- 功能 | - 优化一些拥有持续时间的函数，during可以不填
- 功能 | - 对话框现在支持关联table及序列table

- 优化 | - 修复玩家获取地图区域调用了BJ函数的问题

2019/12/23

- 功能 | - 增强debug(mb方法)
- 功能 | - his新增判断是否敌军、友军的方法

2019/12/22

- 功能 | - 新增物品的满格、超重事件、属性计算
- 优化 | - 修复runtime的参数重名bug，优化item和unit的slk数据获取

2019/12/21

- 功能 | - htime最重要的两个计时器方法的参数顺序变更，优化nil的编写
- 功能 | - 现在system的print_r归到debug里面去了
- 优化 | - 删除系统默认的空冷却slk

2019/12/20

- 功能 | - skills删除了0秒无敌（合并到无敌的方法去了）
- 功能 | - slk新增一棵树，修复镜头有几率出错的bug
- 功能 | - 新增一波技能的通用方法(添加、删除)
- 优化 | - 修复BJ函数错误

2019/12/18

- 功能 | - effect现在只支持 $>=0$ (0为删除型特效调用法)
- 优化 | - 修复已知的bug

2019/12/16

- 功能 | - 现在onLevelUp事件支持获取提升了的等级差
- 优化 | - 修了很多bug (sound、event、hero、player等)

2019/12/15

- 功能 | - 加入leaderBoard (排行榜) 功能
- 优化 | - 修复了is库使用了BJ函数导致的bug
- 优化 | - 修复技能、属性系统，飘浮字，事件一些bug，优化一些效果
- 优化 | - 调整默认的奖励范围，删除一些调试
- 优化 | - 修复玩家金钱木材设置无效的bug
- 优化 | - 修复runtime bug
- 优化 | - 同步unit库在其他库中的引用

2019/12/13

- 优化 | - 修复计时器一次性时窗口不显示的bug

2019/12/12

- 致命 | - 修复创建单位初始index不正确的bug
- 致命 | - 修复 timer bug

- 功能 | - 加入dialog (对话框) , 用更精简的方法优化msg
- 功能 | - 删除无用的技能空冷却, 增加一个栈信息打印方便调试, 优化代码中判断

2019/02/21 ~ 2019/10/11

- 功能 | - 功能补充

2019/02/21

- start

:功能库

API大全

- [B]blizzard.lua
- [L]**console\build.lua**

```
print(...)  
print_stack(...)  
print_mb(...)  
print_err(val)  
print_r(t, printMethod, showDetail)  
print_mbr(t)
```

- [L]**console\hslk.lua**

```
rem(key1, key2)  
print_stack(...)  
print_r(t)  
sub_print_r(tt, indent)
```

- [L]**console\test.lua**

```
JassRuntime.error_handle(msg)  
print(...)  
print_stack(...)  
print_mb(...)  
print_err(val)  
print_r(t, printMethod, showDetail)  
sub_print_r(tt, indent)  
print_mbr(t)
```

- [B]const\abilityTarget.lua
- [B]const\attribute.lua
- [B]const\attributeXtras.lua
- [B]const\breakArmorType.lua

- [B]const\cache.lua
- [B]const\damageSource.lua
- [B]const\damageType.lua
- [B]const\enchant.lua
- [B]const\event.lua
- [B]const\hero.lua
- [B]const\hotKey.lua
- [B]const\item.lua
- [B]const\keyboard.lua
- [B]const\monitor.lua
- [B]const\target.lua
- [B]const\textTag.lua
- [L]**const\ubertip.lua**

```
CONST_UBERTIP_TARGET_LABEL(target, actionType, actionField, isValue)
CONST_UBERTIP_IS_PERCENT(key)
CONST_UBERTIP_IS_LEVEL(key)
CONST_UBERTIP_ATTR_XTRAS(data, indent)
CONST_UBERTIP_ATTR(attr, sep, indent)
CONST_UBERTIP_RING_ABILITY(data)
CONST_UBERTIP_RING_ITEM(data)
CONST_RESEARCH_UBERTIP_ATTR(attr)
CONST_RESEARCH_UBERTIP_RING_ABILITY(data)
```

- [B]const\unit.lua
- [L]**echo.lua**

```
echo(msg, whichPlayer, duration, x, y)
```

- [B]engine.lua
- [L]**foundation\Mapping.lua**

```
Mapping:new()
Mapping:length()
Mapping:set(key, value)
Mapping:get(key)
Mapping:del(...)
Mapping:forEach(action)
```

- [L]**foundation\color.lua**

```
hcolor.hex(str, color)
hcolor.mixed(str, color)
hcolor.gold(str)
```

```
hcolor.white(str)
hcolor.black(str)
hcolor.grey(str)
hcolor.greyDeep(str)
hcolor.redLight(str)
hcolor.red(str)
hcolor.greenLight(str)
hcolor.green(str)
hcolor.yellowLight(str)
hcolor.yellow(str)
hcolor.orangeLight(str)
hcolor.orange(str)
hcolor.skyLight(str)
hcolor.sky(str)
hcolor.seaLight(str)
hcolor.sea(str)
hcolor.purpleLight(str)
hcolor.purple(str)
```

- [L]**foundation\json.lua**

```
kind_of(obj)
escape_str(s)
skip_delim(str, pos, delim, err_if_missing)
parse_str_val(str, pos, val)
parse_num_val(str, pos)
json.stringify(obj, as_key)
json.parse(str, pos, end_delim)
```

- [L]**foundation\math.lua**

```
math.random(n, m)
math.polarProjection(x, y, dist, angle)
math.round(decimal, n)
math.numberFormat(value, n)
math.integerFormat(value)
math.getDegBetweenXY(x1, y1, x2, y2)
math.getDegBetweenUnit(u1, u2)
math.getDistanceBetweenXY(x1, y1, x2, y2)
math.getDistanceBetweenUnit(u1, u2)
math.getMaxDistanceInRect(w, h, deg)
math.date(timestamp)
```

- [L]**foundation\string.lua**

```
string.char2id(idChar)
string.id2char(id)
```

```

string.mb_len(inputStr)
string.vkey(t)
string.addslashes(s)
string.stripslashes(s)
string.base64Encode(source_str)
string.base64Decode(str64)
string.explode(delimiter, str)
string.implode(delimiter, table)
string.split(str, size)
string.mb_split(str, size)
stringstrpos(str, pattern)
string.findCount(str, pattern)
string.attrBuffKey(val)
string.convert(dec, cvt)

```

- [L]**foundation\table.lua**

```

table.len(table)
table.random(arr)
table.shuffle(arr)
table.clone(org)
copy(org1, res)
table.merge(...)
table.includes(arr, val)
table.delete(arr, val, qty)
table.value(arr, key)
table.obj2arr(obj, keyMap)

```

- [B]h-lua.lua
- [L]**lib\attribute\setter.lua**

```

hattributeSetter.getDecimalTemporaryStorage(whichUnit, attr)
hattributeSetter.setDecimalTemporaryStorage(whichUnit, attr, value)
hattributeSetter.relyRegister(whichUnit)
hattributeSetter.setUnitMaxLife(whichUnit, currentVal, futureVal, diff)
hattributeSetter.setUnitMaxMana(whichUnit, currentVal, futureVal, diff)
hattributeSetter.setUnitAttackWhite(whichUnit, futureVal, diff)
hattributeSetter.setUnitAttackGreen(whichUnit, futureVal)
hattributeSetter.setUnitAttackRange(whichUnit, futureVal)
hattributeSetter.setUnitAttackSpace(whichUnit, futureVal)
hattributeSetter.setUnitAttackSpeed(whichUnit, futureVal)
hattributeSetter.setUnitDefendWhite(whichUnit, futureVal)
hattributeSetter.setUnitDefendGreen(whichUnit, futureVal)
hattributeSetter.setUnitSight(whichUnit, futureVal)
hattributeSetter.setUnitThree(whichUnit, futureVal, attr, diff)
hattributeSetter.relation(whichUnit, attr, diff)

```

- [L]**lib\attribute\xtras.lua**

```
hattribute.xtrasPassAlive(targetUnit, type, field)
hattribute.getXtras(whichUnit, eventKey)
hattribute.hasXtras(whichUnit, eventKey)
hattribute.xtras(triggerUnit, eventKey, evtData)
```

- [L]**lib\attribute.lua**

```
hattribute.isValType(field, valType)
hattribute.setRelation(relation)
hattribute.init(whichUnit)
hattribute.setHandle(whichUnit, attr, opr, val, during)
hattribute.set(whichUnit, during, data)
hattribute.get(whichUnit, attr, default)
hattribute.caleAttribute(damageSrc, isAdd, whichUnit, attr, times)
hattribute.getArmorReducePercent(defend)
```

- [L]**lib\award.lua**

```
haward.setShareRange(range)
haward.forUnit(whichUnit, exp, gold, lumber)
haward.forUnitExp(whichUnit, exp)
haward.forUnitGold(whichUnit, gold)
haward.forUnitLumber(whichUnit, lumber)
haward.forGroup(whichUnit, exp, gold, lumber)
haward.forGroupExp(whichUnit, exp)
haward.forGroupGold(whichUnit, gold)
haward.forGroupLumber(whichUnit, lumber)
haward.forPlayer(gold, lumber)
haward.forPlayerGold(gold)
haward.forPlayerLumber(lumber)
```

- [L]**lib\buff.lua**

```
hbuff.uniqueKey()
hbuff.count(handleUnit, groupKey)
hbuff.create(during, handleUnit, groupKey, diff, purpose, rollback)
hbuff.purpose(handleUnit, buffKey)
hbuff.delete(handleUnit, buffKey)
```

- [L]**lib\cache.lua**

```
hcache.len()
hcache.exist(handle, key)
hcache.alloc(handle)
```

```
hcache.protect(handle)
hcache.protected(handle)
hcache.free(handle, key)
hcache.set(handle, key, value)
hcache.get(handle, key, default)
```

- [L]lib\camera.lua

```
hcamera.reset(whichPlayer, during)
hcamera.apply(whichPlayer, during, camerasetup)
hcamera.toXY(whichPlayer, during, x, y)
hcamera.toUnit(whichPlayer, during, whichUnit)
hcamera.lock(whichPlayer, whichUnit)
hcamera.changeDistance(whichPlayer, diffDistance)
hcamera.shock(whichPlayer, whichType, during, scale)
```

- [L]lib\cmd.lua

```
hcmd.conf(commands, playerIndexes)
```

- [L]lib\dialog.lua

```
hdialog.hotkey(key)
hdialog.del(whichDialog)
hdialog.create(whichPlayer, options, action)
```

- [B]lib\dzapi\cache.lua
- [L]lib\dzapi.lua

```
hdzapi.isRPGLobby()
hdzapi.isRPGLadder()
hdzapi.isRedVip(whichPlayer)
hdzapi.isBlueVip(whichPlayer)
hdzapi.isPlatformVIP(whichPlayer)
hdzapi.mapLevel(whichPlayer)
hdzapi.hasMallItem(whichPlayer, key)
hdzapi.timestamp()
hdzapi.date()
hdzapi.loadServer(whichPlayer, key)
hdzapi.loadServerBool(whichPlayer, key)
hdzapi.loadServerNumber(whichPlayer, key)
hdzapi.loadServerInteger(whichPlayer, key)
hdzapi.roulette(whichPlayer, method, params)
hdzapi.saveServer(whichPlayer, key, value)
```

```
hdzapi.clearServer(whichPlayer, key)
hdzapi.setRoomStat(whichPlayer, key, value)
```

- [B]lib\dzui\cache.lua
- [L]lib\dzui.lua

```
hdzui.wideScreen()
hdzui.hideInterface()
hdzui.loadToc(tocFilePath)
hdzui.frame(fdfName, parent, id)
hdzui.frameTag(fdfType, fdfName, parent, tag, id)
hdzui.framePoint(frameId, relation, align, alignRelation, x, y)
hdzui.onMouse(frameId, mouseOrder, callFunc)
hdzui.miniMapTrack(options)
```

- [L]lib\effect.lua

```
heffect.del(e)
heffect.toXY(effectModel, x, y, during)
heffect.toUnit(effectModel, targetUnit, during)
heffect.bindUnit(effectModel, targetUnit, attach, during)
```

- [L]lib\enchant.lua

```
henchant.setIntrinsicAddition(percent)
henchant.enableAppend(status)
henchant.setAppendAttachEffect(whichEnchant, effects)
henchant.setEnvReaction(onEnchant, toEnchant, reaction)
henchant.append(options)
```

- [L]lib\enemy.lua

```
henemy.set(name, color, playerIndexes, isShareSight)
henemy.getPlayer(createQty, teamNo)
henemy.isShareSight(teamNo)
henemy.create(options)
```

- [L]lib\env.lua

```
henv.setFogStatus(enableFog, enableMark)
henv.clearDestructable(whichRect)
henv.build(whichRect, typeStr, isInvulnerable, isDestroyRect, ground, doodad,
```

```
units)
henv.random(whichRect, typeStr, isInvulnerable, isDestroyRect)
```

- [L]**lib\event\defaultActions.lua**

- [L]**lib\event.lua**

```
hevent.free(handle)
hevent.pool(handle, conditionAction, regEvent)
hevent.setLastDamage(sourceUnit, targetUnit)
hevent.getUnitLastDamageSource(whichUnit)
hevent.getUnitLastDamageTarget(whichUnit)
hevent.getPlayerLastDamageTarget(whichPlayer)
hevent.registerEvent(handle, key, callFunc)
hevent.triggerData(triggerData)
hevent.hslk(key, triggerData)
hevent.triggerEvent(handle, key, triggerData)
hevent.deleteEvent(handle, key, eventId)
hevent.onAttackDetect(whichUnit, callFunc)
hevent.onAttackGetTarget(whichUnit, callFunc)
hevent.onBeAttackReady(whichUnit, callFunc)
hevent.onAttack(whichUnit, callFunc)
hevent.onBeAttack(whichUnit, callFunc)
hevent.onSkill(whichUnit, callFunc)
hevent.onBeSkill(whichUnit, callFunc)
hevent.onSkillStudy(whichUnit, callFunc)
hevent.onSkillReady(whichUnit, callFunc)
hevent.onSkillCast(whichUnit, callFunc)
hevent.onSkillStop(whichUnit, callFunc)
hevent.onSkillEffect(whichUnit, callFunc)
hevent.onSkillFinish(whichUnit, callFunc)
hevent.onItem(whichUnit, callFunc)
hevent.onBeItem(whichUnit, callFunc)
hevent.onItemUsed(whichUnit, callFunc)
hevent.onItemDrop(whichUnit, callFunc)
hevent.onItemGet(whichUnit, callFunc)
hevent.onItemPawn(whichUnit, callFunc)
hevent.onItemSell(whichUnit, callFunc)
hevent.onUnitSell(whichUnit, callFunc)
hevent.onItemDestroy(whichItem, callFunc)
hevent.onItemSeparate(whichItem, callFunc)
hevent.onItemSynthesis(whichUnit, callFunc)
hevent.onItemOverWeight(whichUnit, callFunc)
hevent.onItemOverSlot(whichUnit, callFunc)
hevent.onDamage(whichUnit, callFunc)
hevent.onBeDamage(whichUnit, callFunc)
hevent.onDamageResistance(whichUnit, callFunc)
hevent.onAvoid(whichUnit, callFunc)
```

```
hevent.onBeAvoid(whichUnit, callFunc)
hevent.onBreakArmor(whichUnit, callFunc)
hevent.onBeBreakArmor(whichUnit, callFunc)
hevent.onSwim(whichUnit, callFunc)
hevent.onBeSwim(whichUnit, callFunc)
hevent.onBroken(whichUnit, callFunc)
hevent.onBeBroken(whichUnit, callFunc)
hevent.onSilent(whichUnit, callFunc)
hevent.onBeSilent(whichUnit, callFunc)
hevent.onUnarm(whichUnit, callFunc)
hevent.onBeUnarm(whichUnit, callFunc)
hevent.onFetter(whichUnit, callFunc)
hevent.onBeFetter(whichUnit, callFunc)
hevent.onBomb(whichUnit, callFunc)
hevent.onBeBomb(whichUnit, callFunc)
hevent.onLightningChain(whichUnit, callFunc)
hevent.onBeLightningChain(whichUnit, callFunc)
hevent.onCrackFly(whichUnit, callFunc)
hevent.onBeCrackFly(whichUnit, callFunc)
hevent.onRebound(whichUnit, callFunc)
hevent.onBeRebound(whichUnit, callFunc)
hevent.onKnocking(whichUnit, callFunc)
hevent.onBeKnocking(whichUnit, callFunc)
hevent.onSplit(whichUnit, callFunc)
hevent.onBeSplit(whichUnit, callFunc)
hevent.onHemophagia(whichUnit, callFunc)
hevent.onBeHemophagia(whichUnit, callFunc)
hevent.onSkillHemophagia(whichUnit, callFunc)
hevent.onBeSkillHemophagia(whichUnit, callFunc)
hevent.onSiphon(whichUnit, callFunc)
hevent.onBeSiphon(whichUnit, callFunc)
hevent.onSkillSiphon(whichUnit, callFunc)
hevent.onBeSkillSiphon(whichUnit, callFunc)
hevent.onPunish(whichUnit, callFunc)
hevent.onDead(whichUnit, callFunc)
hevent.onKill(whichUnit, callFunc)
hevent.onReborn(whichUnit, callFunc)
hevent.onExp(whichUnit, callFunc)
hevent.onLevelUp(whichUnit, callFunc)
hevent.onUpgradeStart(whichUnit, callFunc)
hevent.onUpgradeCancel(whichUnit, callFunc)
hevent.onUpgradeFinish(whichUnit, callFunc)
hevent.onEnterUnitRange(whichUnit, radius, callFunc)
hevent.onEnterRect(whichRect, callFunc)
hevent.onLeaveRect(whichRect, callFunc)
hevent.onConstructStart(whichPlayer, callFunc)
hevent.onConstructCancel(whichPlayer, callFunc)
hevent.onConstructFinish(whichPlayer, callFunc)
hevent.onChat(whichPlayer, pattern, callFunc)
hevent.onEsc(whichPlayer, callFunc)
hevent.onSelection(whichPlayer, qty, callFunc)
hevent.onDeSelection(whichPlayer, callFunc)
hevent.onPlayerLeave(callFunc)
hevent.onPlayerResourceChange(callFunc)
```

```

hevent.onPickHero(callFunc)
hevent.onDestructableDestroy(whichDestructable, callFunc)
hevent.onMapDestructableDestroy(callFunc)
hevent.onCourierBlink(whichUnit, callFunc)
hevent.onCourierRangePickUp(whichUnit, callFunc)
hevent.onCourierSeparate(whichUnit, callFunc)
hevent.onCourierDeliver(whichUnit, callFunc)
hevent.onMoveStart(whichUnit, callFunc)
hevent.onMoving(whichUnit, callFunc)
hevent.onMoveStop(whichUnit, callFunc)
hevent.onHoldOn(whichUnit, callFunc)
hevent.onStop(whichUnit, callFunc)

```

- [L]lib\group.lua

```

hgroup.forEach(whichGroup, action)
hgroup.count(whichGroup)
hgroup.includes(whichGroup, whichUnit)
hgroup.isEmpty(whichGroup)
hgroup.addUnit(whichGroup, whichUnit)
hgroup.removeUnit(whichGroup, whichUnit)
hgroup.createByXY(x, y, radius, filterFunc)
hgroup.createByUnit(u, radius, filterFunc)
hgroup.createByRect(r, filterFunc)
hgroup.getClosest(whichGroup, x, y)
hgroup.portal(whichGroup, x, y, eff, isFollow)
hgroup.animate(whichGroup, animate)
hgroup.clear(whichGroup, isDestroy, isDestroyUnit)

```

- [L]lib\hero.lua

```

hhero.getPrimary(whichHero)
hhero.getPrimaryLabel(whichHero)
hhero.getStrPlus(whichHero)
hhero.getAgiPlus(whichHero)
hhero.getIntPlus(whichHero)
hhero.getProperName(whichHero)
hhero.setPrevLevel(whichHero, lv)
hhero.getPrevLevel(whichHero)
hhero.getCurLevel(whichHero)
hhero.setCurLevel(whichHero, newLevel, showEffect)
hhero.getExp(whichHero)
hhero.getExpNeed(targetLevel)
hhero.getSkillPoints(whichHero)
hhero.setPlayerAllowQty(whichPlayer, max)
hhero.getPlayerAllowQty(whichPlayer)
hhero.setBornXY(x, y)
hhero.reborn(whichHero, delay, invulnerable, x, y, showDialog)
hhero.buildSelector(options)

```

- [L]**lib\id.lua**

```
HL_ID_INIT()
```

- [L]**lib\initialization.lua**

- [L]**lib\is.lua**

```
his.night()
his.day()
his.computer(whichPlayer)
his.playerSite(whichPlayer)
his.playing(whichPlayer)
his.neutral(whichPlayer)
his.detected(whichUnit, whichPlayer)
his.hasSlot(whichUnit, slotId)
his.canAttack(whichUnit)
his.dead(whichUnit)
his.alive(whichUnit)
his.deleted(whichUnit)
his.invincible(whichUnit)
his.invisible(whichUnit)
his.hero(whichUnit)
his.courier(whichUnit)
his.structure(whichUnit)
his.illusion(whichUnit)
his.ground(whichUnit)
his.air(whichUnit)
his.melee(whichUnit)
his.ranged(whichUnit)
his.summoned(whichUnit)
his.mechanical(whichUnit)
his.ancient(whichUnit)
his.sapper(whichUnit)
his.ethereal(whichUnit)
his.immune(whichUnit)
his.race(whichUnit, whichRace)
his.locust(whichUnit)
his.swim(whichUnit)
his.enablePunish(whichUnit)
his.punish(whichUnit)
his.silent(whichUnit)
his.unarm(whichUnit)
his.crackFly(whichUnit)
his.beDamaging(whichUnit)
his.damaging(whichUnit)
```

```
his.playerBeDamaging(whichPlayer)
his.playerDamaging(whichPlayer)
his.water(whichUnit)
his.floor(whichUnit)
his.unit(whichUnit, otherUnit)
his.enemy(whichUnit, otherUnit)
his.ally(whichUnit, otherUnit)
his.parallel(u1, u2, maxDistance, forcedOrder)
his.intersect(u1, u2, maxDistance, face2face)
his.enemyPlayer(whichUnit, whichPlayer)
his.allyPlayer(whichUnit, whichPlayer)
his.marking(whichPlayer)
his.inRect(whichRect, x, y)
his.borderRect(whichRect, x, y)
his.borderMap(x, y)
his.borderCamera(x, y)
his.destroy(whichItem)
his.hasItem(whichUnit, whichItemId)
```

- [L]lib\item.lua

```
hitem.x(it)
hitem.y(it)
hitem.z(it)
hitem.embed(u)
hitem.free(whichItem)
hitem.setHolder(whichItem, holder)
hitem.getHolder(whichItem)
hitem.setLastHolder(whichItem, holder)
hitem.getLastHolder(whichItem)
hitem.backToLastHolder(whichItem)
hitem.isRobber(whichItem, robber)
hitem.used(whichUnit, whichItem, triggerData)
hitem.del(it, delay)
hitem.delFromUnit(whichUnit)
hitem.getId(itOrId)
hitem.getName(itOrId)
hitem.isShadowFront(itOrId)
hitem.isShadowBack(itOrId)
hitem.shadowID(itOrId)
hitem.getArt(itOrId)
hitem.getFile(itOrId)
hitem.getClass(itOrId)
hitem.getGoldCost(itOrId)
hitem.getLumberCost(itOrId)
hitem.getIsUsable(itOrId)
hitem.getIsPowerUp(itOrId)
hitem.getIsPerishable(itOrId)
hitem.getIsSellAble(itOrId)
hitem.getOverlie(itOrId)
hitem.getWeight(itOrId, charges)
hitem.getAttribute(itOrId)
```

```
hitem.getLevel(it)
hitem.getCharges(it)
hitem.setCharges(it, charges)
hitem.getTotalCharges(itemId, whichUnit)
hitem.getEmptySlot(whichUnit)
hitem.forEach(whichUnit, action)
hitem.addProperty(whichUnit, itId, charges)
hitem.subProperty(whichUnit, itId, charges)
hitem.overlyingSlot(itemSlot)
hitem.synthesis(whichUnit, items)
hitem.separate(whichItem, separateType, formulaIndex, whichUnit)
hitem.create(options)
hitem.fleeting(fleetingType, x, y, during, yourFunc)
hitem.give(origin, target)
hitem.pick(it, targetUnit)
hitem.copy(origin, target)
hitem.drop(origin, slot)
hitem.pickRect(u, x, y, w, h)
hitem.pickRound(u, x, y, r)
```

- [L]lib\itemPool.lua

```
hitemPool.poolName(poolName)
hitemPool.insert(poolName, whichItem)
hitemPool.delete(poolName, whichItem)
hitemPool.free(whichItem)
hitemPool.forEach(poolName, action)
```

- [L]lib\japi\formatter.lua

```
local toBool(value)
local toInt(value, def)
```

- [L]lib\japi\triumph.lua

```
hjapi.echo(msg)
hjapi.has(method)
hjapi.formatter(method, params)
hjapi.triumph(method, params, result)
hjapi.exec(method, params)
does not exist, please check the WE environment! You should make friends with
5382557(QQ)")
```

```
hjapi.DzAPI_CommonFunc_GetARGBColorValue(...)  
hjapi.DzAPI_CommonFunc_GetARGBColorValuePercent(...)  
hjapi.DzAPI_CommonFunc_SetARGBColorValue(...)  
hjapi.DzAPI_CommonFunc_SetARGBColorValuePercent(...)  
hjapi.DzAPI_Map_ChangeStoreItemCoolDown(...)  
hjapi.DzAPI_Map_ChangeStoreItemCount(...)  
hjapi.DzAPI_Map_GetActivityData()  
hjapi.DzAPI_Map_GetGameStartTime()  
hjapi.DzAPI_Map_GetGuildName(whichPlayer)  
hjapi.DzAPI_Map_GetGuildRole(whichPlayer)  
hjapi.DzAPI_Map_GetLadderLevel(whichPlayer)  
hjapi.DzAPI_Map_GetLadderRank(whichPlayer)  
hjapi.DzAPI_Map_GetMapConfig(key)  
hjapi.DzAPI_Map_GetMapLevel(whichPlayer)  
hjapi.DzAPI_Map_GetMapLevelRank(whichPlayer)  
hjapi.DzAPI_Map_GetMatchType()  
hjapi.DzAPI_Map_GetPlatformVIP(whichPlayer)  
hjapi.DzAPI_Map_IsPlatformVIP(whichPlayer)  
hjapi.DzAPI_Map_GetPublicArchive(whichPlayer, key)  
hjapi.DzAPI_Map_GetServerArchiveDrop(whichPlayer, key)  
hjapi.DzAPI_Map_GetServerArchiveEquip(whichPlayer, key)  
hjapi.DzAPI_Map_GetServerValue(whichPlayer, key)  
hjapi.DzAPI_Map_GetServerValueErrorCode(whichPlayer)  
hjapi.GetPlayerServerValueSuccess(whichPlayer)  
hjapi.DzAPI_Map_GetUserID(...)  
hjapi.DzAPI_Map_HasMallItem(whichPlayer, key)  
hjapi.DzAPI_Map_IsBlueVIP(whichPlayer)  
hjapi.DzAPI_Map_IsRPGLadder()  
hjapi.DzAPI_Map_IsRPGLobby()  
hjapi.DzAPI_Map_IsRedVIP(whichPlayer)  
hjapi.DzAPI_Map_Ladder_SetPlayerStat(whichPlayer, key, value)  
hjapi.DzAPI_Map_Ladder_SubmitPlayerRank(whichPlayer, value)  
hjapi.DzAPI_Map_Ladder_SetStat(whichPlayer, key, value)  
hjapi.DzAPI_Map_Ladder_SubmitTitle(whichPlayer, value)  
hjapi.DzAPI_Map_Ladder_SubmitPlayerExtraExp(whichPlayer, value)  
hjapi.DzAPI_Map_MissionComplete(whichPlayer, key, value)  
hjapi.DzAPI_Map_OrgTrigger(whichPlayer, key)  
hjapi.DzAPI_Map_SavePublicArchive(whichPlayer, key, value)  
hjapi.DzAPI_Map_SaveServerValue(whichPlayer, key, value)  
hjapi.DzAPI_Map_Stat_SetStat(whichPlayer, key, value)  
hjapi.DzAPI_Map_Statistics(whichPlayer, eventKey, eventType, value)  
hjapi.DzAPI_Map_ToggleStore(...)  
hjapi.DzAPI_Map_UpdatePlayerHero(...)  
hjapi.DzAPI_Map_UseConsumablesItem(whichPlayer, key)  
hjapi.DzAPI_UnitType_CountUnitTypedataArrayAbilID(...)  
hjapi.DzAPI_UnitType_CountUnitTypedataArrayBoolean(...)  
hjapi.DzAPI_UnitType_CountUnitTypedataArrayItemID(...)  
hjapi.DzAPI_UnitType_CountUnitTypedataArrayReal(...)  
hjapi.DzAPI_UnitType_CountUnitTypedataArrayString(...)  
hjapi.DzAPI_UnitType_CountUnitTypedataArrayTechID(...)  
hjapi.DzAPI_UnitType_CountUnitTypedataArrayUnitID(...)  
hjapi.DzAPI_UnitType_CountUnitTypeDataRequires(...)  
hjapi.DzAPI_UnitType_CountUnitTypeDataRequiresamount(...)  
hjapi.DzAPI_UnitType_GetEnum_PreventOrReguirePlace(...)
```

```
hjapi.DzAPI_UnitType_GetEnum_PreventOrReguirePlaceCheck(...)  
hjapi.DzAPI_UnitType_GetEnum_Primary(...)  
hjapi.DzAPI_UnitType_GetEnum_TargetTypeCheck(...)  
hjapi.DzAPI_UnitType_GetEnum_TargetTypeSeries(...)  
hjapi.DzAPI_UnitType_GetEnum_armor(...)  
hjapi.DzAPI_UnitType_GetEnum_atkType(...)  
hjapi.DzAPI_UnitType_GetEnum_buffType(...)  
hjapi.DzAPI_UnitType_GetEnum_deathType(...)  
hjapi.DzAPI_UnitType_GetEnum_defType(...)  
hjapi.DzAPI_UnitType_GetEnum_movetp(...)  
hjapi.DzAPI_UnitType_GetEnum_race(...)  
hjapi.DzAPI_UnitType_GetEnum_regenType(...)  
hjapi.DzAPI_UnitType_GetEnum_type(...)  
hjapi.DzAPI_UnitType_GetEnum_typeCheck(...)  
hjapi.DzAPI_UnitType_GetEnum_warpsOn(...)  
hjapi.DzAPI_UnitType_GetEnum_weapTp(...)  
hjapi.DzAPI_UnitType_GetEnum_weapType(...)  
hjapi.DzAPI_UnitType_GetUnitTypeDataAbiliID(...)  
hjapi.DzAPI_UnitType_GetUnitTypeDataArrayAbiliID(...)  
hjapi.DzAPI_UnitType_GetUnitTypeDataArrayBoolean(...)  
hjapi.DzAPI_UnitType_GetUnitTypeDataArrayItemID(...)  
hjapi.DzAPI_UnitType_GetUnitTypeDataArrayReal(...)  
hjapi.DzAPI_UnitType_GetUnitTypeDataArrayString(...)  
hjapi.DzAPI_UnitType_GetUnitTypeDataArrayTechID(...)  
hjapi.DzAPI_UnitType_GetUnitTypeDataArrayUnitID(...)  
hjapi.DzAPI_UnitType_GetUnitTypeDataBoolean(...)  
hjapi.DzAPI_UnitType_GetUnitTypeDataInt(...)  
hjapi.DzAPI_UnitType_GetUnitTypeDataReal(...)  
hjapi.DzAPI_UnitType_GetUnitTypeDataRequires(...)  
hjapi.DzAPI_UnitType_GetUnitTypeDataRequiresamount(...)  
hjapi.DzAPI_UnitType_GetUnitTypeDataString(...)  
hjapi.DzAPI_UnitType_GettUnitTypeDataRequirescount(...)  
hjapi.DzAPI_UnitType_ResizeUnitTypeDataArrayAbiliID(...)  
hjapi.DzAPI_UnitType_ResizeUnitTypeDataArrayBoolean(...)  
hjapi.DzAPI_UnitType_ResizeUnitTypeDataArrayItemID(...)  
hjapi.DzAPI_UnitType_ResizeUnitTypeDataArrayReal(...)  
hjapi.DzAPI_UnitType_ResizeUnitTypeDataArrayString(...)  
hjapi.DzAPI_UnitType_ResizeUnitTypeDataArrayTechID(...)  
hjapi.DzAPI_UnitType_ResizeUnitTypeDataArrayUnitID(...)  
hjapi.DzAPI_UnitType_ResizeUnitTypeDataRequires(...)  
hjapi.DzAPI_UnitType_ResizeUnitTypeDataRequiresamount(...)  
hjapi.DzAPI_UnitType_SetEnum_ProtectOrReguirePlace(...)  
hjapi.DzAPI_UnitType_SetEnum_ProtectOrReguirePlaceModify(...)  
hjapi.DzAPI_UnitType_SetEnum_Primary(...)  
hjapi.DzAPI_UnitType_SetEnum_TargetTypeModify(...)  
hjapi.DzAPI_UnitType_SetEnum_TargetTypeSeries(...)  
hjapi.DzAPI_UnitType_SetEnum_armor(...)  
hjapi.DzAPI_UnitType_SetEnum_atkType(...)  
hjapi.DzAPI_UnitType_SetEnum_buffType(...)  
hjapi.DzAPI_UnitType_SetEnum_deathType(...)  
hjapi.DzAPI_UnitType_SetEnum_defType(...)  
hjapi.DzAPI_UnitType_SetEnum_movetp(...)  
hjapi.DzAPI_UnitType_SetEnum_race(...)  
hjapi.DzAPI_UnitType_SetEnum_regenType(...)
```

```
hjapi.DzAPI_UnitType_SetEnum_type(...)  
hjapi.DzAPI_UnitType_SetEnum_typeModify(...)  
hjapi.DzAPI_UnitType_SetEnum_warpsOn(...)  
hjapi.DzAPI_UnitType_SetEnum_weapTp(...)  
hjapi.DzAPI_UnitType_SetEnum_weapType(...)  
hjapi.DzAPI_UnitType_SetUnitTypeDataAbilID(...)  
hjapi.DzAPI_UnitType_SetUnitTypeDataArrayAbilID(...)  
hjapi.DzAPI_UnitType_SetUnitTypeDataArrayBoolean(...)  
hjapi.DzAPI_UnitType_SetUnitTypeDataArrayItemID(...)  
hjapi.DzAPI_UnitType_SetUnitTypeDataArrayReal(...)  
hjapi.DzAPI_UnitType_SetUnitTypeDataArrayString(...)  
hjapi.DzAPI_UnitType_SetUnitTypeDataArrayTechID(...)  
hjapi.DzAPI_UnitType_SetUnitTypeDataArrayUnitID(...)  
hjapi.DzAPI_UnitType_SetUnitTypeDataBoolean(...)  
hjapi.DzAPI_UnitType_SetUnitTypeDataInt(...)  
hjapi.DzAPI_UnitType_SetUnitTypeDataReal(...)  
hjapi.DzAPI_UnitType_SetUnitTypeDataRequires(...)  
hjapi.DzAPI_UnitType_SetUnitTypeDataRequiresamount(...)  
hjapi.DzAPI_UnitType_SetUnitTypeDataString(...)  
hjapi.DzAPI_UnitstateToInteger(...)  
hjapi.DzClickFrame(frameId)  
hjapi.DzConvertWorldPosition(...)  
hjapi.DzCreateFrame(frame, parent, id)  
hjapi.DzCreateFrameByTagName(frameType, name, parent, template, id)  
hjapi.DzCreateSimpleFrame(frame, parent, id)  
hjapi.DzDebugString(...)  
hjapi.DzDestroyFrame(frameId)  
hjapi.DzDestructablePosition(d, x, y)  
hjapi.DzDotaInfo_IsPlayerRandom(...)  
hjapi.DzDotaInfo_IsRepicked(...)  
hjapi.DzDotaInfo_Item(...)  
hjapi.DzDotaInfo_Item_HE(...)  
hjapi.DzDotaInfo_Item_TM(...)  
hjapi.DzEnableWideScreen(enable)  
hjapi.DzEvent_Building_Cancel(...)  
hjapi.DzEvent_Building_Dead(...)  
hjapi.DzEvent_Building_Finish(...)  
hjapi.DzEvent_Building_Start(...)  
hjapi.DzEvent_Hero_Dead(...)  
hjapi.DzEvent_Hero_Level(...)  
hjapi.DzEvent_Item_Drop(...)  
hjapi.DzEvent_Item_Pickup(...)  
hjapi.DzEvent_Item_Sell(...)  
hjapi.DzEvent_Item_Use(...)  
hjapi.DzEvent_Tech_Cancel(...)  
hjapi.DzEvent_Tech_Finish(...)  
hjapi.DzEvent_Tech_Start(...)  
hjapi.DzEvent_Unit_Cancel(...)  
hjapi.DzEvent_Unit_ChangeOwner(...)  
hjapi.DzEvent_Unit_Dead(...)  
hjapi.DzEvent_Unit_Finish(...)  
hjapi.DzEvent_Unit_Hired(...)  
hjapi.DzEvent_Unit_Start(...)  
hjapi.DzExecuteFunc(funcName)
```

```
hjapi.DzFrameCageMouse(frame, enable)
hjapi.DzFrameClearAllPoints(frame)
hjapi.DzFrameEditBlackBorders(upperHeight, bottomHeight)
hjapi.DzFrameFindByName(name, id)
hjapi.DzFrameGetAlpha(frame)
hjapi.DzFrameGetChatMessage()
hjapi.DzFrameGetCommandBarButton(row, column)
hjapi.DzFrameGetEnable(frame)
hjapi.DzFrameGetHeight(frame)
hjapi.DzFrameGetHeroBarButton(buttonId)
hjapi.DzFrameGetHeroHPBar(buttonId)
hjapi.DzFrameGetHeroManaBar(buttonId)
hjapi.DzFrameGetItemBarButton(buttonId)
hjapi.DzFrameGetMinimap()
hjapi.DzFrameGetMinimapButton(buttonId)
hjapi.DzFrameGetName(frame)
hjapi.DzFrameGetParent(frame)
hjapi.DzFrameGetPortrait()
hjapi.DzFrameGetText(frame)
hjapi.DzFrameGetTextSizeLimit(frame)
hjapi.DzFrameGetTooltip()
hjapi.DzFrameGetTopMessage()
hjapi.DzFrameGetUnitMessage()
hjapi.DzFrameGetUpperBarButton(buttonId)
hjapi.DzFrameGetValue(frame)
hjapi.DzFrameHideInterface()
hjapi.DzFrameSetAbsolutePoint(frame, point, x, y)
hjapi.DzFrameSetAllPoints(frame, relativeFrame)
hjapi.DzFrameSetAlpha(frame, alpha)
hjapi.DzFrameSetAnimate(frame, animId, autoCast)
hjapi.DzFrameSetAnimateOffset(frame, offset)
hjapi.DzFrameSetEnable(frame, enable)
hjapi.DzFrameSetFocus(frame, enable)
hjapi.DzFrameSetFont(frame, fileName, height, flag)
hjapi.DzFrameSetMinMaxValue(frame, minValue, maxValue)
hjapi.DzFrameSetModel(frame, modelFile, modelType, flag)
hjapi.DzFrameSetParent(frame, parent)
hjapi.DzFrameSetPoint(frame, point, relativeFrame, relativePoint, x, y)
hjapi.DzFrameSetPriority(frame, priority)
hjapi.DzFrameSetScale(frame, scale)
hjapi.DzFrameSetScript(frame, eventId, funcName, sync)
hjapi.DzFrameSetScriptByCode(frame, eventId, funcHandle, sync)
hjapi.DzFrameSetSize(frame, w, h)
hjapi.DzFrameSetStepValue(frame, step)
hjapi.DzFrameSetText(frame, text)
hjapi.DzFrameSetTextAlignment(frame, align)
hjapi.DzFrameSetTextColor(frame, color)
hjapi.DzFrameSetTextSizeLimit(frame, limit)
hjapi.DzFrameSetTexture(frame, texture, flag)
hjapi.DzFrameSetTooltip(frame, tooltip)
hjapi.DzFrameSetUpdateCallback(funcName)
hjapi.DzFrameSetUpdateCallbackByCode(funcHandle)
hjapi.DzFrameSetValue(frame, value)
hjapi.DzFrameSetVertexColor(frame, vertexColor)
```

```
hjapi.DzFrameShow(frame, enable)
hjapi.DzGetClientHeight(...)
hjapi.DzGetClientWidth(...)
hjapi.DzGetColor(r, g, b, a)
hjapi.DzGetConvertWorldPositionX(...)
hjapi.DzGetConvertWorldPositionY(...)
hjapi.DzGetGameMode(...)
hjapi.DzGetGameUI()
hjapi.DzGetLocale()
hjapi.DzGetMouseFocus()
hjapi.DzGetMouseTerrainX()
hjapi.DzGetMouseTerrainY()
hjapi.DzGetMouseTerrainZ()
hjapi.DzGetMouseX()
hjapi.DzGetMouseXRelative()
hjapi.DzGetMouseY()
hjapi.DzGetMouseYRelative()
hjapi.DzGetPlayerInitGold(...)
hjapi.DzGetPlayerName(...)
hjapi.DzGetPlayerSelectedHero(...)
hjapi.DzGetTriggerKey()
hjapi.DzGetTriggerKeyPlayer()
hjapi.DzGetTriggerSyncData()
hjapi.DzGetTriggerSyncPlayer()
hjapi.DzGetTriggerUIEventFrame()
hjapi.DzGetTriggerUIEventPlayer()
hjapi.DzGetUnitNeededXP(whichUnit, level)
hjapi.DzGetUnitUnderMouse()
hjapi.DzGetWheelDelta()
hjapi.DzGetWindowHeight()
hjapi.DzGetWindowWidth()
hjapi.DzGetWindowX()
hjapi.DzGetWindowY()
hjapi.DzIsKeyDown(iKey)
hjapi.DzIsMouseOverUI()
hjapi.DzIsWindowActive()
hjapi.DzLoadToc(tocFilePath)
hjapi.DzOriginalUIAutoResetPoint(enable)
hjapi.DzPlatform_GameStart(...)
hjapi.DzPlatform_HasGameOver(...)
hjapi.DzPlatform_HasGameOver_Player(...)
hjapi.DzSetCustomFovFix(value)
hjapi.DzSetMemory(address, value)
hjapi.DzSetMousePos(x, y)
hjapi.DzSetParams(...)
hjapi.DzSetUnitID(whichUnit, id)
hjapi.DzSetUnitModel(whichUnit, model)
hjapi.DzSetUnitPosition(whichUnit, x, y)
hjapi.DzSetUnitTexture(whichUnit, path, texId)
hjapi.DzSetWar3MapMap(blp)
hjapi.DzSimpleFontStringFindByName(name, id)
hjapi.DzSimpleFrameFindByName(name, id)
hjapi.DzSimpleTextureFindByName(name, id)
hjapi.DzSyncBuffer(...)
```

```
hjapi.DzSyncData(prefix, data)
hjapi.DzTriggerRegisterKeyEvent(trig, key, status, sync, funcName)
hjapi.DzTriggerRegisterKeyEventByCode(trig, key, status, sync, funcHandle)
hjapi.DzTriggerRegisterMouseEvent(trig, btn, status, sync, funcName)
hjapi.DzTriggerRegisterMouseEventByCode(trig, btn, status, sync, funcHandle)
hjapi.DzTriggerRegisterMouseMoveEvent(trig, sync, funcName)
hjapi.DzTriggerRegisterMouseMoveEventByCode(trig, sync, funcHandle)
hjapi.DzTriggerRegisterMouseWheelEvent(trig, sync, funcName)
hjapi.DzTriggerRegisterMouseWheelEventByCode(trig, sync, funcHandle)
hjapi.DzTriggerRegisterSyncData(trig, prefix, server)
hjapi.DzTriggerRegisterWindowResizeEvent(trig, sync, funcName)
hjapi.DzTriggerRegisterWindowResizeEventByCode(trig, sync, funcHandle)
hjapi.DzUnitDisableAttack(...)
hjapi.DzUnitDisableInventory(...)
hjapi.DzUnitLearningSkill(...)
hjapi.DzUnitSilence(...)
hjapi.EXBlendButtonIcon(...)
hjapi.EXDclareButtonIcon(...)
hjapi.EXDisplayChat(...)
hjapi.EXEffectMatReset(effect)
hjapi.EXEffectMatRotateX(effect, angle)
hjapi.EXEffectMatRotateY(effect, angle)
hjapi.EXEffectMatRotateZ(effect, angle)
hjapi.EXEffectMatScale(effect, x, y, z)
hjapi.EXExecuteScript(script)
hjapi.EXGetAbilityDataInteger(abil, level, dataType)
hjapi.EXGetAbilityDataReal(abil, level, dataType)
hjapi.EXGetAbilityDataString(abil, level, dataType)
hjapi.EXGetAbilityId(abil)
hjapi.EXGetAbilityState(abil, stateType)
hjapi.EXGetAbilityString(...)
hjapi.EXGetBuffDataString(buffCode, dataType)
hjapi.EXGetEffectSize(effect)
hjapi.EXGetEffectX(effect)
hjapi.EXGetEffectY(effect)
hjapi.EXGetEffectZ(effect)
hjapi.EXGetEventDataDamageData(eddType)
hjapi.isEventPhysicalDamage()
hjapi.isEventAttackDamage()
hjapi.isEventRangedDamage()
hjapi.isEventDamageType(damageType)
hjapi.isEventWeaponType(weaponType)
hjapi.isEventAttackType(attackType)
hjapi.EXGetItemDataString(itemCode, dataType)
hjapi.EXGetUnitAbility(whichUnit, abilityID)
hjapi.EXGetUnitAbilityByIndex(whichUnit, index)
hjapi.EXGetUnitArrayString(...)
hjapi.EXGetUnitInteger(...)
hjapi.EXGetUnitReal(...)
hjapi.EXGetUnitString(...)
hjapi.EXPauseUnit(whichUnit, enable)
hjapi.UnitAddSwim(whichUnit)
hjapi.UnitRemoveSwim(whichUnit)
hjapi.EXSetAbilityAEmeDataA(...)
```

```

hjapi.EXSetAbilityDataInteger(...)
hjapi.EXSetAbilityDataReal(...)
hjapi.EXSetAbilityDataString(...)
hjapi.EXSetAbilityState(ability, stateType, value)
hjapi.EXSetAbilityString(...)
hjapi.EXSetBuffDataString(buffCode, dataType, value)
hjapi.EXSetEffectSize(e, size)
hjapi.EXSetEffectSpeed(e, speed)
hjapi.EXSetEffectXY(e, x, y)
hjapi.EXSetEffectZ(e, z)
hjapi.EXSetEventDamage(amount)
hjapi.EXSetItemDataString(itemCode, dataType, value)
hjapi.EXSetUnitArrayString(...)
hjapi.EXSetUnitCollisionType(enable, u, t)
hjapi.EXSetUnitFacing(u, angle)
hjapi.EXSetUnitInteger(...)
hjapi.EXSetUnitMoveType(u, t)
hjapi.EXSetUnitReal(...)
hjapi.EXSetUnitString(...)
hjapi.GetEventDamage()
hjapi.GetUnitState(whichUnit, state)
hjapi.RequestExtraBooleanData(dataType, whichPlayer, param1, param2, param3,
param4, param5, param6)
hjapi.RequestExtraIntegerData(dataType, whichPlayer, param1, param2, param3,
param4, param5, param6)
hjapi.RequestExtraRealData(dataType, whichPlayer, param1, param2, param3, param4,
param5, param6)
hjapi.RequestExtraStringData(dataType, whichPlayer, param1, param2, param3,
param4, param5, param6)
hjapi.SetUnitState(whichUnit, state, value)

```

- [L]**lib\leaderBoard.lua**

```

hleaderBoard.LeaderboardResize(whichLeaderBoard)
hleaderBoard.create(key, title, refreshFrequency, response)
hleaderBoard.show(key)
hleaderBoard.hide()
hleaderBoard.setTitle(key, title)
hleaderBoard.pos(key, n)
hleaderBoard.top(key)
hleaderBoard.bottom(key)

```

- [L]**lib\lightning.lua**

```

hlightning.del(lightning, delay)
hlightning.xyz2xyz(lightningType, x1, y1, z1, x2, y2, z2, during)
hlightning.loc2loc(lightningType, loc1, loc2, during)
hlightning.unit2unit(lightningType, unit1, unit2, during)

```

- [L]lib\monitor.lua

```
hmonitor.create(key, frequency, action, ignoreFilter)
hmonitor.destroy(key)
hmonitor.isListening(key, obj)
hmonitor.listen(key, obj)
hmonitor.ignore(key, obj)
```

- [L]lib\multiBoard.lua

```
hmultiBoard.create(key, refreshFrequency, yourData)
hmultiBoard.setTitle(whichBoard, title)
```

- [L]lib\player.lua

```
hplayer.adjustPlayerState(delta, whichPlayer, whichPlayerState)
hplayer.setPlayerState(whichPlayer, whichPlayerState, value)
hplayer.forEach(action)
hplayer.index(whichPlayer)
hplayer.loc()
hplayer.setConvertRatio(ratio)
hplayer.getConvertRatio()
hplayer.getName(whichPlayer)
hplayer.setName(whichPlayer, name)
hplayer.getSelection(whichPlayer)
hplayer.getStatus(whichPlayer)
hplayer.setStatus(whichPlayer, status)
hplayer.getPrestige(whichPlayer)
hplayer.setPrestige(whichPlayer, prestige)
hplayer.getApm(whichPlayer)
hplayer.setIsolated(whichPlayer, isIsolated)
hplayer.isIsolated(whichPlayer)
hplayer.getRandomHero()
hplayer.hideUnit(whichPlayer)
hplayer.clearUnit(whichPlayer)
hplayer.defeat(whichPlayer, tips)
hplayer.victory(whichPlayer, tips)
hplayer.setIsAutoConvert(whichPlayer, b)
hplayer.getIsAutoConvert(whichPlayer)
hplayer.setIsShocking(whichPlayer, b)
hplayer.getIsShocking(whichPlayer)
hplayer.getDamage(whichPlayer)
hplayer.addDamage(whichPlayer, val)
hplayer.getBeDamage(whichPlayer)
hplayer.addBeDamage(whichPlayer, val)
hplayer.getKill(whichPlayer)
hplayer.addKill(whichPlayer, val)
hplayer.diffGoldRatio(whichPlayer, diff, during)
```

```
hplayer.setGoldRatio(whichPlayer, val, during)
hplayer.addGoldRatio(whichPlayer, val, during)
hplayer.subGoldRatio(whichPlayer, val, during)
hplayer.getGoldRatio(whichPlayer)
hplayer.diffLumberRatio(whichPlayer, diff, during)
hplayer.setLumberRatio(whichPlayer, val, during)
hplayer.addLumberRatio(whichPlayer, val, during)
hplayer.subLumberRatio(whichPlayer, val, during)
hplayer.getLumberRatio(whichPlayer)
hplayer.diffExpRatio(whichPlayer, diff, during)
hplayer.setExpRatio(whichPlayer, val, during)
hplayer.addExpRatio(whichPlayer, val, during)
hplayer.subExpRatio(whichPlayer, val, during)
hplayer.getExpRatio(whichPlayer)
hplayer.diffSellRatio(whichPlayer, diff, during)
hplayer.setSellRatio(whichPlayer, val, during)
hplayer.addSellRatio(whichPlayer, val, during)
hplayer.subSellRatio(whichPlayer, val, during)
hplayer.getSellRatio(whichPlayer)
hplayer.getTotalGold(whichPlayer)
hplayer.addTotalGold(whichPlayer, val)
hplayer.getTotalGoldCost(whichPlayer)
hplayer.addTotalGoldCost(whichPlayer, val)
hplayer.getTotalLumber(whichPlayer)
hplayer.addTotalLumber(whichPlayer, val)
hplayer.getTotalLumberCost(whichPlayer)
hplayer.addTotalLumberCost(whichPlayer, val)
hplayer.adjustGold(whichPlayer)
hplayer.adjustLumber(whichPlayer)
hplayer.getGold(whichPlayer)
hplayer.setGold(whichPlayer, gold, u)
hplayer.addGold(whichPlayer, gold, u)
hplayer.subGold(whichPlayer, gold, u)
hplayer.getLumber(whichPlayer)
hplayer.setLumber(whichPlayer, lumber, u)
hplayer.addLumber(whichPlayer, lumber, u)
hplayer.subLumber(whichPlayer, lumber, u)
hplayer.getFoodUsed(whichPlayer)
hplayer.setFoodUsed(whichPlayer, value)
hplayer.getFoodCap(whichPlayer)
hplayer.setFoodCap(whichPlayer, value)
hplayer.getFoodCapCeiling(whichPlayer)
hplayer.setFoodCapCeiling(whichPlayer, value)
```

- [L]lib\quest.lua

```
hquest.del(q, delay)
hquest.create(options)
hquest.flash()
hquest.setCompleted(q)
hquest.setFailed(q)
hquest.setDiscovered(q)
```

- [L]**lib\rect.lua**

```
hrect.alloc(r, name)
hrect.world()
hrect.camera()
hrect.playable()
hrect.create(x, y, w, h, name)
hrect.getName(whichRect)
hrect.getX(whichRect)
hrect.getY(whichRect)
hrect.getWidth(whichRect)
hrect.getHeight(whichRect)
hrect.getMinX(whichRect)
hrect.getMinY(whichRect)
hrect.getMaxX(whichRect)
hrect.getMaxY(whichRect)
hrect.del(whichRect, delay)
hrect.lock(options)
```

- [L]**lib\shop.lua**

- [L]**lib\skill\basic.lua**

```
hskill.avoid(whichUnit)
hskill.invulnerable(whichUnit, during, effect)
hskill.invulnerableRange(x, y, radius, filter, during, effect)
hskill.pause(whichUnit, during, pauseColor)
hskill.invisible(whichUnit, during, transition, effect)
hskill.visible(whichUnit, during, transition, effect)
hskill.modelEffect(whichUnit, whichAbility, abilityLevel, during)
hskill.diy(options)
```

- [L]**lib\skill\complex.lua**

```
hskill.knocking(options)
hskill.split(options)
hskill.broken(options)
hskill.swim(options)
hskill.silent(options)
hskill.unarm(options)
hskill.fetter(options)
hskill.bomb(options)
hskill.lightningChain(options)
```

```
hskill.crackFly(options)
hskill.rangeSwim(options)
hskill.whirlwind(options)
hskill.leap(options)
hskill.leapPaw(options)
hskill.leapRange(options)
hskill.leapReflex(options)
hskill.rectangleStrike(options)
```

- [L]**lib\skill\damage.lua**

```
local _damageTtg(targetUnit, damage, fix, rgb, speed)
hskill.damage(options)
hskill.damageStep(options)
hskill.damageRange(options)
hskill.damageGroup(options)
```

- [L]**lib\skill\ring.lua**

```
hring.check(id)
hring.insert(whichUnit, id, level)
hring.remove(whichUnit, id)
```

- [L]**lib\skill.lua**

```
hskill.getAttribute(abilityId)
hskill.addProperty(whichUnit, abilityId, level)
hskill.subProperty(whichUnit, abilityId, level)
hskill.getName(abilityId)
hskill.add(whichUnit, abilityId, level, during)
hskill.set(whichUnit, abilityId, level, during)
hskill.del(whichUnit, abilityId, delay)
hskill.forever(whichUnit, abilityId)
hskill.has(whichUnit, abilityId)
hskill.setCoolDown(whichUnit, abilityID, coolDown)
```

- [L]**lib\slk.lua**

```
hslk.i2v(id, ...)
hslk.n2v(name)
hslk.n2i(name)
hslk.classIds(class)
hslk.typeIds(t)
hslk.misc(...)
```

- [L]lib\sound.lua

```
hsound.voiceCreate(path, duration, is3D, channel, volume, pitch)
hsound.voice(s)
hsound.voice2Player(s, whichPlayer)
hsound.voice2Unit(s, volumePercent, u)
hsound.voice2XYZ(s, x, y, z)
hsound.voice2Rect(s, whichRect, during)
hsound.bgmStop(whichPlayer)
hsound.bgm(musicFileName, whichPlayer)
hsound.bgmVolume(percent, whichPlayer)
```

- [L]lib\sync.lua

```
hsync.key()
hsync.mix(key, array)
hsync.call(key, callback, array)
hsync.init()
hsync.send(key, array)
hsync.onSend(key, callback)
```

- [L]lib\textTag.lua

```
htextTag.del(ttg, delay)
htextTag.create(msg, size, color, opacity, during)
htextTag.create2XY(x, y, msg, size, color, opacity, during, zOffset)
htextTag.create2Unit(u, msg, size, color, opacity, during, zOffset)
htextTag.createFollowUnit(u, msg, size, color, opacity, during, zOffset)
htextTag.getSize(ttg)
htextTag.getColor(ttg)
htextTag.getMsg(ttg)
htextTag.getOpacity(ttg)
htextTag.getDuring(ttg)
htextTag.setVelocity(ttg, xSpeed, ySpeed)
htextTag.style(ttg, showType, xSpeed, ySpeed)
htextTag.model(options)
```

- [L]lib\texture.lua

```
htexture.cinematicFilterGeneric(duration, bmode, tex, red0, green0, blue0, trans0,
red1, green1, blue1, trans1)
htexture.mark(path, during, whichPlayer, red, green, blue)
htexture.alertCircle(diameter, x, y, during, token, color)
```

- [L]lib\time.lua

```
htime.clock()
htime.his()
htime.timerInPool()
htime.timerInKernel(period, yourFunc, isInterval)
htime.kernelInfo(t)
htime.getSetTime(t)
htime.getRemainTime(t)
htime.getElapsedTime(t)
htime.pause(t)
htime.resume(t)
htime.delTimer(t)
htime.setTimeout(frequency, yourFunc, title)
htime.setInterval(frequency, yourFunc, title)
```

- [L]lib\unit.lua

```
hunit.getAvatar(uOrId)
hunit.getSight(uOrId)
hunit.getNSight(uOrId)
hunit.getAttackSides1(uOrId)
hunit.getMaxLife(u)
hunit.getCurLife(u)
hunit.setCurLife(u, val)
hunit.addCurLife(u, val)
hunit.subCurLife(u, val)
hunit.getMaxMana(u)
hunit.getCurMana(u)
hunit.setCurMana(u, val)
hunit.addCurMana(u, val)
hunit.subCurMana(u, val)
hunit.getCurLifePercent(u)
hunit.setCurLifePercent(u, val)
hunit.getCurManaPercent(u)
hunit.setCurManaPercent(u, val)
hunit.addExp(u, val, showEffect)
hunit.setPeriod(u, life)
hunit.getPeriod(u)
hunit.getPeriodRemain(u)
hunit.setFacing(u, facing)
hunit.getFacing(u)
hunit.show(u)
hunit.hide(u)
hunit.pause(u)
hunit.resume(u)
hunit.x(u)
hunit.y(u)
hunit.z(u)
hunit.enablePunish(u)
hunit.disablePunish(u)
hunit.setInvulnerable(u, flag)
```

```
hunit.setAnimateSpeed(whichUnit, speed, during)
hunit.setRGBA(whichUnit, red, green, blue, opacity, during)
hunit.delRGBA(whichUnit, buffKey)
hunit.resetRGBA(whichUnit)
hunit.getOwner(whichUnit)
hunit.portal(whichUnit, x, y, facing)
hunit.animate(whichUnit, animate)
hunit.embed(u, options)
hunit.create(options)
hunit.getId(uOrId)
hunit.getName(u)
hunit.getUserData(u)
hunit.setUserData(u, val, during)
hunit.setColor(u, color)
hunit.del(targetUnit, delay)
hunit.kill(targetUnit, delay)
hunit.exploded(targetUnit, delay)
hunit.setCanFly(u)
hunit.getFlyHeight(u)
hunit.setFlyHeight(u, height, speed)
```

- [L]lib\weather.lua

```
hweather.del(w, delay)
hweather.create(options)
```

- [L]slk\pilot.lua

```
_ability(_v)
_item(_v)
_unit(_v)
_buff(_v)
_upgrade(_v)
_attr(_v)
_xtras(_v)
_ring(_v)
_onSkillStudy(_v)
_onSkillEffect(_v)
_onItemGet(_v)
_onItemUsed(_v)
_onRing(_v)
```

- [L]slk\setter.lua

```
F6V_I_SYNTHESIS(formula)
local F6_RING(_v)
F6V_A(_v)
```

```
F6V_U(_v)
F6V_COURIER_SKILL()
F6V_I_CD(_v)
F6V_I_SHADOW(_v)
F6V_I(_v)
F6V_B(_v)
F6V_UP(_v)
```

- [L]slk\slk.lua

```
hslk_init()
local hslk_cli_set(_v)
hslk_conf(conf)
hslk_ability(_v)
hslk_ability_empty(_v)
hslk_ability_ring(_v)
hslk_unit(_v)
hslk_hero(_v)
hslk_courier(_v)
hslk_item_synthesis(formula)
hslk_item(_v)
hslk_item_fleeting(Name, file, modelScale, moveHeight)
hslk_buff(_v)
hslk_upgrade(_v)
```

- [B]slk\system.lua

属性系统

h-lua 有强大的属性系统

能轻松几行代码就实现绝大多数属性的实时改动

hattribute.setRelation

属性每1单位映射 可以模拟三围加成等

gold_ratio / lumber_ratio / exp_ratio / sell_ratio

黄金率 / 木头率 / 经验率 / 售卖率 (%)

- 影响资源获取率 售卖率以默认的50%为准，假如某玩家作者设置TA的售卖率不等于50，那么会自动算出差值，给予扣款或补偿

life / life_back

生命 / 生命恢复

- 单位的最大生命值 / 每秒对生命恢复的量 最小值 1

mana / mana_back

魔法 / 魔法恢复

- 单位的最大魔法值 / 每秒对魔法恢复的量 最小值 1

move

移动力

- [范围] 0 ~ ?取决于地图的平衡常数设置(一般设为522)
每秒移动最大距离
- 数值无上限，但地图移动上限为522
- 假设移动力为2000，则实际移动仍为522，而其他由移动力影响的技能属性将以2000点计算

defend / defend_white / defend_green

护甲 (总/白/绿)

- 你可以使用`hattr.set`改动`defend`, 实际改的是`defend_green`

damage_reduction / damage_decrease

固定值减伤 / 百分比减伤

- 可以直接抵消伤害
- 减伤最低为0，也就是不能反增伤

sight / sight_day /sight_night

视野(变化) / 白天视野 / 黑夜视野

- 单位的视野`sight`指的是变化值，`attr`设定=时不是终点值而是“变化”的目标值
- 你可以使用`hattr.set`设置`sight_day` /`sight_night`, 但并不能真的设置分别的视野，实际改动的变化值

avoid

回避 (%)

- 当英雄受到的是“攻击”造成的伤害时会有一定几率闪避抵抗，并且使攻击特效失效
- 闪避效果只有当英雄回避属性大于0%时才会开始生效 回避可以无限堆，回避率达到100%时一般情况下
回避所有伤害 回避率=(回避-命中) (%)
- 如果该次伤害为无视回避，则回避失效 (`breakArmorType`)

aim

命中 (%)

- `aim` 命中直接抵消回避生效的效果值 回避率=(回避-命中) (%)

punish / punish_current

僵直 / 当前硬直

- 僵直的单位移动攻速会被限制 硬直扣减量与受到伤害相关
- 受到快速多段伤害或高伤害时，将很容易被击破而僵直 硬直被破坏时会僵直5秒，并重置硬直条，期间受到伤害可能会再次僵直
- 僵直的僵直5秒的体现是攻击速度和移动速度被极大地减少 硬直未被击破时每隔5秒恢复100点

punish_oppose

僵直抵抗 (%)

- 不造成伤害时，不减少硬直
- 硬直抵抗按百分比减少硬直的减速效果(减的是攻速及移速)
- 硬直抵抗的效果最高只能减少90% 力量 (总/白/绿)

str / str_white / str_green

力量 (总/白/绿)

- 你可以使用hattr.set设置str,但实际改动的是str_green

agi / agi_white / agi_green

敏捷 (总/白/绿)

- 你可以使用hattr.set设置str,但实际改动的是agi_green

int / int_white / int_green

智力 (总/白/绿)

- 你可以使用hattr.set设置int,但实际改动的是int_green

attack / attack_white / attack_green

攻击

- 一个单位的当前攻击浮动+白+绿
- 由于存在区间浮动，所以这个值每次get都会不一样
- 你可以使用hattr.set设置attack,但实际改动的是attack_green
- 在h-lua下所有单位造成的伤害都视为攻击attack (包括原生技能)
- 在调用攻击附魔后，攻击可以附带元素 单位进行攻击的能力，一般分为近战或者远程 攻击间隔

attack_space

攻击间隔

- 根据原始攻速和攻击速度计算出的间隔值
- 但是set的时候实际设置的是原始间隔(attack_space_origin)

attack_range

攻击范围

- 单位进行攻击的最大距离

attack_speed

攻击速度 (%)

- 攻击一次的间隔 假设当前攻速增益为 x% : 攻击间隔公式: (基本间隔*100) / (100+x) [单位: 击/秒]

reborn

复活时间 (秒)

- 此值默认-999, 无复活效果 当这个值大于等于0时, 且单位是英雄, 单位会拥有复活的效果 复活的流程执行是一个默认函数, 有需要可修改 hevent_default_actions.hero.reborn

cure

治疗效果 (%)

- 增强治疗的效果 影响hunit.addLife/Mana的效果 life_back/mana_back已自动生效

hemophagia / hemophagia_skill

吸血 (%) / 技能吸血 (%)

- 攻击 / 技能造成伤害时触发吸血 以实际伤害及吸血百分比为计算

damage_amplitude

伤害增幅 (%)

- 增幅属性可以提高所有的伤害 (包括真实伤害)
- 如增幅100%, 则攻击、物品、技能等所有伤害都会提高100% 伤害增幅在暴击分裂等之前 负重 (Kg)

weight / weight_current

负重 (Kg)

- 负重决定你身上能够佩戴多少重量的物品装备
- 常规地图物品都设为0kg, 所以负重是无效的

invincible

无敌 (%)

- 收到伤害时瞬间无敌抵消的几率

damage_rebound / damage_rebound_oppose

伤害反射 (%) / 反伤抵抗 (%)

- 可反射受到的一切伤害
- 反伤数值机制精度为0.01, 小于0.01无效 默认最低为0% (无反射) 最高无限定

- 以最终伤害值计算反射，反射的伤害不会再次成为反射源

knocking_odds / knocking_extent / knocking_oppose

暴击几率 (%) / 暴击伤害加成 (%) / 暴击抵抗 (%)

- 这是单位本体的暴击，与xtras的无关，可线性叠加
- 本体暴击自动无视回避（模拟原生魔兽效果）
- 暴击抵抗阻碍物理暴击发生，包括本体的和xtras

split_oppose

分裂抵抗 (%)

- 阻碍分裂发生

hemophagia_oppose / hemophagia_skill_oppose

吸血抵抗 (%) / 技能吸血抵抗 (%)

- 抵抗攻击吸血 / 技能吸血效果

punish_oppose

僵直抵抗 (%)

- 抵抗僵直效果

split_oppose

分裂抵抗 (%)

- 阻碍分裂发生

swim_oppose

眩晕抵抗 (%)

- 眩晕抵抗同时减少受到眩晕的几率及眩晕时间 眩晕几率=原眩晕几率%-眩晕抵抗 (%) ,最小0%即无效
- 眩晕时间=原眩晕时间*(100%-眩晕抵抗%) (s) ,最小0秒即无效 沉默抵抗 (%)

silent_oppose

沉默抵抗 (%)

- 阻碍沉默发生

unarm_oppose

缴械抵抗 (%)

- 阻碍缴械发生

fetter_oppose

定身抵抗 (%)

- 阻碍定身发生

bomb_oppose

爆破抵抗 (%)

- 阻碍爆破发生

lightning_chain_oppose

闪电链抵抗 (%)

- 阻碍闪电链发生

crack_fly_oppose

击飞抵抗 (%)

- 阻碍击飞发生

buff_oppose

强化阻碍 (%)

- 当xtras使用attr增强单位时，效果会被此值减轻

debuff_oppose

负面抵抗 (%)

- 当xtras使用attr削弱单位时，效果会被此值减轻

BREAK_ARMOR_TYPE

无视装甲

- 详情请看const
- 无视护甲 defend
- 无视回避 avoid
- 无视无敌 invincible
- 无视附魔 enchant
- 无视只对正数值有效 额外特效

xtras

以事件作驱动，引发额外限定效果的操作

- 由于非常复杂，请查看attr的xtras源码段注释说明 常见例子：

```

{ on = CONST_EVENT.attack, action = "triggerUnit.attr.attack_speed", odds = 20.0,
val = 1.5, during = 3.0, effect = nil },
{ on = CONST_EVENT.attack, action = "attackUnit.attr.attack_speed", odds = 20.0,
val = 1.5, during = 3.0, effect = nil },
{ on = CONST_EVENT.skill, action = "castUnit.attr.attack_green", odds = 20.0, val
= 2, during = 3.0, effect = nil },
{ on = CONST_EVENT.item, action = "useUnit.attr.int_white", odds = 20.0, val = 2,
during = 3.0, effect = nil },
{ on = CONST_EVENT.attack, action = "targetUnit.spec.knocking", odds = 100,
percent = 100, effect = nil },
{ on = CONST_EVENT.skill, action = "targetUnit.spec.knocking", odds = 100, percent
= 100, effect = nil },
{ on = CONST_EVENT.attack, action = "targetUnit.spec.split", odds = 100, percent =
{30,50}, radius = 250 },
{ on = CONST_EVENT.attack, action = "targetUnit.spec.swim", odds = 0.0, val = 0.0,
during = 0.0, effect = nil },
{ on = CONST_EVENT.attack, action = "targetUnit.spec.broken", odds = 0.0, val =
0.0, effect = nil },
{ on = CONST_EVENT.attack, action = "targetUnit.spec.silent", odds = 0.0, val =
0.0, during = 0.0, effect = nil },
{ on = CONST_EVENT.attack, action = "targetUnit.spec.unarm", odds = 0.0, val = 0.0,
during = 0.0, effect = nil },
{ on = CONST_EVENT.attack, action = "targetUnit.spec.fetter", odds = 0.0, val =
0.0, during = 0.0, effect = nil },
{ on = CONST_EVENT.attack, action = "targetUnit.spec.bomb", odds = 0.0, radius =
0.0, val = 0.0, effect = nil },
{ on = CONST_EVENT.damage, action = "targetUnit.spec.lightning_chain", odds = 50,
val = 100, qty = 0, rate = 0.0 },
{ on = CONST_EVENT.beDamage, action = "sourceUnit.spec.crack_fly", odds = 50, val
= 100, distance = 300, height = 200, during = 0.5 },
{ on = CONST_EVENT.attack, action = "sourceUnit.spec.paw", odds = 50, val =
'damage', effect = 'Abilities\\Weapons\\GargoyleMissile\\GargoyleMissile.mdl',
radius = 50, odds = 50, speed = 10, deg = 30, distance = 500, qty = 3 }

```

enchant

附魔系统介绍

- e_开头的系列设定如：e_fire / e_fire_attack / e_fire_oppose 附魔，是框架内定义的一种概念 附魔带出元素集合，它们有：物理、魔法、火、焰、水、风、雷、木、毒等等很多种
附魔被伤害类型和attr引用，独立存在，仅呈现辅助式伤害
- 每个属性在造成对应的伤害时，会获得百分比加成，无上限，而抵抗则直接减少加成比例
- 下列例子不考虑其他因素：默认+10%参考henchantINTRINSIC_ADDITION 原伤100%，类型包含火。火加成为100%，火抵抗为40%，最终伤害170%（默认会+10%）
原伤100%，类型包含火。火加成为40%，火抵抗为40%，最终伤害110%（默认会+10%） 原伤100%，类型包含火。火加成为0%，火抵抗为40%，最终伤害70%（反而不如无类型了）
原伤100%，类型不包含火。火加成为40%，火抵抗为40%，最终伤害100%（类型无火，则火加成/抵抗皆无效） 伤害类型(伤害类型绑定附魔的元素)

DAMAGE_TYPE

伤害类型一般指伤害的类型，类型之间可以随意的重复叠加

- 详情请看const
- 例如一个伤害类型：可以是{physical,fire,fire}物理+火，而且火占到了三分之二的比例。一般情况下，一次伤害总是各种类型混合的，各个类型会按照各自的规则来计算削弱和加强，不必担心混乱
例如{magic,fire}这种类型，这次伤害是200点，他们会各自平分伤害：magic100, fire100，如果魔法就按魔抗计算被抵消，火伤害依然是有100点的伤害来源

DAMAGE_SRC

伤害方式共4种

- 详情请看const
- [未知方式]没有被定义的都会是未知来源，一种概念
- [攻击方式]普通攻击、魔兽原生技能造成的伤害
- [技能方式]技能设定（触发技能buff、技能debuff、技能特效）
- [物品方式]物品伤害，一种概念
- 伤害一般计算优先级：原生护甲伤害计算->方式类型判定->无视判定->附魔数值整合->暴击及抵抗->回避、命中->减伤->造成伤害->分裂、吸血、技能吸血->僵直、反伤->额外特效

JAPI

SDK马仔自带一些JAPI，实装函数一览

以代码提示函数存在作为依据，下列函数名仅供参考

```
EXSetEffectXY
DzIsKeyDown
EXGetUnitArrayString
DzFrameSetPoint
DzAPI_UnitType_SetEnum_PreventOrRequirePlace
EXSetUnitReal
EXSetUnitCollisionType
DzAPI_UnitType_GetUnitTypeDataAbilID
GetUnitState
DzAPI_UnitType_SetEnum_armor
SetUnitState
RequestExtraIntegerData
EXGetUnitReal
DzAPI_UnitType_GetUnitTypeDataBoolean
EXGetUnitString
EXSetUnitString
DzAPI_Map_SaveServerValue
EXGetUnitInteger
EXSetEffectSpeed
DzAPI_Map_IsBlueVIP
EXEffectMatReset
EXSetUnitInteger
DzGetWheelDelta
EXBlendButtonIcon
DzAPI_UnitType_CountUnitTypedataArrayTechID
```

```
EXSetItemDataString
DzAPI_UnitType_GetEnum_weapTp
EXSetUnitArrayString
EXGetEffectY
EXPauseUnit
EXGetAbilityDataString
EXSetUnitMoveType
EXSetUnitFacing
DzSetCustomFovFix
EXGetUnitAbility
EXExecuteScript
DzGetPlayerName
EXGetUnitAbilityByIndex
EXGetAbilityId
DzAPI_UnitType_SetUnitTypeDataArrayList
EXGetAbilityState
EXSetAbilityState
EXEffectMatRotateX
EXGetAbilityDataReal
EXSetAbilityDataReal
DzEvent_Tech_Finish
EXGetAbilityDataInteger
DzSimpleFontStringFindByName
EXSetAbilityDataInteger
DzAPI_UnitType_ResizeUnitTypeDataArrayListAbilID
EXSetAbilityString
DzAPI_UnitType_GetEnum_type
EXSetAbilityDataString
DzDotaInfo_Item_HE
DzAPI_UnitType_CountUnitTypeDataArrayListID
EXSetAbilityAEMeDataA
DzGetPlayerSelectedHero
DzAPI_UnitType_ResizeUnitTypeDataArrayListReal
EXGetBuffDataString
GetEventDamage
DzGetPlayerInitGold
EXSetBuffDataString
EXGetAbilityString
EXGetItemDataString
DzTriggerRegisterMouseEventByCode
DzDotaInfo_Item
EXGetEventDamageData
EXSetEventDamage
EXDisplayChat
EXSetEffectSize
EXDclareButtonIcon
EXSetEffectZ
DzSetParams
DzLoadToc
EXGetEffectX
DzAPI_UnitType_SetUnitTypeDataArrayListAbilID
EXGetEffectZ
RequestExtraBooleanData
EXGetEffectSize
```

EXEffectMatRotateY
DzFrameSetAnimateOffset
DzDotaInfo_IsPlayerRandom
EXEffectMatRotateZ
DzAPI_UnitType_GetUnitTypeDataRequiresamount
DzEvent_Building_Dead
DzAPI_UnitType_SetUnitTypeDataReal
DzAPI_Map_Ladder_SetPlayerStat
DzEvent_Unit_Hired
EXEffectMatScale
DzDebugString
DzUnitLearningSkill
DzAPI_UnitType_CountUnitTypeDataArrayAbilID
DzGetGameMode
DzDotaInfo_IsRepicked
DzDotaInfo_Item_TM
DzEvent_Building_Cancel
DzAPI_UnitType_SetEnum_typeModify
DzEvent_Building_Start
DzEvent_Building_Finish
DzAPI_UnitType_GetUnitTypeDataRequires
DzEvent_Hero_Dead
DzFrameGetText
DzAPI_UnitType_GetUnitTypeDataArrayString
DzEvent_Unit_Dead
DzEvent_Unit_Start
DzEvent_Unit_Cancel
DzAPI_UnitType_GetEnum_weapType
DzEvent_Unit_Finish
DzAPI_UnitType_GetUnitTypeDataReal
DzEvent_Tech_Start
DzGetClientHeight
DzEvent_Tech_Cancel
DzSimpleFrameFindByname
DzFrameGetPortrait
DzGetMouseTerrainX
DzEvent_Hero_Level
DzGetColor
DzEvent_Unit_ChangeOwner
DzEvent_Item_Drop
DzEvent_Item_Pickup
DzEvent_Item_Use
DzEvent_Item_Sell
DzGetY
DzPlatform_HasGameOver
DzPlatform_HasGameOver_Player
DzPlatform_GameStart
DzAPI_UnitType_GetEnum_movetp
DzAPI_Map_MissionComplete
DzAPI_Map_GetActivityData
DzAPI_Map_GetMapLevel
DzAPI_Map_GetServerValue
DzAPI_Map_Stat_SetStat
DzAPI_Map_Ladder_SetStat

```
DzAPI_Map_IsRPGLobby
DzAPI_Map_GetGameStartTime
DzAPI_Map_IsRPGLadder
DzFrameSetAlpha
DzAPI_Map_GetMatchType
DzAPI_Map_UpdatePlayerHero
DzFrameClearAllPoints
DzAPI_Map_GetLadderLevel
DzAPI_Map_IsRedVIP
DzAPI_Map_GetLadderRank
DzAPI_Map_GetMapLevelRank
DzAPI_Map_GetServerValueErrorCode
DzAPI_Map_GetGuildName
DzAPI_Map_GetGuildRole
DzAPI_UnitType_GetUnitTypeDataArrayUnitID
DzAPI_Map_GetMapConfig
DzAPI_UnitType_SetUnitTypeDataArrayReal
DzAPI_Map_HasMallItem
DzTriggerRegisterWindowResizeEventByCode
DzAPI_Map_ChangeStoreItemCount
DzFrameGetParent
DzAPI_Map_ChangeStoreItemCoolDown
DzAPI_Map_ToggleStore
DzAPI_Map_GetServerArchiveEquip
DzAPI_Map_GetServerArchiveDrop
DzFrameSetMinMaxValue
DzFrameSetScript
DzFrameGetMinimapButton
DzAPI_Map_OrpgTrigger
DzAPI_Map_GetUserID
DzAPI_Map_GetPlatformVIP
DzAPI_Map_SavePublicArchive
DzAPI_Map_GetPublicArchive
DzAPI_Map_UseConsumablesItem
DzAPI_Map_Statistics
RequestExtraStringData
DzFrameGetHeroManaBar
RequestExtraRealData
DzSyncData
DzAPI_UnitType_GetUnitTypeDataString
DzAPI_CommonFunc_SetARGBColorValue
DzAPI_UnitType_SetUnitTypeDataArrayUnitID
DzAPI_CommonFunc_SetARGBColorValuePercent
DzAPI_CommonFunc_GetARGBColorValue
DzUnitDisableInventory
DzAPI_CommonFunc_GetARGBColorValuePercent
DzFrameSetScale
DzAPI_UnitType_GetUnitTypeDataArrayReal
DzAPI_UnitstateToInteger
DzAPI_UnitType_GetUnitTypeDataRequirescount
DzAPI_UnitType_GetEnum_PreventOrRequirePlaceCheck
DzAPI_UnitType_SetUnitTypeDataInt
DzAPI_UnitType_SetUnitTypeDataInt
DzFrameCageMouse
```

```
DzFrameSetScriptByCode
DzAPI_UnitType_CountUnitTypeDataArrayReal
DzAPI_UnitType_SetUnitTypeDataBoolean
DzAPI_UnitType_CountUnitTypeDataArrayBoolean
DzAPI_UnitType_ResizeUnitTypeDataArrayBoolean
DzGetWindowHeight
DzAPI_UnitType_GetUnitTypeDataArrayBoolean
DzTriggerRegisterMouseWheelEventByCode
DzAPI_UnitType_SetUnitTypeDataArrayBoolean
DzAPI_UnitType_SetUnitTypeDataString
DzAPI_UnitType_CountUnitTypeDataArrayList
DzAPI_UnitType_ResizeUnitTypeDataArrayList
DzAPI_UnitType_ResizeUnitTypeDataTechID
DzAPI_UnitType_GetUnitTypeDataTechID
DzAPI_UnitType_SetUnitTypeDataTechID
DzAPI_UnitType_SetUnitTypeDataAbilID
DzAPI_UnitType_GetUnitTypeDataAbilID
DzAPI_UnitType_CountUnitTypeDataUnitID
DzAPI_UnitType_SetEnum_TargetTypeSeries
DzAPI_UnitType_SetEnum_atkType
DzAPI_UnitType_ResizeUnitTypeDataUnitID
DzTriggerRegisterMouseMoveEventByCode
DzAPI_UnitType_ResizeUnitTypeDataItemID
DzAPI_UnitType_GetUnitTypeDataItemID
DzAPI_UnitType_SetUnitTypeDataItemID
DzAPI_UnitType_GetEnum_regenType
DzAPI_UnitType_SetEnum_regenType
DzAPI_UnitType_SetEnum_race
DzAPI_UnitType_SetEnum_weapTp
DzAPI_UnitType_GetEnum_defType
DzAPI_UnitType_GetEnum_PreventOrRequirePlace
DzAPI_UnitType_SetEnum_defType
DzFrameGetTooltip
DzAPI_UnitType_GetEnum_Primary
DzAPI_UnitType_GetEnum_warpsOn
DzFrameSetTooltip
DzAPI_UnitType_SetEnum_warpsOn
DzAPI_UnitType_GetEnum_atkType
DzAPI_UnitType_SetEnum_weapType
DzFrameSetStepValue
DzAPI_UnitType_SetEnum_Primary
DzFrameSetTextColor
DzAPI_UnitType_SetEnum_movetp
DzAPI_UnitType_GetEnum_buffType
DzAPI_UnitType_SetEnum_buffType
DzGetTriggerUIEventPlayer
DzAPI_UnitType_SetEnum_race
DzAPI_UnitType_GetEnum_deathType
DzAPI_UnitType_SetEnum_deathType
DzAPI_UnitType_SetEnum_armor
DzAPI_UnitType_GetEnum_TargetTypeSeries
DzAPI_UnitType_GetEnum_TargetTypeCheck
DzAPI_UnitType_SetEnum_TargetTypeModify
DzAPI_UnitType_SetEnum_type
```

```
DzAPI_UnitType_GetEnum_typeCheck
DzAPI_UnitType_SetEnum_PreventOrRequirePlaceModify
DzAPI_UnitType_CountUnitTypeDataRequires
DzAPI_UnitType_ResizeUnitTypeDataRequires
DzGetConvertWorldPositionX
DzAPI_UnitType_SetUnitTypeDataRequires
DzSetUnitTexture
DzAPI_UnitType_CountUnitTypeDataRequiresamount
DzAPI_UnitType_ResizeUnitTypeDataRequiresamount
DzAPI_UnitType_SetUnitTypeDataRequiresamount
DzGetXRelative
DzGetY
DzFrameGetHeroBarButton
DzGetMouseTerrainZ
DzIsMouseOverUI
DzGetX
DzGetLocale
DzGetYRelative
DzSetMousePos
DzTriggerRegisterMouseEvent
DzTriggerRegisterKeyEvent
DzTriggerRegisterKeyEventByCode
DzTriggerRegisterMouseWheelEvent
DzOriginalUIAutoResetPoint
DzTriggerRegisterMouseMoveEvent
DzGetTriggerKey
DzGetTriggerKeyPlayer
DzGetWidth
DzGetClientWidth
DzGetX
DzGetY
DzTriggerRegisterWindowResizeEvent
DzIsWindowActive
DzTriggerRegisterSyncData
DzSyncBuffer
DzGetTriggerSyncData
DzGetTriggerSyncPlayer
DzFrameHideInterface
DzFrameEditBlackBorders
DzFrameGetMinimap
DzFrameGetCommandBarButton
DzFrameGetHeroHPBar
DzExecuteFunc
DzFrameGetItemBarButton
DzFrameGetUpperButtonBarButton
DzFrameGetChatMessage
DzFrameGetUnitMessage
DzFrameGetTopMessage
DzGetConvertWorldPositionY
DzFrameSetUpdateCallback
DzGetUnitNeededXP
DzFrameSetUpdateCallbackByCode
DzFrameShow
DzCreateFrame
```

```
DzCreateSimpleFrame  
DzDestroyFrame  
DzFrameSetAbsolutePoint  
DzFrameSetEnable  
DzFrameGetAlpha  
DzGetTriggerUIEventFrame  
DzFrameFindByName  
DzSimpleTextureFindByName  
DzGetGameUI  
DzClickFrame  
DzEnableWideScreen  
DzFrameSetText  
DzFrameSetTextSizeLimit  
DzFrameGetTextSizeLimit  
DzGetMouseFocus  
DzFrameSetPriority  
DzFrameSetAllPoints  
DzFrameSetFocus  
DzFrameSetModel  
DzFrameGetEnable  
DzFrameSetAnimate  
DzFrameSetTexture  
DzFrameGetValue  
DzFrameSetValue  
DzFrameSetSize  
DzCreateFrameByTagName  
DzFrameSetVertexColor  
DzFrameSetParent  
DzFrameGetHeight  
DzFrameSetFont  
DzFrameSetTextAlignment  
DzFrameGetName  
DzConvertWorldPosition  
DzGetUnitUnderMouse  
DzDestructablePosition  
DzSetUnitPosition  
DzSetMemory  
DzSetUnitModel  
DzSetUnitID  
DzSetWar3MapMap  
DzUnitSilence  
DzUnitDisableAttack
```

hslk 使用入门

hslk是h-lua的物编生成方式，只要在hslk目录中运用设定好的方法，即可创建物编

类型

```
unit(common/hero/courier) 单位  
item(common/synthesis) 物品
```

```
ability(common/empty/ring) 技能
buff(common) 魔法效果
upgrade(common) 科技
```

专属字段

改不了的字段定义了也没效

```
_id = "[改不了]自动生成的物编ID"
_id_force = "[一般不用]自定义物编ID"
_class = "[改不了]hslk数据类别"
_type = "[一般不用]hslk数据形式"
_parent = "模版物编ID"
_overlie = "叠加"
_weight = "重量"
_attr = "属性"
_ring = "光环"
_remarks = "结尾灰字描述"
_cooldown = "冷却时间"
_cooldownTarget = "冷却技能目标"
_shadow = "强制使用影子物品"
```

配置hslk

hslk的数据配置在【你的项目】的hslk目录中，即可测试自动构建并读取

支持默认程度的自定义配置，例如物编文本生成时的颜色等

默认内置一套配置，使用conf方法全覆盖即可

```
hslk_conf({
    courierSkill = {
        -- 信使技能-名称、热键、图标位置、冷却
        blink = {
            Ubertip = "闪烁到任何地方", Art =
            "ReplaceableTextures\\CommandButtons\\BTNBlink.blp",
            Hotkey = 'Q', Buttonpos_1 = 0, Buttonpos_2 = 2, Cool1 = 10
        },
        rangePickUp = {
            Ubertip = "将附近地上的物品拾取到身上", Art =
            "ReplaceableTextures\\CommandButtons\\BTNPickUpItem.blp",
            Hotkey = 'W', Buttonpos_1 = 1, Buttonpos_2 = 2, Cool1 = 5
        },
        separate = {
            Ubertip = "将合成或重叠的物品拆分成零件", Art =
            "ReplaceableTextures\\CommandButtons\\BTNRepair.blp",
            Hotkey = 'E', Buttonpos_1 = 2, Buttonpos_2 = 2, Cool1 = 5
        },
        deliver = {
```

```

        Ubertip = "将所有物品依照顺序传送给英雄，当你的英雄没有空余物品位置，物品会返回  

给信使", Art = "ReplaceableTextures\\CommandButtons\\BTNLoadPeson.blp",  

        Hotkey = 'R', Buttonpos_1 = 3, Buttonpos_2 = 2, Cool1 = 5  

    },  

},  

-- 描述文本颜色,可配置 hcolor 里拥有的颜色函数,也可以配置 hex 6位颜色码  

color = {  

    hotKey = "ffcc00", -- 热键  

    itemCoolDown = "ccffff", -- 物品冷却时间  

    itemAttr = "b0f26e", -- 物品属性  

    itemOverlie = "ff59ff", -- 物品叠加  

    itemWeight = "ee82ee", -- 物品重量  

    itemRemarks = "969696", -- 物品备注  

    itemFragment = hcolor.orange, -- 物品零部件  

    itemProfit = "ffd88c", -- 物品合成品  

    abilityCoolDown = "ccffff", -- 技能冷却时间  

    abilityAttr = "b0f26e", -- 技能属性  

    abilityRemarks = "969696", -- 技能备注  

    ringArea = "99ccff", -- 光环范围  

    ringTarget = "99ccff", -- 光环作用目标  

    heroWeapon = "ff3939", -- 英雄攻击武器类型  

    heroAttack = "ff8080", -- 英雄基础攻击  

    heroRange = "99ccff", -- 英雄攻击范围  

    heroPrimary = "ffff00", -- 英雄主属性  

    heroSecondary = "ffffcc", -- 英雄次属性  

    heroMove = "ccffcc", -- 英雄移动
},  

})

```

配置物品合成公式（一般完美支持hslk创建的注册物品）

必须提前配置公式，再调用hslk创建物品，否则物品说明文本不会自动协助合成补充

例子1 "小刀割大树=小刀+大树" 2个不一样的合1个
例子2 "三头地狱犬的神识=地狱狗头x3" 3个一样的合1个
例子3 "精灵神水x2=精灵的眼泪x50" 50个一样的合一种,但得到2个
例子4 {
 {"小刀割大树",1}, {"小刀",1}, {"大树",1}
}

对对象型配置第一项为结果物品(适合物品名称包含特殊字符的物品，如+/-影响公式的符号)

支持多种公式配置，当公式内存在+=这种破坏范式的名称时，可使用table进行设定

```

hslk_item_synthesis({  

    "璀璨精气球=精气球+璀璨",  

    "重剑=铁剑x2",  

    "刺客短剑=铁剑+闪避护符",  

    "淬火剑=重剑+淬火石",  

}

```

```

        {"小刀割大树", 1}, {"小刀", 1}, {"大树", 1}
    }
})

```

构建物编单位

```

hslk_unit({
    Name = "大步兵",
    abilList = "Avul,Apit,Aneu",
    file = "units\human\TheCaptain\TheCaptain",
    Art = "ReplaceableTextures\CommandButtons\BTNTTheCaptain.blp",
    pathTex = "PathTextures\4x4SimpleSolid.tga",
    modelScale = 1.10,
    scale = 1.10,
    HP = 100,
    spd = 0,
    sight = 800,
    nsight = 800,
    unitSound = "HighElfSwordsman",
    weapson = 0,
    race = "human",
    UberSplat = "HSMA",
})

```

构建物编物品

```

hslk_item({
    Name = "闪避护符",
    Art = "ReplaceableTextures\CommandButtons\BTNTalisman.blp",
    goldcost = 3500,
    lumbercost = 0,
    powerup = 0,
    sellable = 1,
    pawnable = 1,
    droppable = 1,
    Hotkey = "Q",
    _remarks = "戴上这个护符就能闪避，天下哪有那么神奇的事？！",
    _weight= 0.4,
    _attr = _attr({
        avoid = "+3",
        xtras= {
            {
                on = CONST_EVENT.attack
                action = "triggerUnit.attr.attack_speed",
                odds = 8,
                val = 10,
                during = 3,
                effect = "Abilities\Spells\NightElf\Starfall\StarfallTarget.mdl",
            },
        }
    })
})

```

```

        }
    }),
})
}
```

构建fake光环技能

```

hslk_ability_ring({
    Name = "月神强击",
    Hotkey = "E",
    race = "human",
    Art = "ReplaceableTextures\\PassiveButtons\\PASBTNTTrueShot.blp",
    _ring = _ring({
        effect = "Abilities\\Spells\\NightElf\\TrueshotAura\\TrueshotAura.mdl",
        effectTarget =
"Abilities\\Spells\\Other\\GeneralAuraTarget\\GeneralAuraTarget.mdl",
        radius = 600,
        target = "air,ground,friend,self,vuln,invu",
        attr = {
            attack_green = "+90",
            aim = "+20"
        }
    }),
})
}
```

物品也可以嵌套光环技能

```

hslk_item({
    Name = "铁剑",
    Art = "ReplaceableTextures\\CommandButtons\\BTNSSteelMelee.blp",
    goldcost = 800,
    lumbercost = 0,
    powerup = 0,
    sellable = 1,
    pawnable = 1,
    droppable = 1,
    Hotkey = "D",
    _remarks = "精铁铸造的大剑",
    _attr = {
        attack_green = "+33",
    },
    _ring = {
        effect = "Abilities\\Spells\\Orc\\CommandAura\\CommandAura.mdl",
        effectTarget = "Abilities\\Spells\\NightElf\\BattleRoar\\RoarTarget.mdl",
        attachTarget = "overhead",
        radius = 600,
        target = "air,ground,friend,self,vuln,invu",
        attr = {
            life = "+50",
        }
    }
})
```

```
    },
})
```

自定义物编模版

```
hslk_unit({
    _parent = "hpea"
})
```

自定义物编ID

```
hslk_unit({
    _id_force = "U001"
})
```

hslk 使用进阶

物品主动技能绑定

```
hslk_item({
    Name = "跳跳剑",
    _cooldown = 2,
    _cooldownTarget = CONST_ABILITY_TARGET.location.value,
    _onItemUsed = _onItemUsed(function(evtData)
        hunit.portal(evtData.triggerUnit, evtData.targetX, evtData.targetY)
    end),
})
```

技能主动绑定

```
hslk_ability({
    Name = "主动技能一个",
    _onSkillEffect = _onSkillEffect(function(evtData)
        -- 技能搞起来
    end)
})
```

多级技能配置

```
hslk_ability({
    Name = "多级的技能",
```

```
levels= 3,  
DataA = {0,0,0},  
DataB = {0,0,0},  
Cool = {0,0,0},  
Cost = {0,0,0},  
})
```

hslk属性的获取

hslk生成的物编，在游戏脚本运行时，可以超级方便地自由读取数据

```
hslk.i2v(id) --根据物编ID，获取全数据  
hslk.n2v(id) --根据物编Name，获取全数据，如果名字有重复，会返回多组数据  
hslk.n2i(id) --根据物编Name，获取物编ID，如果名字有重复，只返回第一个ID
```

hslk属性的获取slk数据

```
hslk.i2v(id, "slk") --slk数据放在第1级的slk的key里  
hslk.i2v(id, "_id") --hlua的自定义数据也在第1级里
```

注意，slk数据的值，一开始获取都是字符串，需要自己转换

hslk.i2v的多级获取写法

```
hslk.i2v(id, "slk", "Primary") --获取一个英雄的主属性串 STR|AGI|INT|_
```

无效路径，则返回nil

hslk的 _class 字段

```
hslk每组数据都必定会含有一个_class字段，是构建时用来记录数据类别的  
技能时 _class = ability  
单位时 _class = unit  
物品时 _class = item  
魔法效果时 _class = buff  
科技时 _class = upgrade  
在运行时，可以一次性获取所有某 class 下的ID字符串数组  
hslk.classIds({"ability", "item"})
```

hslk的 _type 字段

hs1k每组数据都必定会含有一个 `_type` 字段，是构建时用来记录数据类别的
`_type` 字段一般情况下都是 `common`
而引用hs1k官方的拓展函数，如`hs1k_ability_empty`,会有固定的搭配

`_type` 可以在定义时自定义值，你写什么就是什么，可用于后续脚本中的ID集合获取

在运行时，可以一次性获取所有某 `type` 下的ID字符串数组

```
hs1k.typeIds({"hero"})
```

hs1k的 Misc

hs1k里面可以获取misc数据,如:

```
hs1k.misc("Misc","FadeBuffMinDuration") --"10"
```

misc数据就是平衡常数界面等等的一些游戏框架式数据

如果你不清楚有什么值，可以参考 `h-lua/docs/slk-misc.md`

hs1k的 自定义数据

-- hs1k配置时，table里面可以像设置`_type`一样，定义你自己随心的值

```
hs1k_unit({
    _id_force = "xTYG",
    Name = "一个单位",
    _myData1 = 1,
    _myData2 = {0},
})
```

-- 在脚本中就可以像平时获取值一样获取

```
hs1k.i2v("xTYG","_myData1")
hs1k.i2v("xTYG","_myData2")
```

```
hs1k.i2v(hs1k.n2i("一个单位"),"_myData1")
hs1k.i2v(hs1k.n2i("一个单位"),"_myData2")
```

```
hs1k.n2v("一个单位","_myData1")
hs1k.n2v("一个单位","_myData2")
```

hs1k为单位添加技能例子

-- hs1k可以自定义值上面已讲过，所以我们可以设定一些技能后面调用

```
hs1k_unit({
    _id_force = "xTYG",
    Name = "一个单位",
    _mySkillIds = {"A001","A002"},
})
```

```
-- 后面脚本里:
local unitId = "xTYG"
local whichUnit= hunit.create({whichPlayer=hplayer.players[1],id =
unitId,x=0,y=0})
local mySkillIds = hslk.i2v(unitId,"_mySkillIds")
for _,id in ipairs(mySkillIds)do
    hskill.add(whichUnit,id)
end

-- 除此之外, 你可以和原生一样直接在hslk设定技能:
hslk_unit({
    _id_force = "xTYG",
    Name = "一个单位",
    abillist = "A001,A002",
})
-- 后面框架会自动帮你处理数据, 完事
```

Slk原生物编字段

```
abillist = "主动技能ID列表"
Requires = "科技树"
Requiresamount = "科技树-需求值"
Name = "名称"
Description = "描述"
Tip = "描述标题"
Ubertip = "说明"
Hotkey = "热键"
Art = "图标"
scale = "模型大小"
file = "模型文件路径"
Buttonpos = "X坐标"
Buttonpos = "Y坐标"
UnButtonpos = "X坐标-关闭"
UnButtonpos = "Y坐标-关闭"
selSize = "选择圈大小"
colorR = "红"
colorG = "绿"
colorB = "蓝"
armor = "装甲类型"
Level = "等级"
oldLevel = "等级(旧)"
class = "分类"
goldcost = "黄金"
lumbercost = "木头"
HP = "生命"
stockStart = "开始库存"
stockRegen = "补货周期"
stockMax = "最大库存"
prio = "优先权"
morph = "有效的物品转换目标"
drop = "死亡时掉落"
```

```
powerup = "捡到时自动使用"
sellable = "可以出售"
pawnable = "可以抵押"
droppable = "可以丢弃"
pickRandom = "可以作为随机物品"
uses = "使用次数"
perishable = "使用后完全消失"
usable = "主动使用"
level = "等级"
race = "种族"
type = "类别"
manan = "最大魔法值"
regenMana = "魔法恢复"
mana0 = "初始魔法"
regenHP = "生命恢复"
regenType = "生命恢复类型"
fmade = "提供人口"
fused = "占用人口"
stockStart = "雇佣开始时间"
stockRegen = "雇佣开始间隔"
stockMax = "最大库存"
sight = "白天视野"
nsight = "夜晚视野"
collision = "接触体积"
modelScale = "模型缩放"
fileVerFlags = "模型文件版本"
scaleBull = "缩放投射物"
scale = "选择圈缩放"
selZ = "选择圈高度"
selCircOnWater = "选择圈在水面上"
red = "红"
green = "绿"
blue = "蓝"
occH = "闭塞高度"
maxPitch = "X轴最大旋转角度"
maxRoll = "Y轴最大旋转角度"
impactZ = "射弹碰撞偏移Z"
impactSwimZ = "射弹碰撞偏移Z深水"
launchX = "射弹偏移X"
launchY = "射弹偏移Y"
launchZ = "射弹偏移Z"
launchSwimZ = "射弹偏移Z深水"
unitSound = "单位声音"
RandomSoundLabel = "声音随机"
MovementSoundLabel = "声音移动"
LoopingSoundFadeOut = "声音渐出"
LoopingSoundFadeIn = "声音渐入"
auto = "主动自动技能ID"
Sellitems = "售出物品"
Sellunits = "售出单位"
Markitems = "制造物品"
Builds = "可建造单位"
Specialart = "特殊效果"
unitShadow = "单位阴影"
```

```
buildingShadow = "建筑阴影"
shadowH = "阴影H"
shadowW = "阴影W"
shadowX = "阴影X"
shadowY = "阴影Y"
shadowOnWater = "深水有阴影"
death = "死亡时间"
deathType = "死亡类型"
movetp = "移动类型"
moveHeight = "移动高度"
moveFloor = "最小高度"
spd = "基础速度"
maxSpd = "最大速度"
minSpd = "最小速度"
turnRate = "转身速率"
acquire = "主动攻击范围"
minRange = "最小攻击范围"
weapsOn = "允许攻击"
Missileart = "箭矢模型"
Missilespeed = "箭矢速度"
Missilearc = "箭矢弧度"
MissileHoming = "箭矢自导"
targs = "目标允许"
atkType = "攻击类型"
weapTp = "武器类型"
weapType = "武器声音"
spillRadius = "穿透伤害范围"
spillDist = "穿透距离"
damageLoss = "伤害衰减参数"
showUI = "显示攻击UI"
backSw = "攻击后摇"
dmgpt = "攻击前摇"
rangeN = "攻击范围"
RngBuff = "攻击范围缓冲"
dmgplus = "基础伤害"
dmgUp = "攻击科技升级"
sides = "伤害骰子面数"
dice = "伤害骰子数量"
splashTargs = "范围影响目标"
cool = "攻击间隔"
Farea = "全伤害范围"
targCount = "最大目标数"
Qfact = "小伤害参数"
Qarea = "小伤害范围"
Hfact = "中伤害参数"
Harea = "中伤害范围"
defType = "护甲类型"
defUp = "护甲科技升级"
def = "基础护甲"
armor = "装甲类型 (被击声音) "
targType = "作为目标类型"
Propernames = "英雄称谓"
nameCount = "英雄称谓个数"
Awakentip = "英雄唤醒提示"
```

```
Revivetip = "英雄重生提示"
Primary = "主属性"
STR = "力量"
STRplus = "力量成长"
AGI = "敏捷"
AGIplus = "敏捷成长"
INT = "智力"
INTplus = "智力成长"
heroAbilList = "英雄技能ID列表"
hideHeroMinimap = "隐藏英雄小地图"
hideHeroBar = "隐藏英雄栏图标"
hideHeroDeathMsg = "隐藏英雄死亡信息"
Requiresaccount = "科技树-使用等级数"
Requires = "科技树-等级"
Reviveat = "指定复活点"
buffRadius = "路径-AI放置范围"
buffType = "路径-AI放置类型"
Revive = "可复活单位"
Trains = "可训练单位"
Upgrade = "可升级建筑"
requirePlace = "路径-放置不允许"
preventPlace = "路径-放置要求"
requireWaterRadius = "路径-放置要求距离水的范围"
pathTex = "路径纹理"
uberSplat = "建筑地面纹理"
nbrandom = "中立-可作为随机建筑"
nbmmlcon = "中立-显示小地图"
canBuildOn = "能被其他建筑建造"
isBuildOn = "能建造在其他建筑上"
tilesets = "大地图-地形设置"
special = "大地图-分类-特殊"
campaign = "大地图-分类-战役"
inEditor = "大地图-可放置"
dropItems = "大地图-死亡掉落物品"
hostilePal = "大地图-可作为中立敌对"
useClickHelper = "大地图-使用点击帮助"
tilesetSpecific = "大地图-有地形指定数据"
DependencyOr = "科技树-从属等价物"
Researches = "科技树-可研究项目"
upgrades = "使用的科技"
EditorSuffix = "编辑器后缀"
Casterupgradename = "魔法升级名字"
Casterupgradetip = "魔法升级说明"
Casterupgradeart = "魔法升级图标"
ScoreScreenIcon = "图标-记分屏"
animProps = "要求动画名"
Attachmentanimprops = "要求动画名-附加动画"
Attachmentlinkprops = "要求动画名-链接名"
Boneprops = "要求骨骼名"
castpt = "动画-施法前摇"
castbsw = "动画-施法后摇"
blend = "动画-混合时间(秒)"
run = "动画-跑步"
walk = "动画-行走"
```

```
propWin = "动画-转向角度"
orientInterp = "动画-转向补正"
teamColor = "队伍颜色"
customTeamColor = "允许更改队伍颜色"
elevPts = "高度变化-采样点数量"
elevRad = "高度变化-采样范围"
fogRad = "战争迷雾-采样范围"
fatLOS = "不可见区域显示单位"
repulse = "组群分离-允许"
repulsePrio = "组群分离-优先"
repulseParam = "组群分离-参数"
repulseGroup = "组群分离-组号"
isbldg = "是一个建筑"
bldtm = "建造时间"
bountyplus = "黄金奖励-值"
bountysides = "黄金奖励-骰子面数"
bountydice = "黄金奖励-骰子数量"
goldRep = "修理黄金消耗"
lumberRep = "修理木材消耗"
reptm = "修理时间"
lumberbountyplus = "木材奖励-值"
lumberbountysides = "木材奖励-骰子面数"
lumberbountydice = "木材奖励-骰子数量"
cargoSize = "运输尺寸"
hideOnMinimap = "隐藏小地图显示"
points = "单位附加值"
prio = "编队优先权"
formation = "队形排列"
canFlee = "可以逃跑"
canSleep = "可以睡眠"
checkDep = "科技树-检查等价所属"
Effectsound = "声效-音效"
Effectsoundlooped = "声效-音效 (循环)"
Researchtip = "学习技能提示"
Researchhubertip = "学习技能描述"
Unorder = "命令-关闭"
Orderon = "命令-激活"
Order = "命令串"
Orderoff = "命令-取消激活"
Unhotkey = "热键-关闭"
Researchhotkey = "热键-学习"
Researchbuttonpos1 = "X坐标-学习"
Researchbuttonpos2 = "Y坐标-学习"
Unart = "图标-关闭"
Researchart = "图标-学习"
SpecialArt = "效果-特殊"
Specialattach = "效果-特殊附加点"
LightningEffect = "效果-闪电效果"
EffectArt = "效果-目标点"
TargetArt = "效果-目标"
Targetattachcount = "效果-目标-附加数量"
Targetattach = "效果-目标-附加"
Areaeffectart = "效果-区域"
Animnames = "施法动作"
```

```

CasterArt = "效果-施法者"
Casterattachcount = "效果-施法者-附加数量"
Casterattach = "效果-施法者-附加"
hero = "是否英雄技能"
item = "是否物品技能"
levels = "等级"
reqLevel = "等级要求"
levelSkip = "学习跳等级"
priority = "魔法偷取优先级"
BuffID = "魔法效果"
EfctID = "区域持续效果"
DataA = "数据A"
DataB = "数据B"
DataC = "数据C"
DataD = "数据D"
DataE = "数据E"
DataF = "数据F"
Cast = "施法时间"
Cool = "冷却时间"
Dur = "持续时间"
HeroDur = "持续时间 (英雄) "
Cost = "魔法消耗"
Rng = "施法距离"
Area = "施法范围"

```

Slk与MISC数据

```

[Ping](4) => table: {
    [PingAttackColor] => <string>"255,255,0,0"
    [PingColor] => <string>"255,0,255,0"
    [PingAdvisorColor] => <string>"255,255,255,0"
    [PingWaypointColor] => <string>"255,255,255,25"
}
[Misc](232) => table: {
    [FadeBuffMinDuration] => <string>"10"
    [BountyTextVelocity] => <string>"0,0.03,100"
    [GrantNormalXP] => <string>"25"
    [LumberTextVelocity] => <string>"0,0.03,100"
    [GoldTextFadeStart] => <string>"1"
    [DamageBonusMagic] => <string>"1.25,0.75,2.00,0.35,1.00,0.50,0.05,1.00"
    [FoggedAttackRevealRadius] => <string>"200.0"
    [BuildingUnblightRadius] => <string>"350"
    [FadeBuffMinAlpha] => <string>"96"
    [FollowItemRange] => <string>"1000"
    [FadeBuffMaxAlpha] => <string>"255"
    [GoldTextColor] => <string>"255,255,220,0"
    [BashTextVelocity] => <string>"0,0.04,100"
    [GoldTextLifetime] => <string>"2"
    [ShadowStrikeTextVelocity] => <string>"0,0.04,100"
    [ScaledAnimTime] => <string>"60"
    [AwakenLevelFactor] => <string>".20"
}

```

```
[GoldTextVelocity] => <string>"0,0.03,100"
[CanDeactivateBarkskin] => <string>"1"
[Dusk] => <string>"18"
[ChanceToMiss] => <string>"0"
[BountyTextColor] => <string>"255,255,220,0"
[BulletDeathTime] => <string>"5"
[StructureFollowRange] => <string>"100"
[LumberTextColor] => <string>"255,0,200,80"
[BuildingAngle] => <string>"270"
[CriticalStrikeTextVelocity] => <string>"0,0.04,100"
[BountyTextLifetime] => <string>"3"
[LumberTextLifetime] => <string>"2"
[MaxCollisionRadius] => <string>"200"
[BountyTextFadeStart] => <string>"2"
[NeedHeroXP] => <string>"200"
[LumberTextFadeStart] => <string>"1"
[// revivalTime] => <string>"originalTime * level * ReviveTimeFactor"
[TradingIncLarge] => <string>"0"
[SpellCastRangeBuffer] => <string>"300"
[ReactionDelay] => <string>"0.25"
[StructureDecayTime] => <string>"15"
[MissTextColor] => <string>"255,255,0,0"
[ItemShadowOffset] => <string>"50,50"
[MissTextVelocity] => <string>"0,0.03,100"
[MissTextLifetime] => <string>"3"
[MissTextFadeStart] => <string>"1"
[CriticalStrikeTextColor] => <string>"255,255,0,0"
[DefendDeflection] => <string>"0"
[ShadowStrikeTextFadeStart] => <string>"2"
[CriticalStrikeTextLifetime] => <string>"5"
[CriticalStrikeTextFadeStart] => <string>"2"
[ShadowStrikeTextColor] => <string>"255,160,255,0"
[ManaBurnTextVelocity] => <string>"0,0.04,100"
[ShadowStrikeTextLifetime] => <string>"5"
[ManaBurnTextColor] => <string>"255,82,82,255"
[BuildingPlacementNotifyRadius] => <string>"600"
[MorphLandClosest] => <string>"1"
[ManaBurnTextLifetime] => <string>"5"
[ManaBurnTextFadeStart] => <string>"2"
[BashTextColor] => <string>"255,0,0,255"
[IllusionsGetAttackSpeedBonus] => <string>"0"
[BashTextLifetime] => <string>"5"
[BashTextFadeStart] => <string>"2"
[HeroExpRange] => <string>"1200"
[LavaSpawnColor] => <string>"255,255,255,255,255,220,220,220,255,185,185,185"
[ActiveButtonColor] => <string>"255,75,150,255"
[DamageBonusNormal] => <string>"1.00,1.50,1.00,0.70,1.00,1.00,0.05,1.00"
[CloseEnoughRange] => <string>"100"
[NeutralUseNotifyRadius] => <string>"900"
[StrRegenBonus] => <string>"0"
[ReviveMaxTimeFactor] => <string>"2.0"
[RootAngle] => <string>"250"
[AttackHalfAngle] => <string>"0.5"
[InvisibleUnitsBestowAuras] => <string>"0"
```

```
[DyingRevealRadius] => <string>"500.0"
[BoneDecayTime] => <string>"60"
[DecayTime] => <string>"2"
[FlyingHeroesBestowAuras] => <string>"0"
[FogFlashTime] => <string>"3"
[DissipateTime] => <string>"3"
[CancelTime] => <string>"3"
[EffectDeathTime] => <string>"5"
[CanDeactivateBearForm] => <string>"1"
[CreepCampPathingCellDistance] => <string>"26"
[FollowRange] => <string>"300"
[RallyZOffset] => <string>"200"
[HeroMaxAwakenCostGold] => <string>"1400"
[DayLength] => <string>"480"
[Dawn] => <string>"6"
[DayHours] => <string>"24"
[GoldMineMaxGold] => <string>"1000000"
[AllowMultiBounce] => <string>"1"
[InvisSpeed] => <string>"0.4"
[LowGoldAmount] => <string>"1500"
[CanDeactivateWindWalk] => <string>"0"
[GoldMineOwnDuration] => <string>"2.0"
[IntManaBonus] => <string>"0"
[// this is the speed units change visibility (smaller] => <string>""cloak"
slower)
[SelectionCircleBaseZ] => <string>"16"
[ItemShadowFile] => <string>"Shadow"
[ItemShadowSize] => <string>"120,120"
[StrAttackBonus] => <string>"0"
[TownPortalCluster] => <string>"1"
[AttackNotifyDelay] => <string>"30.0"
[DrainTransfersLife] => <string>"0"
[AttackNotifyRange] => <string>"1250"
[CanDeactivateRavenForm] => <string>"1"
[TradingIncSmall] => <string>"0"
[MissDamageReduction] => <string>"0."
[MagicImmunesResistDamage] => <string>"1"
[MagicImmunesResistThorns] => <string>"0"
[MagicImmunesResistLeech] => <string>"0"
[MagicImmunesResistUltimates] => <string>"0"
[CycloneStasis] => <string>"0"
[GrantNormalXPFormulaB] => <string>"5"
[ReviveTimeFactor] => <string>"0.65"
[DepCheckAlias] => <string>"1"
[MassTeleportCluster] => <string>"1"
[DarkSummoningCluster] => <string>"1"
[AmuletOfRecallCluster] => <string>"1"
[MorphAlternateDisable] => <string>"0"
[InvulnSummonDispelDamage] => <string>"1"
[ConstructionDamageRefundPenalty] => <string>"1"
[UpgradeDamageRefundPenalty] => <string>"0"
[MaxUnitSpeed] => <string>"522"
[EnsnareIsMagic] => <string>"0"
[MinUnitSpeed] => <string>"0"
```

```
[WebIsMagic] => <string>"0"
[IllusionsGetAttackBonus] => <string>"0"
[IllusionsGetMoveSpeedBonus] => <string>"0"
[IllusionsGetDefenseBonus] => <string>"0"
[IllusionsCanRestoreLife] => <string>"0"
[IllusionsCanRestoreMana] => <string>"0"
[IllusionsBestowAuras] => <string>"0"
[CanDeactivateAvengerForm] => <string>"0"
[IllusionsGetAutocast] => <string>"0"
[GrantNormalXPFormulaA] => <string>"1"
[DrainGivesBonusMana] => <string>"0"
[PolymorphedUnitsBestowAuras] => <string>"0"
[BurrowedUnitsBestowAuras] => <string>"0"
[DamageBonusChaos] => <string>"1.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00"
[AnimatedUnitsBestowAuras] => <string>"0"
[MoveSpeedBonusesStack] => <string>"1"
[DrainUsesEtherealBonus] => <string>"0"
[DrainTransfersMana] => <string>"0"
[AgiAttackSpeedBonus] => <string>"0"
[DrainGivesBonusLife] => <string>"0"
[EtherealDamageBonusAlly] => <string>"1"
[CanDeactivateAvatar] => <string>"0"
[CanDeactivateBladestorm] => <string>"0"
[AgiMoveBonus] => <string>"0"
[CanDeactivateBurrow] => <string>"1"
[CanDeactivateCallToArms] => <string>"1"
[CanDeactivateChemicalRage] => <string>"0"
[CanDeactivateCorporealForm] => <string>"1"
[DisplayBuildingStatus] => <string>"0"
[CanDeactivateDefend] => <string>"1"
[GrantNormalXPFormulaC] => <string>"5"
[CanDeactivateDivineShield] => <string>"1"
[CanDeactivateImmolation] => <string>"1"
[CanDeactivateManaFlare] => <string>"0"
[CanDeactivateManaShield] => <string>"1"
[CanDeactivateMetamorphosis] => <string>"0"
[CanDeactivateRoboGoblin] => <string>"1"
[CanDeactivateStoneForm] => <string>"1"
[CanDeactivateSubmerge] => <string>"1"
[RelativeUpgradeCost] => <string>"1"
[ItemSaleAggroRange] => <string>"0"
[UnitSaleAggroRange] => <string>"0"
[AbilSaleAggroRange] => <string>"0"
[AbolishMagicDispelsSmart] => <string>"0"
[UpgradeInProgressIdChange] => <string>"1"
[GlobalExperience] => <string>"1"
[HeroMaxAwakenCostLumber] => <string>"350"
[MaxLevelHeroesDrainExp] => <string>"1"
[BuildingKillsGiveExp] => <string>"0"
[DisplayEnemyInventory] => <string>"1"
[AgiDefenseBonus] => <string>"0"
[HeroMaxReviveCostGold] => <string>"700"
[AwakenLumberLevelFactor] => <string>".20"
[HeroMaxReviveCostLumber] => <string>"0"
```

```
[HeroMaxReviveTime] => <string>"150"
[GuardReturnTime] => <string>"5.0"
[HeroReviveManaStart] => <string>"1"
[HeroReviveManaFactor] => <string>"0.0"
[HeroReviveLifeFactor] => <string>"1.0"
[HeroAwakenManaStart] => <string>"0"
[PawnItemRate] => <string>"0.50"
[HeroAwakenManaFactor] => <string>"0.0"
[HeroAwakenLifeFactor] => <string>"0.5"
[// goldRevivalCost] => <string>"originalCost * (ReviveBaseFactor +
(ReviveLevelFactor*(level-1)))"
[// lumberRevivalCost] => <string>"originalCost * (ReviveBaseLumberFactor +
(ReviveLumberLevelFactor*(level-1)))"
[ReviveBaseFactor] => <string>".40"
[ReviveLevelFactor] => <string>".10"
[ReviveBaseLumberFactor] => <string>"0"
[DamageBonusSpells] => <string>"1.00,1.00,1.00,1.00,1.00,0.75,0.05,1.00"
[ReviveLumberLevelFactor] => <string>"0"
[ReviveMaxFactor] => <string>"4.0"
[FrostAttackSpeedDecrease] => <string>"0"
[AwakenBaseFactor] => <string>".80"
[FrostMoveSpeedDecrease] => <string>"0"
[AwakenBaseLumberFactor] => <string>".80"
[AwakenMaxFactor] => <string>"8.0"
[MinBldgSpeed] => <string>"25"
[MaxBldgSpeed] => <string>"400"
[MaxHeroLevel] => <string>"10"
[MaxUnitLevel] => <string>"99"
[GrantHeroXP] => <string>"100,120,160,220,300"
[HeroFactorXP] => <string>"80,70,60,50,0"
[EtherealHealBonus] => <string>"1.66"
[SummonedKillFactor] => <string>"0.5"
[StrHitPointBonus] => <string>"0"
[IntRegenBonus] => <string>"0"
[AgiDefenseBase] => <string>"0"
[// f(x)] => <string>"A*f(x-1) + B*x + C"
[NeedHeroXPFormulaA] => <string>"1"
[NeedHeroXPFormulaB] => <string>"100"
[NeedHeroXPFormulaC] => <string>"0"
[GrantHeroXPFormulaA] => <string>"1"
[GrantHeroXPFormulaB] => <string>"0"
[GrantHeroXPFormulaC] => <string>"100"
[HeroAbilityLevelSkip] => <string>"2"
[DropItemRange] => <string>"100"
[GiveItemRange] => <string>"150"
[PickupItemRange] => <string>"150"
[PawnItemRange] => <string>"300"
[CallForHelp] => <string>"600"
[CreepCallForHelp] => <string>"600"
[DefenseArmor] => <string>"0.06"
[DamageBonusPierce] => <string>"2.00,0.75,1.00,0.35,1.00,0.50,0.05,1.50"
[DamageBonusSiege] => <string>"1.00,0.50,1.00,1.50,1.00,0.50,0.05,1.50"
[DamageBonusHero] => <string>"1.00,1.00,1.00,0.50,1.00,1.00,0.05,1.00"
[EtherealDamageBonus] => <string>"0,0,0,1.66,0,1.66,0"
```

```
[GuardDistance] => <string>"600"
[MaxGuardDistance] => <string>"1000"
}
[Ques](1) => table: {
    [QuestIndicatorTimeout] => <string>"10"
}
[Came](19) => table: {
    [AOA] => <string>"20"
    [top] => <string>"2"
    [FarZ] => <string>"5000,5000,5000,5000,5000,5000"
    [FOV] => <string>"20"
    [SmoothMaxDist] => <string>"1000"
    [SmoothScale] => <string>"0.80"
    [NearZ] => <string>"100,100,100,100,100,100,60"
    [Rotation] => <string>"30"
    [Height] => <string>"0,0,0,0,0,0,0"
    [Distance] => <string>"1200"
    [Smoothing] => <string>"0.00"
    [SmoothBias] => <string>"0.15"
    [SmoothMinFPS] => <string>"10"
    [right] => <string>"4"
    [SmoothBeta] => <string>"1.5"
    [Forward] => <string>"3000"
    [Strafe] => <string>"3000"
    [left] => <string>"4"
    [bottom] => <string>"2"
}
[Info](5) => table: {
    [MeleeRangeMax] => <string>"128"
    [SpeedSlow] => <string>"220"
    [SpeedAverage] => <string>"280"
    [AttackAverage] => <string>"1.5"
    [AttackSlow] => <string>"2"
}
[Flye](2) => table: {
    [MaximizeRadius] => <string>"6"
    [SmoothLevels] => <string>"3"
}
[Unpa](4) => table: {
    [left] => <string>"6"
    [right] => <string>"6"
    [top] => <string>"8"
    [bottom] => <string>"4"
}
[Defa](5) => table: {
    [Style] => <string>"0"
    [Start] => <string>"20000"
    [End] => <string>"50000"
    [Density] => <string>"0"
    [Color] => <string>"0,0,0,0"
}
[Arth](33) => table: {
    [File1] => <string>"Doodads\Cinematic\ArthasIllidanFight\ArthasCape0001.mrf"
    [File21] => <string>"Doodads\Cinematic\ArthasIllidanFight\ArthasCape0004.mrf"
```

```

[Style] => <string>"0"
[File13] =>
<string>"Doodads\Cinematic\ArthasIllidanFight\IllidanDeathShadowA.blp"
[Start] => <string>"0"
[End] => <string>"6000"
[Density] => <string>"1"
[File3] => <string>"Doodads\Cinematic\ArthasIllidanFight\ArthasCloakFight.blp"
[Color] => <string>"255,92,112,120"
[File0] => <string>"Sound\Ambient\DoodadEffects\FinalCinematic.mp3"
[File4] => <string>"Doodads\Cinematic\ArthasIllidanFight\Arthas1.blp"
[File5] =>
<string>"Doodads\Cinematic\ArthasIllidanFight\StoneTiledSnowTALLtrev.blp"
[File6] => <string>"Doodads\Cinematic\ArthasIllidanFight\IllidanBlade.blp"
[File7] => <string>"Doodads\Cinematic\ArthasIllidanFight\FrostMorne.blp"
[File8] => <string>"Doodads\Cinematic\ArthasIllidanFight\BigAssSky.blp"
[File9] => <string>"Doodads\Cinematic\ArthasIllidanFight\ArthasCape0000.mrf"
[File10] => <string>"Doodads\Cinematic\ArthasIllidanFight\ArthasCape0002.mrf"
[File11] => <string>"Doodads\Cinematic\ArthasIllidanFight\ArthasCape0003.mrf"
[File12] =>
<string>"Doodads\Cinematic\ArthasIllidanFight\SpeedLines1_solid.blp"
[File14] =>
<string>"Doodads\Cinematic\ArthasIllidanFight\IllidanCineStuff.blp"
[File15] => <string>"Doodads\Cinematic\ArthasIllidanFight\IllidanCine.blp"
[File16] => <string>"Doodads\Cinematic\ArthasIllidanFight\IceCrownWalls.blp"
[File17] =>
<string>"Doodads\Cinematic\ArthasIllidanFight\IceCrownDoodads0.blp"
[File18] => <string>"Doodads\Cinematic\ArthasIllidanFight\Ice_Natural01.blp"
[File19] => <string>"Doodads\Cinematic\ArthasIllidanFight\Clouds8x8.blp"
[File20] => <string>"Doodads\Cinematic\ArthasIllidanFight\Ice_Natural01.blp"
[File22] => <string>"Doodads\Cinematic\ArthasIllidanFight\ArthasCape0005.mrf"
[File23] =>
<string>"Doodads\Cinematic\ArthasIllidanFight\SacrificialAltarskull1.blp"
[File24] => <string>"Doodads\Cinematic\ArthasIllidanFight\rockShadow.blp"
[File25] =>
<string>"Doodads\Cinematic\ArthasIllidanFight\IronForgeSnow01solid1.blp"
[File26] => <string>"Doodads\Cinematic\ArthasIllidanFight\furboots2.blp"
[File27] =>
<string>"Doodads\Cinematic\ArthasIllidanFight\IllidanBlade_mip1.blp"
[Scene1] => <string>"FinalCinematic"
}

[Terr](9) => table: {
    [MaxHeight] => <string>"1536"
    [Style] => <string>"0"
    [Start] => <string>"3000"
    [End] => <string>"5000"
    [// 0] => <string>"low ground, each cliff level = 128"
    [Density] => <string>"0.5"
    [MinHeight] => <string>"-128"
    [Color] => <string>"255,0,0,0"
    [MaxSlope] => <string>"50"
}

[Menu](5) => table: {
    [Style] => <string>"0,0"
    [Start] => <string>"100,0"
}

```

```
[End] => <string>"5500,7000"
[Density] => <string>"1,1"
[Color] => <string>"255,11,20,11,255,178,178,204"
}
[Soun](5) => table: {
    [MusicFadeInterval] => <string>"500"
    [MinDist] => <string>"1200"
    [Attenuation] => <string>"3.0"
    [MaxDist] => <string>"20000"
    [MusicFadeRate] => <string>"16"
}
[Targ](2) => table: {
    [TargetLightColor] => <string>"180,0,255,0"
    [SelTargetColor] => <string>"255,240,200,0"
}
[Ligh](1) => table: {
    [Direction] => <string>"0.3,0.3,-0.25"
}
[Sele](15) => table: {
    [NumSizes] => <string>"3"
    [TextureDotted00] =>
<string>"ReplaceableTextures\Selection\SelectionCircleSmallDotted"
    [ColorEnemy] => <string>"255,255,0,0"
    [Texture00] => <string>"ReplaceableTextures\Selection\SelectionCircleSmall"
    [AllyOffset] => <string>"0.2"
    [Size00] => <string>"100"
    [ColorNeutral] => <string>"255,255,255,0"
    [TextureDotted01] =>
<string>"ReplaceableTextures\Selection\SelectionCircleMedDotted"
    [Texture01] => <string>"ReplaceableTextures\Selection\SelectionCircleMed"
    [Size01] => <string>"300"
    [ScaleFactor] => <string>"72"
    [TextureDotted02] =>
<string>"ReplaceableTextures\Selection\SelectionCircleLargeDotted"
    [Texture02] => <string>"ReplaceableTextures\Selection\SelectionCircleLarge"
    [ImageWalkableZOffset] => <string>"15"
    [ColorFriend] => <string>"255,0,255,0"
}
[Blig](7) => table: {
    [DCT_s2] => <string>"10"
    [Texture] => <string>"TerrainArt\Blight\Blight"
    [DCT_d2] => <string>"20"
    [PuffModel] => <string>"Environment\BlightDoodad\BlightDoodad.mdl"
    [DCT_s1] => <string>"5"
    [DCT_sd] => <string>"15"
    [DCT_d1] => <string>"5"
}
[Wate](5) => table: {
    [DeepLevelPathing] => <string>"52"
    [MinDepth] => <string>"10"
    [DeepLevel] => <string>"64"
    [MaxDepth] => <string>"72"
    [WavesDepth] => <string>"25"
}
```

```
[Team](4) => table: {
    [ColorIndexPlayer] => <string>"1"
    [ColorIndexAlly] => <string>"2"
    [ColorIndexCreep] => <string>"12"
    [ColorIndexEnemy] => <string>"0"
}
[Fog0](25) => table: {
    [BlackMaskedObject] => <string>"255,0,0,0"
    [FoggedTerrain] => <string>"170,16,16,32"
    [FoggedObject] => <string>"255,64,64,96"
    [BlackMaskedTerrain] => <string>"255,0,0,0"
    [DarkMaskedTerrain] => <string>"230,0,0,0"
    [BoundaryTerrain] => <string>"230,0,0,0"
    [FogColorCreepAllied] => <string>"255,0,0,50"
    [BoundaryObject] => <string>"255,0,0,0"
    [FoggedBoundaryTerrain] => <string>"170,16,16,32"
    [FogColorPlayer] => <string>"255,255,255,255"
    [DarkMaskedObject] => <string>"255,32,32,48"
    [FoggedBoundaryObject] => <string>"255,64,64,96"
    [ShadowImageColor] => <string>"170,0,0,0"
    [FogColorAlly] => <string>"255,0,255,210"
    [FogColorEnemy] => <string>"255,255,0,0"
    [FogColorResource] => <string>"255,32,128,128"
    [FogColorItem] => <string>"255,240,240,0"
    [FogColorHero] => <string>"255,255,255,128"
    [FogColorCreepNormal] => <string>"255,0,0,50"
    [FogColorDestructable] => <string>"255,0,160,128"
    [FogColorBlackMaskAlpha] => <string>"0"
    [FogColorDarkMaskAlpha] => <string>"96"
    [FogColorFogMaskAlpha] => <string>"192"
    [FogColorVisibleAlpha] => <string>"255"
    [FogColorBlightMask] => <string>"0,63,63,64"
}
[Mini](8) => table: {
    [MinimapToughCampColor] => <string>"255,220,0,0"
    [MinimapToughCampThreshold] => <string>"20"
    [MinimapColorBorder] => <string>"255,60,60,70"
    [MinimapMiddleCampThreshold] => <string>"10"
    [MinimapMiddleCampScale] => <string>"1.3"
    [MinimapWeakCampColor] => <string>"255,0,200,0"
    [MinimapMiddleCampColor] => <string>"255,255,128,0"
    [MinimapCampPulseScale] => <string>"0.5"
}
[Glue](3) => table: {
    [ControlFadeDuration] => <string>"0.25"
    [ControlFadeDurationDebug] => <string>"0.01"
    [CustomLoadTimerDuration] => <string>"10.0"
}
[Batt](3) => table: {
    [LargeMapRange] => <string>"9,12"
    [SmallMapRange] => <string>"2,4"
    [MediumMapRange] => <string>"5,8"
}
[Occl](4) => table: {
```

```
[BuildingsOcclude] => <string>"0"
[OccluderColor] => <string>"128,255,255,255"
[MarkImage] => <string>"ReplaceableTextures\Occlusion\OcclusionMark"
[MarkColor] => <string>"255,255,0,255"
}
[Plac](2) => table: {
    [PaidStructureColor] => <string>"96,128,128,255"
    [UnpaidStructureColor] => <string>"96,255,128,128"
}
```

sync 同步

网易给我们提供的与同步数据有关的函数有4条：

hjapi已附带

```
DzSyncData
DzTriggerRegisterSyncData
DzGetTriggerSyncPlayer
DzGetTriggerSyncData
```

原理和服务器请求响应一样，简单易懂

一个发，一个收（注意过程时间）

若使用hasync库，例子

```
-- 通用型操作，两个配套
hasync.onSend("hzg", function(syncData)
    echo(syncData.triggerData[1] .. syncData.triggerData[2] .. "人类")
end)
local a = 0
htime.setInterval(2, function(curTimer)
    hasync.send("hzg", { "hunzsig", "是个" .. a .. "级" })
    a = a + 1
end)
```

:例子

使用排行榜

创建两种排行榜，一种木头，一种黄金；只记录玩家1、玩家2

并且每20秒换一种展示

```
hleaderBoard.create('gold', '黄金榜', 1, function(leaderBoardKey)
    print("leaderBoardKey", leaderBoardKey)
```

```

        return {
            { playerIndex = 1, value = hplayer.getGold(hplayer.players[1]) },
            { playerIndex = 2, value = hplayer.getGold(hplayer.players[2]) }
        }
    end)
hleaderBoard.create('lumber', '木头榜', 1, function(leaderBoardKey)
    print("leaderBoardKey", leaderBoardKey)
    return {
        { playerIndex = 1, value = hplayer.getLumber(hplayer.players[1]) },
        { playerIndex = 2, value = hplayer.getLumber(hplayer.players[2]) }
    }
end)

htime.setInterval(20, function(_)
    if (hleaderBoard.CURRENT == 'gold') then
        hleaderBoard.show('lumber')
    else
        hleaderBoard.show('gold')
    end
end)

```

使用多面板

创建常见的英雄属性多面板，1.5秒更新一次数据

```

hmultiBoard.create(
    "player",
    1.5,
    function(mb)
        --拼凑多面板数据，二维数组，行列模式
        hmultipBoard.setTitle(mb, "剑圣求生多面板")
        --开始当然是title了
        local data = {}
        table.insert(data, {
            { value = "玩家", icon = nil },
            { value = "称号", icon = nil },
            { value = "战力", icon = nil },
            { value = "杀敌", icon = nil },
            { value = "英雄", icon = nil },
            { value = "回血", icon = nil },
            { value = "回魔", icon = nil },
            { value = "攻速", icon = nil },
            { value = "命中", icon = nil },
            { value = "增幅", icon = nil },
            { value = "减伤", icon = nil },
            { value = "魔抗", icon = nil },
            { value = "回避", icon = nil },
            { value = "背包", icon = nil },
            { value = "硬直", icon = nil },
            { value = "攻击类型", icon = nil },
        })
    end)

```

```
--然后是form
for pi = 1, hplayer.qty_max, 1 do
    local p = hplayer.players[pi]
    if (his.playing(p)) then
        local hero = "-"
        local prestige = "-"
        local power = "-"
        local kill = "-"
        local avatar = nil
        local name = "-"
        local life_back = "-"
        local mana_back = "-"
        local attack_speed = "-"
        local damage_extent = "-"
        local aim = "-"
        local toughness = "-"
        local resistance = "-"
        local avoid = "-"
        local weight = "-"
        local punish = "-"
        local attack_damage_type = "-"
        hero = hhero.player_heroes[pi][1]
        if (hero ~= nil) then
            avatar = hunit.getAvatar(hero)
            name = hunit.getName(hero)
            prestige = hplayer.getPrestige(p)
            power = math.integerFormat(game.playerData.power[pi] or 0)
            kill = math.integerFormat(hplayer.getKill(p))
            life_back = math.round(hattr.get(hero, "life_back")) .. "/秒"
            mana_back = math.round(hattr.get(hero, "mana_back")) .. "/秒"
            attack_speed = math.round(100 + hattr.get(hero,
                "attack_speed")) .. "%"
            damage_extent = math.round(hattr.get(hero, "damage_extent"))
            .. "%"
            aim = math.round(hattr.get(hero, "aim")) .. "%"
            toughness = math.round(hattr.get(hero, "toughness"))
            resistance = math.round(hattr.get(hero, "resistance")) .. "%"
            avoid = math.round(hattr.get(hero, "avoid")) .. "%"
            weight = math.round(hattr.get(hero, "weight_current")) .. "/"
                .. math.round(hattr.get(hero, "weight")) .. "Kg"
            punish = math.round(hattr.get(hero, "punish_current")) .. "/"
                .. math.round(hattr.get(hero, "punish"))
            local adt = {}
            for _, v in ipairs(hattr.get(hero, "attack_damage_type")) do
                local label = CONST_ATTR[v]
                if (table.includes(label, adt) == false) then
                    table.insert(adt, label)
                end
            end
            attack_damage_type = string.implode('、', adt)
        end
        table.insert(data, {
            { value = "[" .. hplayer.getStatus(p) .. "]" .. cj.GetPlayerName(p), icon = nil },
        })
    end
end
```

```

        { value = prestige, icon = nil },
        { value = power, icon = nil },
        { value = kill, icon = nil },
        { value = name, icon = avatar },
        { value = life_back, icon = nil },
        { value = mana_back, icon = nil },
        { value = attack_speed, icon = nil },
        { value = aim, icon = nil },
        { value = damage_extent, icon = nil },
        { value = toughness, icon = nil },
        { value = resistance, icon = nil },
        { value = avoid, icon = nil },
        { value = weight, icon = nil },
        { value = punish, icon = nil },
        { value = attack_damage_type, icon = nil },
    })
end
return data
end
)

```

难度选择对话框

创建一个难度选择对话框，自动挑选第一位玩家，让他选

```

hdialog.create(
nil, --自动挑选第一位玩家，让他选
{
    title = "选择难度",
    buttons = {
        "一般困难",
        "非常困难",
        "极奇困难",
        "破天荒难",
    }
},
function(btnIdx)
    if (btnIdx == "非常困难") then
        --搞点事情
    elseif (btnIdx == "极奇困难") then
        --搞点事情
    elseif (btnIdx == "破天荒难") then
        --搞点事情
    else
        --搞点事情
    end
    echo("选择了: " .. btnIdx)
end
)

```

创建一个带热键的难度选择对话框，自动挑选玩家3，让他选

```

hdialog.create(
    hplayer.players[3],
    {
        title = "选择难度",
        buttons = [
            { value = "Q", label = "一般困难" },
            { value = "W", label = "非常困难" },
            { value = "E", label = "极奇困难" },
            { value = "R", label = "破天荒难" },
        ]
    },
    function(btnIdx)
        if (btnIdx == "Q") then
            --搞点事情
        elseif (btnIdx == "W") then
            --搞点事情
        elseif (btnIdx == "E") then
            --搞点事情
        elseif (btnIdx == "R") then
            --搞点事情
        end
        echo("选择了: " .. btnIdx)
    end
)

```

常见两种选英雄

酒馆模式

```

hhero.setBornXY(0, -300) --设定英雄出生点
hhero.buildSelector({
    during = 60,
    type = "tavern",
    buildX = -512, -- 酒馆构建点X
    buildY = 512, -- 酒馆构建点Y
    buildDistance = 256, --酒馆间距离
    buildRowQty = 5, --酒馆一个横行有几个
    tavernUnitQty = 12 --酒馆内最多有几种英雄
})

```

双击模式

```

hhero.setBornXY(0, -300) --设定英雄出生点
hhero.buildSelector({
    during = 60,
    type = "click",
})

```

```

buildX = -512, -- 第一个英雄构建点X
buildY = 512, -- 第一个英雄构建点Y
buildDistance = 256,--每位英雄间距离
buildRowQty = 5 --一个横行有几个英雄
})

```

简单刷兵

每3秒在 (0,0) 刷2个步兵，攻击 (100,100) 坐标，一共刷10次

步兵的初始移动速度为175

```

local counter = 10;
htime.setInterval(3.00, function(curTimer)
    if (counter <= 0) then
        htime.delTimer(curTimer)
    end
    counter = counter - 1
    hunit.create({
        whichPlayer = hplayer.players[1], --归属玩家
        id = "hfoo", --类型id,如'H001'
        x = 0, --创建坐标X, 可选
        y = 0, --创建坐标Y, 可选
        qty = 2, --数量, 可选, 可选
        attackX = 100, --攻击X, 可选
        attackY = 100, --攻击Y, 可选
        attr = { --自定义属性, 可选
            move = '=175'
        },
    });
end)

```

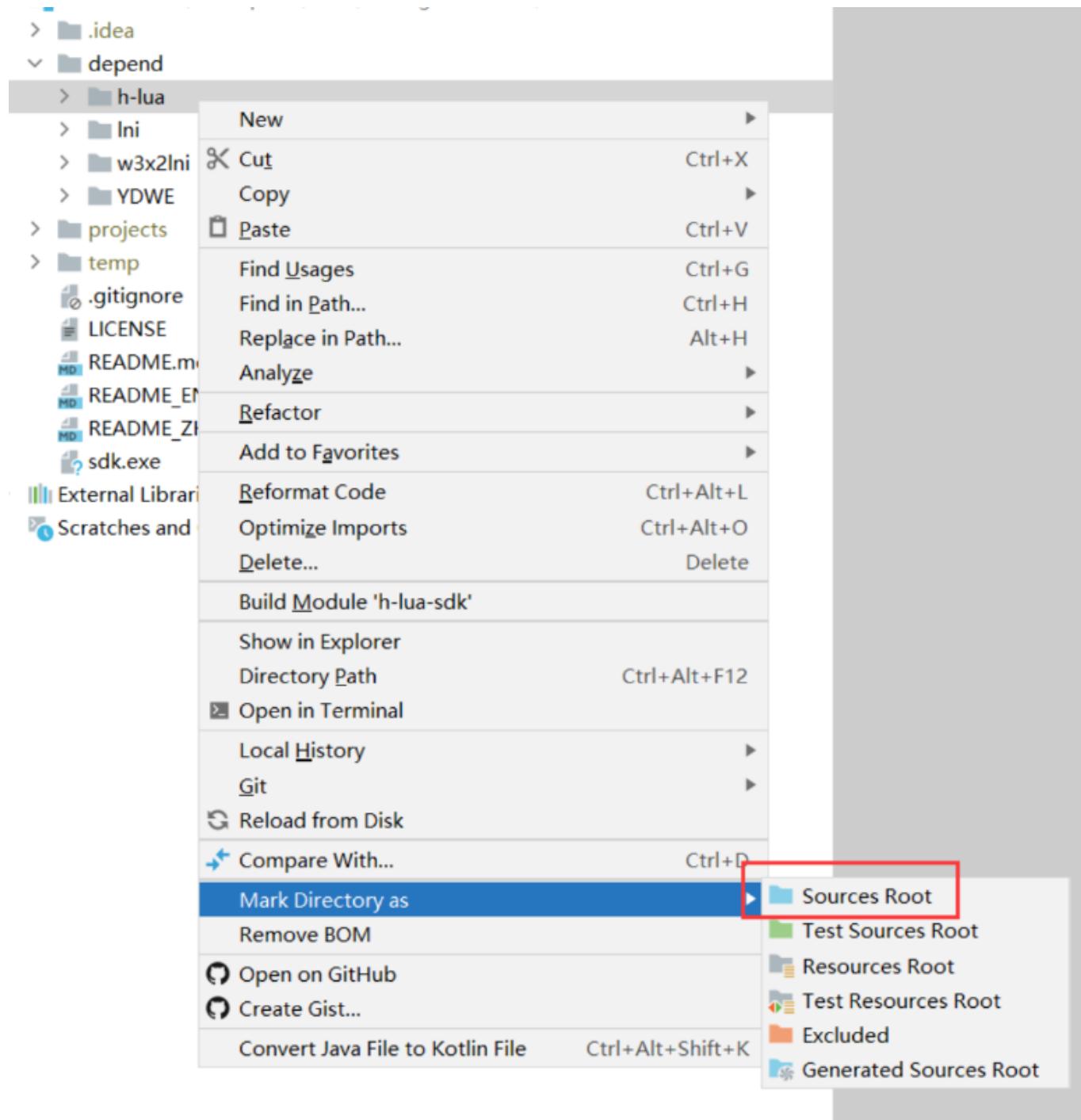
:其他学习

Jetbrain设置学习

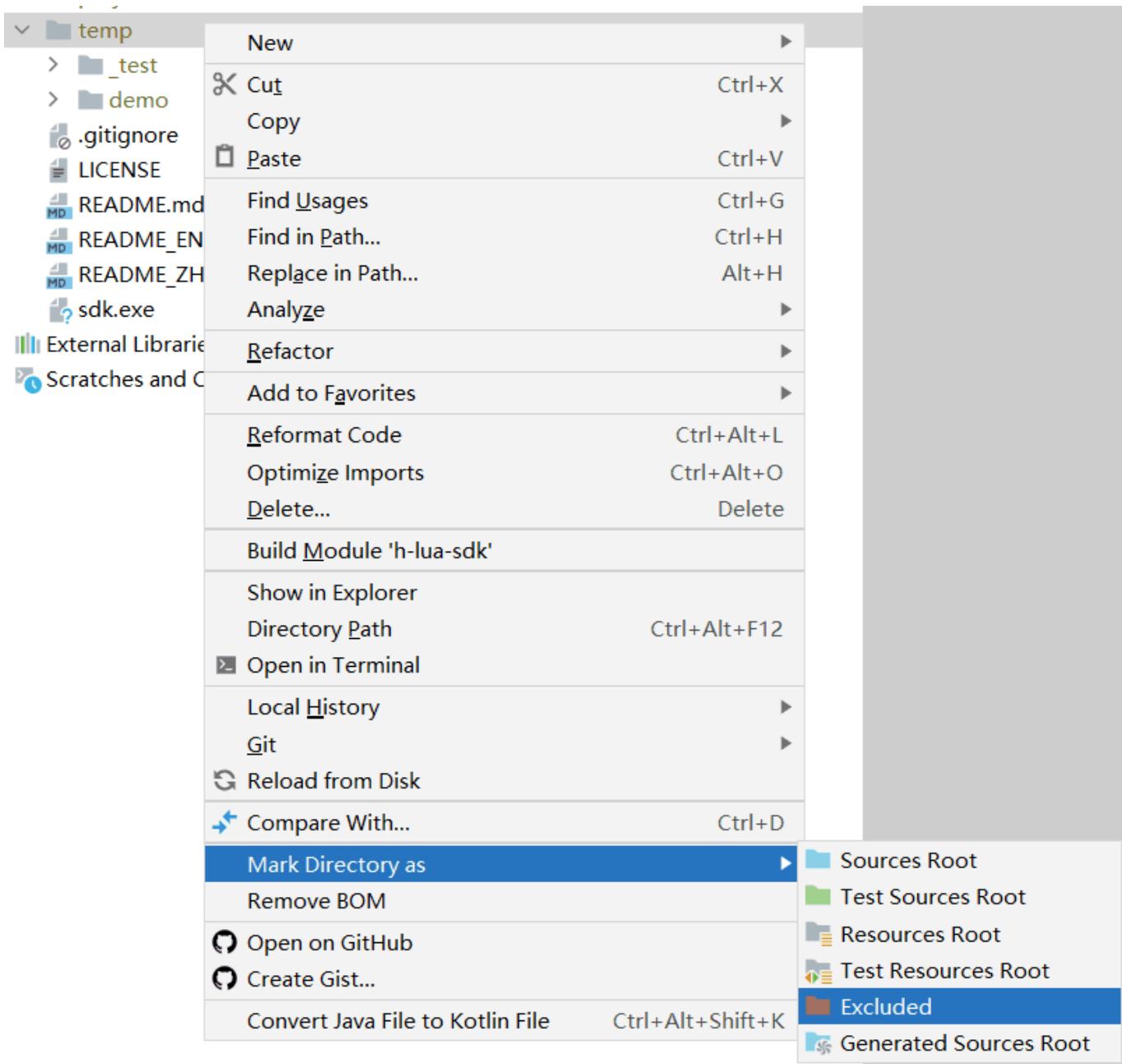
安装 Emmylua 插件，如虎添翼



可以为 h-lua 目录设置sources，提高索引效率



可以为 temp 目录设置 excluded，忽略其索引



Q & A

懂得问问题才能让你变得更强！

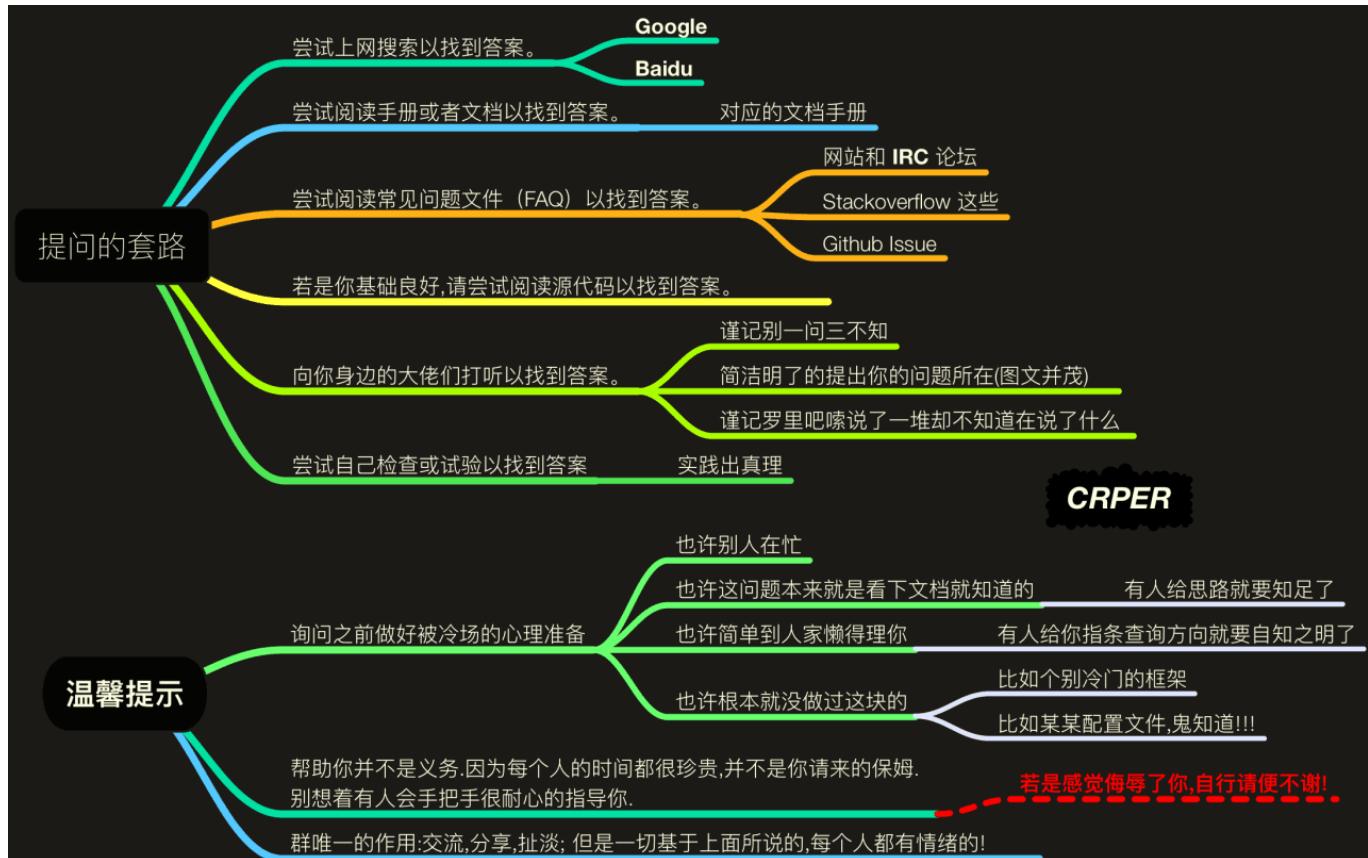
下列问问题方法相信可以帮助你~

错误的提问：

- x. 在？
- x. 有人吗？
- x. 有大佬吗？
- x. 我这个是怎么回事？【一张报错截图】
- x. 有谁能帮我解决一下这个问题？【一张报错截图】

正确的提问：

✓. 我遇到了一个问题：【问题描述】，我想要实现【X功能】但出现【Y现象】，我经过了以下尝试【思路细节与代码实现】，但不能解决，报错如下【报错信息/截图】，我使用【关键词】谷歌但找不到答案。请问我该怎么解决？【截图相关代码片段】



上线官方对战平台须知

上线的地图需要在发布后台勾选上启用平台的 japi 选项, 否则无法调用 lua 引擎库

- 1、第一步做好你的图, 做不好就不用看后面了
- 2、使用 sdk 将项目打包, 如我的项目叫 h-lua-sdk-helloworld

```
.\sdk.exe build h-lua-sdk-helloworld
或缩写
.\sdk.exe -b h-lua-sdk-helloworld
```

- 3、项目打包完会启动测试, 此时已经在你的 war3 根目录的 map/test 目录内有了结果文件, 如:

\Warcraft3\Maps\Test\WorldEditTestMap.w3x 这个文件已经经过 slk 加密

- 4、把优化好的地图文件改个名, 例如你的地图是剑圣求生之路: jsqsl.v1.0.0.w3x, 就可以上传地图到官方对战平台了!

如果你够强还可以对脚本进行加密, 由于是很简单的东西, 这里不做赘述。

CONGRATULATIONS!

魔兽作图小技巧

如何令加载地图时播放音乐

随意创建一个 F4 触发器

转为自定义代码并删除里面所有代码

贴上以下代码即可：

- 此方法在重制版魔兽中可能是无效的

```
function hBgm takes string s returns nothing
    local string uri = "main.mp3" //这个路径你可以播放默认的音乐（在F5）也可以播放F12
导入的音乐
    call SetMapDescription(s)
    call PlayMusic(uri)
    set uri = null
endfunction
#define SetMapDescription(s) hBgm(s)
```

为什么我的mp3背景音乐不能循环播放

背景音乐无法循环播放时，可以尝试格式工厂转wav再转回mp3

一般问题已解决~

如何令一个技能隐藏但是依然有效

设置它的图标坐标为 (0,-11)

物品编辑器如何输入负数的数值

按住shift再双击即可输入负值，某些技能负数会引起崩溃，如“水元素”

如何修改单位的最大生命/魔法

利用生命牌的bug可以实现，例如X点生命值。

有个叫生命护身符的物品，将它的技能设为2级，1级数据为0，2级数据为-X，将技能添加给目标单位后，设为2级，然后删除

目标单位会永久增加X点生命值

- 最大魔法值同理

魔兽地形贴图路径大全

白杨谷

```
泥土: TerrainArt\Ashenvale\Ashen_Dirt.blp
坑洼的泥土: TerrainArt\Ashenvale\Ashen_DirtRough.blp
草地: TerrainArt\Ashenvale\Ashen_Grass.blp
岩石: TerrainArt\Ashenvale\Ashen_Rock.blp
粗草地: TerrainArt\Ashenvale\Ashen_GrassLumpy.blp
藤蔓: TerrainArt\Ashenvale\Ashen_Vines.blp
```

草色泥土: TerrainArt\Ashenvale\Ashen_DirtGrass.blp
叶子: TerrainArt\Ashenvale\Ashen_Leaves.blp

贫瘠之地

泥土: TerrainArt\Barrens\Barrens_Dirt.blp
坑洼的泥土: TerrainArt\Barrens\Barrens_DirtRough.blp
石子地: TerrainArt\Barrens\Barrens_Pebbles.blp
草色泥土: TerrainArt\Barrens\Barrens_DirtGrass.blp
沙漠: TerrainArt\Barrens\Barrens_Desert.blp
深色沙漠: TerrainArt\Barrens\Barrens_DesertDark.blp
岩石: TerrainArt\Barrens\Barrens_Rock.blp
草地: TerrainArt\Barrens\Barrens_Grass.blp

城邦

泥土: TerrainArt\Cityscape\City_Dirt.blp
坑洼的泥土: TerrainArt\Cityscape\City_DirtRough.blp
黑色大理石: TerrainArt\Cityscape\City_BlackMarble.blp
砖: TerrainArt\Cityscape\City_BrickTiles.blp
正方形地形: TerrainArt\Cityscape\City_SquareTiles.blp
圆形地形: TerrainArt\Cityscape\City_RoundTiles.blp
草地: TerrainArt\Cityscape\City_Grass.blp
平整草地: TerrainArt\Cityscape\City_GrassTrim.blp
白色大理石: TerrainArt\Cityscape\City_WhiteMarble.blp

达拉然

泥土: TerrainArt\Dalaran\Dalaran_Dirt.blp
坑洼的泥土: TerrainArt\Dalaraan\Dalaran_DirtRough.blp
黑色大理石: TerrainArt\Dalaraan\Dalaran_BlackMarble.blp
砖: TerrainArt\Dalaraan\Dalaran_BrickTiles.blp
正方形地形: TerrainArt\Dalaraan\Dalaran_SquareTiles.blp
圆形地形: TerrainArt\Dalaraan\Dalaran_RoundTiles.blp
草地: TerrainArt\Dalaraan\Dalaran_Grass.blp
平整的草地: TerrainArt\Dalaraan\Dalaran_GrassTrim.blp
白色大理石: TerrainArt\Dalaraan\Dalaran_WhiteMarble.blp

地下城

泥土: TerrainArt\Dungeon\Cave_Dirt.blp
砖: TerrainArt\Dungeon\Cave_Brick.blp
红色石头: TerrainArt\Dungeon\Cave_RedStones.blp
熔岩碎片: TerrainArt\Dungeon\Cave_LavaCracks.blp

火山岩: TerrainArt\Dungeon\Cave_Lava.blp
深色岩石: TerrainArt\Dungeon\Cave_DarkRocks.blp
灰色的石头: TerrainArt\Dungeon\Cave_GreyStones.blp
正方形地形: TerrainArt\Dungeon\Cave_SquareTiles.blp

洛丹伦(秋)

泥土: TerrainArt\LordaeronFall\Lordf_Dirt.blp
坑洼的泥土: TerrainArt\LordaeronFall\Lordf_DirtRough.blp
草色泥土: TerrainArt\LordaeronFall\Lordf_DirtGrass.blp
岩石: TerrainArt\LordaeronFall\Lordf_Rock.blp
草地: TerrainArt\LordaeronFall\Lordf_Grass.blp
深色草地: TerrainArt\LordaeronFall\Lordf_GrassDark.blp

洛丹伦(夏)

泥土: TerrainArt\LordaeronSummer\Lords_Dirt.blp
坑洼的泥土: TerrainArt\LordaeronSummer\Lords_DirtRough.blp
草色泥土: TerrainArt\LordaeronSummer\Lords_DirtGrass.blp
岩石: TerrainArt\LordaeronSummer\Lords_Rock.blp
草地: TerrainArt\LordaeronSummer\Lords_Grass.blp
深色草地: TerrainArt\LordaeronSummer\Lords_GrassDark.blp

洛丹伦(冬)

泥土: TerrainArt\LordaeronWinter\Lordw_Dirt.blp
坑洼的泥土: TerrainArt\LordaeronWinter\Lordw_DirtRough.blp
积雪的草: TerrainArt\LordaeronWinter\Lordw_SnowGrass.blp
岩石: TerrainArt\LordaeronWinter\Lordw_Rock.blp
草地: TerrainArt\LordaeronWinter\Lordw_Grass.blp
雪: TerrainArt\LordaeronWinter\Lordw_Snow.blp

诺森德

泥土: TerrainArt\Northrend\North_dirt.blp
黑泥: TerrainArt\Northrend\North_dirtdark.blp
岩石: TerrainArt\Northrend\North_rock.blp
草地: TerrainArt\Northrend\North_Grass.blp
冰: TerrainArt\Northrend\North_ice.blp
雪: TerrainArt\Northrend\North_Snow.blp
积雪石头: TerrainArt\Northrend\North_SnowRock.blp

村庄

泥土: TerrainArt\Village\Village_Dirt.blp
坑洼的泥土: TerrainArt\Village\Village_DirtRough.blp
庄稼: TerrainArt\Village\Village_Crops.blp
鹅卵石路: TerrainArt\Village\Village_CobblePath.blp
石头路径: TerrainArt\Village\Village_StonePath.blp
草地: TerrainArt\Village\Village_GrassShort.blp
岩石: TerrainArt\Village\Village_Rocks.blp
厚草地: TerrainArt\Village\Village_GrassThick.blp

村庄(秋)

泥土: TerrainArt\VillageFall\VillageFall_Dirt.blp
坑洼的泥土: TerrainArt\VillageFall\VillageFall_DirtRough.blp
庄稼: TerrainArt\VillageFall\VillageFall_Crops.blp
鹅卵石路: TerrainArt\VillageFall\VillageFall_CobblePath.blp
石头路径: TerrainArt\VillageFall\VillageFall_StonePath.blp
草地: TerrainArt\VillageFall\VillageFall_GrassShort.blp
岩石: TerrainArt\VillageFall\VillageFall_Rocks.blp
厚草地: TerrainArt\VillageFall\VillageFall_GrassThick.blp

地底

泥土: TerrainArt\Castle2\GDirt.blp
砖: TerrainArt\Castle2\GBrick.blp
红色石头: TerrainArt\Castle2\GRedStones.blp
冰块: TerrainArt\Castle2\GLavaCracks.blp
冰: TerrainArt\Castle2\GLava.blp
深色石头: TerrainArt\Castle2\GDrakRocks.blp
灰色石头: TerrainArt\Castle2\GGreyStones.blp
正方形地形: TerrainArt\Castle2\GSquareTiles.blp

费尔伍德

泥土: TerrainArt\Felwood\Felwood_Dirt.blp
坑洼的泥土: TerrainArt\Felwood\Felwood_DirtRough.blp
污染地面: TerrainArt\Felwood\Felwood_Poison.blp
岩石: TerrainArt\Felwood\Felwood_Rock.blp
藤蔓: TerrainArt\Felwood\Felwood_Vines.blp
草地: TerrainArt\Felwood\Felwood_Grass.blp
叶子: TerrainArt\Felwood\Felwood_Leaves.blp

达拉然遗迹

泥土: TerrainArt\DalaranRuins\DRuins_Dirt.blp
坑洼的泥土: TerrainArt\DalaranRuins\DRuins_DirtRough.blp
黑色大理石: TerrainArt\DalaranRuins\DRuins_BlackMarble.blp
砖: TerrainArt\DalaranRuins\DRuins_BrickTiles.blp
正方形地形: TerrainArt\DalaranRuins\DRuins_SquareTiles.blp
圆形地形: TerrainArt\DalaranRuins\DRuins_RoundTiles.blp
草地: TerrainArt\DalaranRuins\DRuins_Grass.blp
平整的草地: TerrainArt\DalaranRuins\DRuins_GrassTrim.blp
白色大理石: TerrainArt\DalaranRuins\DRuins_WhiteMarble.blp

黑色城堡

泥土: TerrainArt\BlackCitadel\Citadel_Dirt.blp
淡色泥土: TerrainArt\BlackCitadel\Citadel_DirtLight.blp
坑洼的泥土: TerrainArt\BlackCitadel\Citadel_RoughDirt.blp
平石: TerrainArt\BlackCitadel\Citadel_FlatStones.blp
小砖: TerrainArt\BlackCitadel\Citadel_SmallBricks.blp
大砖: TerrainArt\BlackCitadel\Citadel_LargeBricks.blp
正方形地形: TerrainArt\BlackCitadel\Citadel_SquareTiles.blp
深色地形: TerrainArt\BlackCitadel\Citadel_DarkTiles.blp

冰封王座

泥土: TerrainArt\Icecrown\Ice_Dirt.blp
坑洼的泥土: TerrainArt\Icecrown\Ice_DirtRough.blp
黑冰: TerrainArt\Icecrown\Ice_DarkIce.blp
黑色的砖: TerrainArt\Icecrown\Ice_BlackBricks.blp
刻有北欧文字的砖: TerrainArt\Icecrown\Ice_RuneBricks.blp
平铺的砖: TerrainArt\Icecrown\Ice_TiledBricks.blp
冰: TerrainArt\Icecrown\Ice_Ice.blp
黑色的正方形: TerrainArt\Icecrown\Ice_BlackSquares.blp
雪: TerrainArt\Icecrown\Ice_Snow.blp

边缘之地

泥土: TerrainArt\Outland\Outland_Dirt.blp
淡色泥土: TerrainArt\Outland\Outland_DirtLight.blp
坑洼的泥土: TerrainArt\Outland\Outland_RoughDirt.blp
碎土: TerrainArt\Outland\Outland_DirtCracks.blp
平石: TerrainArt\Outland\Outland_FlatStones.blp
岩石: TerrainArt\Outland\Outland_Rock.blp
淡色平石: TerrainArt\Outland\Outland_FlatStonesLight.blp
深渊: TerrainArt\Outland\Outland_Abyss.blp

沉沦的遗迹

```
泥土: TerrainArt\Ruins\Ruins_Dirt.blp  
坑洼的泥土: TerrainArt\Ruins\Ruins_DirtRough.blp  
草色泥土: TerrainArt\Ruins\Ruins_DirtGrass.blp  
小砖: TerrainArt\Ruins\Ruins_SmallBricks.blp  
沙地: TerrainArt\Ruins\Ruins_Sand.blp  
大砖: TerrainArt\Ruins\Ruins_LargeBricks.blp  
圆形地形: TerrainArt\Ruins\Ruins_RoundTiles.blp  
草地: TerrainArt\Ruins\Ruins_Grass.blp  
深色草地: TerrainArt\Ruins\Ruins_GrassDark.blp
```

ydwe lua引擎使用说明

来源 [github/actboy168/jass2lua](https://github.com/actboy168/jass2lua)

简介

ydwe lua引擎(以下简称lua引擎)是一个嵌入到《魔兽争霸III》(以下简称魔兽)中的一个插件，它可以让魔兽可以执行lua并且调用魔兽的导出函数(在common.j内定义的函数)，就像使用jass那样。本说明假定你已经掌握了jass和lua的相关语法，有关语法的问题不再另行解释。

入口

在jass内调用 `call Cheat("exec-lua: hello")`，这等价于在lua里调用了 `require 'hello'`。lua引擎已经把地图内的文件加载到搜索路径，所以地图内的hello.lua将会得到执行。

lua引擎对标准lua的修改

为了适合在魔兽内使用lua引擎对lua略有修改。

1. math.randomseed改为使用jass函数SetRandomSeed实现。
2. math.random改为使用jass函数GetRandomReal实现。
3. table元素随机化种子依赖于魔兽内部的随机种子。
4. 屏蔽了部分被认为不安全的函数

内置库

lua引擎一共有12个内置库，可以通过"require '库名'"调用。

- jass.common
- jass.ai
- jass.globals
- jass.japi
- jass.hook
- jass.runtime
- jass.slk
- jass.console
- jass.debug

- jass.log
- jass.message
- jass.bignum

jass.common

jass.common库包含common.j内注册的所有函数。 (不包括BJ)

```
local jass = require 'jass.common'  
print(jass.GetHandleId(jass.Player(0)))
```

jass.ai

jass.ai库包含common.ai内注册的所有函数。

```
local jass = require 'jass.common'  
local ai = require 'jass.ai'  
print(ai.UnitAlive(jass.GetTriggerUnit()))
```

jass.globals

jass.globals库可以让你访问到jass内的全局变量。

你可以使用此库访问预设在大地图的对象。

```
local cg = require 'jass.globals'  
print(cg.udg_i) --获取jass中定义的i整数
```

jass.japi

jass.japi库当前已经注册的所有japi函数。 (包含dz函数)

```
local jass = require 'jass.common'  
local japi = require 'jass.japi'  
japi.EXDisplayChat(jass.Player(0), 0, "Hello!")
```

japi函数不同环境下可能会略有不同，你可以通过pairs遍历当前的所有japi函数

```
for k, v in pairs(require 'jass.japi') do  
    print(k, v)  
end
```

jass.hook

jass.hook库可以对common.j内注册的函数下钩子。注：jass.common库不会受到影响。

同时，为了避免jass和lua之间传递浮点数时产生误差，通过jass.hook传递到lua中的浮点数，并不是number类型，而是userdata。当你需要精确地操纵浮点数时，也请注意这点。

```
local hook = require 'jass.hook'
function hook.CreateUnit(pid, uid, x, y, face, realCreateUnit)
    -- 当jass内调用CreateUnit时，就会被执行
    print('CreateUnit')
    print(type(x))
    return realCreateUnit(pid, uid, x, y, face)
end
```

jass.slk

jass.slk库可以在地图运行时读取地图内的slk/w3*文件。

```
local slk = require 'jass.slk'
print(slk.ability.AHbz.Name)
```

你也可以遍历一个表或者一个物体（不建议方式）

```
local slk = require 'jass.slk'
for k, v in pairs(slk.ability) do
    print(k, v)
end
for k, v in pairs(slk.ability.AHbz) do
    print(k, v)
end
```

slk包含

- unit
- item
- destructable
- doodad
- ability
- buff
- upgrade
- misc

与你物体编辑器中的项目一一对应。

获取数据时使用的索引你可以在物体编辑器中通过Ctrl+D来查询到

注意，当访问正确时返回值永远是字符串。如果你获取的是某个单位的生命值，你可能需要使用tonumber来进行转换。当访问不正确时将返回nil。

jass.runtime

jass.runtime库可以在地图运行时获取lua引擎的信息或修改lua引擎的部分配置。

```
local runtime = require 'jass.runtime'
```

runtime.console(默认为false)

赋值为true后会打开一个cmd窗口，print与console.write函数可以输出到这里

```
runtime.console = true
```

runtime.version

返回当前lua引擎的版本号

```
print(runtime.version)
```

runtime.error_handle

当你的lua脚本出现错误时将会调用此函数。

runtime.error_handle有一个默认值，等价于以下函数

```
runtime.error_handle = function(msg)
    print("Error: ", msg, "\n")
end
```

你也可以让它输出更多的信息，比如输出错误时的调用栈

```
runtime.error_handle = function(msg)
    print("-----")
    print("          LUA ERROR!!          ")
    print("-----")
    print(tostring(msg) .. "\n")
    print(debug.traceback())
    print("-----")
end
```

注意，注册此函数后lua脚本的效率会降低(即使并没有发生错误)。

runtime.handle_level(默认为0)

lua引擎处理的handle的安全等级，有效值为0~2，注，等级越高，效率越低，安全性越高、

0: handle直接使用number，jass无法了解你在lua中对这个handle的引用情况，也不会通过增加引用计数来保护这个handle

```
local t = jass.CreateTimer()
print(t) -- 1048000
type(t) -- "number"
```

1: handle封装在lightuserdata中，保证handle不能和整数相互转换，同样不支持引用计数

```
local t = jass.CreateTimer()
print(t) -- "handle: 0x10005D"
type(t) -- "userdata"
jass.TimerStart(t, 1, false, 0) -- ok
```

```
local t = jass.CreateTimer()
local h1 = jass.CreateTimer()
jass.DestroyTimer(h1)
jass.TimerStart(t, 1, false,
    function()
        local h2 = jass.CreateTimer()
        print(h1) -- "handle: 0x10005E"
        print(h2) -- "handle: 0x10005F"
    end
)
```

2: handle封装在userdata中，lua持有该handle时将增加handle的引用计数。lua释放handle时会释放handle的引用计数。

```
local t = jass.CreateTimer()
local h1 = jass.CreateTimer()
jass.DestroyTimer(h1)
jass.TimerStart(t, 1, false,
    function()
        local h2 = jass.CreateTimer()
        print(h1) -- "handle: 0x10005E"
        print(h2) -- "handle: 0x10005F"
    end
)
```

runtime.sleep(默认为false)

common.j中包含sleep操作的函数有4个，TriggerSleepAction/TriggerSyncReady/TriggerWaitForSound/SyncSelections。当此项为false时，lua引擎会忽略这4个函数的调用，并给予运行时警告。当此项为true时，这4个函数将会得到正确的执行。

但请注意此项为true时将降低lua引擎的运行效率，即使你没有使用这4个函数。

```
local trg = jass.CreateTrigger()
local a = 1
jass.TriggerAddAction(trg, function()
    jass.TriggerSleepAction(0.2)
    print(a) -- 2
end)
jass.TriggerExecute(trg)
a = 2
```

runtime.catch_crash(默认为true)

调用jass.xxx/japi.xxx发生崩溃时，会生产一个lua错误，并忽略这个崩溃。你可以注册jass.runtime.error_handle来获得这个错误。注：开启此项会略微增加运行时消耗（即使没有发生错误）。

runtime.debugger

启动调试器并监听指定端口。需要使用[VSCode](#)并安装[Lua Debug](#)。

```
runtime.debugger = 4279
```

jass.console

jass.console与控制台相关

console.enable(默认为false)

赋值为true后会打开一个cmd窗口，print与console.write函数可以输出到这里

```
console.enable = true
```

console.write

将utf8编码的字符串转化为ansi编码后输出到cmd窗口中，如果你需要输出魔兽中的中文，请使用该函数而不是print

console.read

将控制台中的输入传入魔兽中(会自动转换编码)

首次调用console.read后将允许用户在控制台输入，输入完成后按回车键提交输入。

用户提交完成后，传入一个函数f来调用console.read，将会调用函数f，并将用户的输入作为参数传入(已转换为utf8编码)。

推荐的做法是每0.1秒运行一次console.read，见下面的例子：

```

local jass      = require 'jass.common'
local console = require 'jass.console'

console.write('测试开始...')

--开启计时器,每0.1秒检查输入
jass.TimerStart(jass.CreateTimer(), 0.1, true,
function()

    --检查CMD窗口中的用户输入,如果用户有提交了的输入,则回调函数(按回车键提交输入).否则不做任何动作
    console.read(
        function(str)
            --参数即为用户的输入.需要注意的是这个函数调用是不同步的(毕竟其他玩家不知道你输入了什么)
            jass.DisplayTimedTextToPlayer(jass.Player(0), 0, 0, 60, '你在
控制台中输入了:' .. str)
        end
    )
end
)

```

需要注意的是控制台输入是不同步的。

jass.debug

jass.debug库能帮助你更深入地剖析lua引擎的内部机制。

- functiondef jass.common或者jass.japi函数的定义

```

local jass = require 'jass.common'
local dbg = require 'jass.debug'
print(table.unpack(dbg.functiondef(jass.GetUnitX)))

```

- globaldef jass.globals内值的定义
- handledef handle对对象的内部定义
- currentpos 当前jass执行到的位置
- handlemax jass虚拟机当前最大的handle

- handlecount jass虚拟机当前的handle数
- h2i/i2h handle和integer的转换，当你runtime.handle_level不是0时，你可能会需要它
- handle_ref 增加handle的引用
- handle_unref 减少handle的引用
- gchash (已废弃) 指定一张table的gchash，gchash会决定了在其他table中这个table的排序次序
在默认的情况下，lua对table的排序次序是由随机数决定的，不同玩家的lua生成的随机数不一致，所以下面的代码在不同的玩家上执行的次序是不一致的，这可能会引起不同步掉线

jass.log

日志库

- path 日志的输出路径
- level 日志的等级，指定等级以上的日志才会输出
- 日志有6个等级 trace、debug、info、warn、error、fatal

```
local log = require 'jass.log'  
log.info('这是一行日志')  
log.error('这是一行', '日志')
```

jass.message

- keyboard 一张表，魔兽的键盘码
- mouse 本地玩家的鼠标坐标(游戏坐标)
- button 本地玩家技能按钮的状态
- hook 魔兽的消息回调，可以获得部分鼠标和键盘消息
- selection 获得本地玩家当前选中单位
- order_immediate 发布本地命令，无目标
- order_point 发布本地命令，点目标
- order_target 发布本地命令，单位目标
- order_enable_debug 开启后，会在控制台打印当前的本地命令，调试用

jass.bignum

大数据库

UI 与 FDF

UI制作就是运用界面函数对原有界面进行增删查改的美化

FDF就是一种格式，让你可以编写一些frame静态组件声明，用处不大

至于为什么说用处不大，这里就留给您自行思考了

下面是一些网上的教程，虽然不咋的，反正看看也好

[War3]Fdf文件详解，简单的UI教程演示 - 魔兽争霸3

魔兽默认组件对照

[老CK教程]官方平台界面功能入门

教程看完了吗？为什么说不咋的，因为其实这些教程不过是给你写一些FDF罢了，就是入个门，看个印象

如果你想看FDF，魔兽客户端自带就有一大堆暴雪写好的，完全不需要看他教程，暴雪写的更全更多

已经提取了原生的fdf，有兴趣可自行查看

可参考提取项目：[FrameDef](#)

```
UI\FrameDef\GlobalStrings.fdf
UI\FrameDef\InfoPanelStrings.fdf
UI\FrameDef\NetworkStrings.fdf
UI\FrameDef\DateStrings.fdf
UI\FrameDef\Glue\DialogWar3.fdf
UI\FrameDef\Glue\MapInfoPane.fdf
UI\FrameDef\Glue\AdvancedOptionsDisplay.fdf
UI\FrameDef\Glue\AdvancedOptionsPane.fdf
UI\FrameDef\Glue\ListBoxWar3.fdf
UI\FrameDef\Glue\MapListBox.fdf
UI\FrameDef\Glue\DecoratedMapListBox.fdf
UI\FrameDef\Glue\MapPreferenceBox.fdf
UI\FrameDef\Glue\CheckListBox.fdf
UI\FrameDef\Glue\MainMenu.fdf
UI\FrameDef\Glue\Loading.fdf
UI\FrameDef\Glue\LoadSavedGameScreen.fdf
UI\FrameDef\Glue\OptionsMenu.fdf
UI\FrameDef\Glue\SinglePlayerMenu.fdf
UI\FrameDef\Glue\ViewReplayScreen.fdf
UI\FrameDef\Glue\CustomCampaignMenu.fdf
UI\FrameDef\Glue\CampaignMenu.fdf
UI\FrameDef\Glue\TeamSetup.fdf
UI\FrameDef\Glue\PlayerSlot.fdf
UI\FrameDef\Glue\Skirmish.fdf
UI\FrameDef\Glue\ScoreScreen.fdf
UI\FrameDef\Glue\LocalMultiplayerJoin.fdf
UI\FrameDef\Glue\LocalMultiplayerCreate.fdf
UI\FrameDef\Glue\LocalMultiplayerLoad.fdf
UI\FrameDef\Glue\GameChatroom.fdf
UI\FrameDef\Glue\CampaignListBox.fdf
UI\FrameDef\Glue\BattleNetMain.fdf
UI\FrameDef\Glue\BattleNetNewsBox.fdf
UI\FrameDef\Glue\BattleNetUserListBox.fdf
UI\FrameDef\Glue\BattleNetChatPanel.fdf
UI\FrameDef\Glue\BattleNetChatActionMenu.fdf
UI\FrameDef\Glue\BattleNetStandardPanel.fdf
UI\FrameDef\Glue\BattleNetTeamPanel.fdf
UI\FrameDef\Glue\BattleNetTeamInvitation.fdf
UI\FrameDef\Glue\BattleNetProfilePanel.fdf
```

```
UI\FrameDef\Glue\BattleNetCustomJoinPanel.fdf
UI\FrameDef\Glue\BattleNetCustomCreatePanel.fdf
UI\FrameDef\Glue\BattleNetCustomLoadPanel.fdf
UI\FrameDef\Glue\BattleNetScheduledGame.fdf
UI\FrameDef\Glue\BattleNetChatroom.fdf
UI\FrameDef\Glue\BattleNetFriendsListBox.fdf
UI\FrameDef\Glue\BattleNetFriendsPane.fdf
UI\FrameDef\Glue\BattleNetClanMateListBox.fdf
UI\FrameDef\Glue\BattleNetClanPane.fdf
UI\FrameDef\Glue\BattleNetClanInvitation.fdf
UI\FrameDef\Glue\BattleNetStatusBox.fdf
UI\FrameDef\Glue\BattleNetIconSelectBox.fdf
UI\FrameDef\UI\ConsoleUI.fdf
UI\FrameDef\UI>AllianceSlot.fdf
UI\FrameDef\UI>AllianceDialog.fdf
UI\FrameDef\UI\ChatDialog.fdf
UI\FrameDef\UI\LogDialog.fdf
UI\FrameDef\UI\EscMenuMainPanel.fdf
UI\FrameDef\UI\EscMenuOptionsPanel.fdf
UI\FrameDef\UI\EscMenuSaveGamePanel.fdf
UI\FrameDef\UI\GameResultDialog.fdf
UI\FrameDef\UI\UpperButtonBar.fdf
UI\FrameDef\UI\ResourceBar.fdf
UI\FrameDef\UI\SimpleInfoPanel.fdf
UI\FrameDef\UI\SuspendDialog.fdf
UI\FrameDef\UI\UnresponsiveDialog.fdf
UI\FrameDef\UI\ReplayPanel.fdf
UI\FrameDef\UI\QuestDialog.fdf
UI\FrameDef\UI\TimerDialog.fdf
UI\FrameDef\UI\Leaderboard.fdf
UI\FrameDef\UI\Multiboard.fdf
UI\FrameDef\UI\CinematicPanel.fdf
UI\FrameDef\UI\ScriptDialog.fdf
UI\FrameDef\UI\GameSaveSplashDialog.fdf`
```

:小小专栏

小小演示

h-lua-sdk基础指令介绍

B站演示

魔兽争霸h-lua-sdk基础指令介绍

命令行

必填 ~ 选填

```
./h-lua-sdk> sdk.exe help //提示cmd工具命令
./h-lua-sdk> sdk.exe new [*PROJECT_NAME] //新建一个地图项目
./h-lua-sdk> sdk.exe we [*PROJECT_NAME] //以YDWE打开地图项目
./h-lua-sdk> sdk.exe model [*PROJECT_NAME] [~PAGE:0|~search:''] //以WE浏览项目模
```

型，一页最大289个，可翻页可搜索

```
./h-lua-sdk> sdk.exe clear [*PROJECT_NAME] //清理构建的临时文件
./h-lua-sdk> sdk.exe test [*PROJECT_NAME] //构建测试版本并开启游戏进行调试
./h-lua-sdk> sdk.exe build [*PROJECT_NAME] //构建上线版本并开启游戏进行调试
```

模型命令拓展

```
. ./h-lua-sdk> sdk.exe model demo //查看项目demo的模型，默认第0页
./h-lua-sdk> sdk.exe model demo 2 //查看项目demo的模型，第2页
./h-lua-sdk> sdk.exe model demo ttg //查看项目demo的模型，只要路径带有ttg的
./h-lua-sdk> sdk.exe model demo abc 1 //查看项目demo的模型，第1页且只要路径带有abc的
```

小小演示

hslk高级用法演示1

B站演示

魔兽争霸hslk高级用法演示1

物品主动技能绑定

_onItemUsed

```
hslk_item({
    Name = "物理学圣剑",
    class = "Permanent",
    _cooldown = 1,
    _cooldownTarget = CONST_ABILITY_TARGET.location.value,
    _attr = _attr({
        attack_green = "+9999"
    }),
    _onItemUsed = _onItemUsed(function(evtData)
        hunit.portal(evtData.triggerUnit,evtData.targetX,evtData.targetY)
    end)
})
```

技能主动绑定

_onSkillEffect

```
hslk_ability({
    Name = "穿云箭",
    Art = "ReplaceableTextures\\CommandButtons\\BTNDeathPact.blp",
    Ubertip = hcolor.yellow("穿云箭"),
    Hotkey = "E",
    _parent = "ANc1",
```

```

_type = "skill",
DataA = {0},
DataC = {1},
EffectArt = "",
TargetArt = "",
CasterArt = "",
Cool = { 2 },
DataB = { 2 },
Cost = { 0 },
DataD = { 0 },
Rng = { 800 },
DataF = {"berserk"},
_onSkillEffect = _onSkillEffect(function(evtData)
    local xy =
math.polarProjection(hunit.x(evtData.triggerUnit),hunit.y(evtData.triggerUnit),
2000,math.getDegBetweenXY(hunit.x(evtData.triggerUnit),hunit.y(evtData.triggerUnit),
),evtData.targetX,evtData.targetY))
    hskill.leap({
        sourceUnit = evtData.triggerUnit, --伤害来源 (必须有! 不管有没有伤害)
        x = xy.x, --冲击的x坐标 (可选的, 对点冲击, 与某目标无关)
        y = xy.y, --冲击的y坐标 (可选的, 对点冲击, 与某目标无关)
        speed = 50, --冲击的速度 (可选的, 默认10, 0.02秒的移动距离,大概1秒500px)
        acceleration = 0.05, --冲击加速度 (可选的, 每个周期都会增加0.02秒一次)
        filter = function(unit)
            return his.enemy(evtData.triggerUnit,unit) and his.alive(unit)
        end, --必须有
        tokenArrow = "war3mapImported\\aaaaaaaa.mdx", --前冲的特效
        (arrowUnit=nil时认为必须! 自身冲击就是bind, 否则为马甲本身, 如冲击波的波)
        damageMovement = 200, --移动过程中的伤害 (可选的, 默认为0)
        damageMovementRadius = 100, --移动过程中的伤害 (可选的, 默认为0, 易知0范围是无效的所以有伤害也无法体现)
        damageMovementRepeat = false, --移动过程中伤害是否可以重复造成 (可选的,
        默认为不能)
        damageMovementDrag = false, --移动过程是否拖拽敌人 (可选的, 默认为不能)
        damageSrc = CONST_DAMAGE_SRC.skill, --伤害的种类 (可选)
        damageType = {CONST_DAMAGE_TYPE.thunder} --伤害的类型,注意是table (可
        选)
    })
end)
})
}

```

小小演示

hslk高级用法演示2

B站演示

魔兽争霸hslk高级用法演示2

多级技能配置

```

hslk_ability({
    Name = "分裂",
    _parent = "Agyb", -- 用了一个无效果被动技能做为父ID
    Requires = "",
    levels = 3, -- 3级
    _attr = {
        --这里要分级设置attr属性
        --1级属性
        _attr({
            --做分裂效果这里用到了hlua库的xtras
            xtras = {
                {
                    on = CONST_EVENT.attack,action = "targetUnit.spec.split",
                    odds = 100,val = "triggerUnit.attack",percent = 100, radius =
500
                }
            }
        }),
        --2
        _attr({
            xtras = {
                {
                    on = CONST_EVENT.attack,action = "targetUnit.spec.split",
                    odds = 100,val = "triggerUnit.attack",percent = 200, radius =
500
                }
            }
        }),
        --3
        _attr({
            xtras = {
                {
                    on = CONST_EVENT.attack,action = "targetUnit.spec.split",
                    odds = 100,val = "triggerUnit.attack",percent = 300, radius =
500
                }
            }
        })
    }
})

```

获取SLK数据及自定义值

```

hslk_unit({
    _parent = "hpea",
    Name = "天选勇者",
    __skill = {
        "穿云箭",
        "分裂"
    },
})

```

```

local u = henemy.create({
    teamNo = 1,
    id = hslk.n2i("天选勇者"),
    x = 0,
    y = 0
})
local uS1k = hslk.i2v(hunit.getId(u), "__skill")
for _, v in ipairs(uS1k) do
    hskill.add(u, hslk.n2i(v), 1)
end

```

小小演示

hslk高级用法演示3

B站演示

魔兽争霸hslk高级用法演示3

hslk的 _type 字段

```

-- 循环创建10把武器, type设置成"aaa"
for i = 1, 10 do
    table.insert(it, {
        Name = "小刀【等级】" .. i,
        _type = "aaa",
        _attr = _attr({
            attack_green = "+" .. 100 * i
        })
    })
end

for i = 1, 10 do
    local un = henemy.create({
        teamNo = 1,
        id = hslk.n2i("天选勇者"),
        x = 0,
        y = 0
    })
    --给这几个单位注册个死亡事件
    hevent.onDead(un, function(evtData)
        --获取
        local id = hslk.typeIds({ "aaa" })
        hitem.create({
            id = table.random(id),
            x = hunit.x(evtData.triggerUnit),
            y = hunit.y(evtData.triggerUnit)
        })
    end)
end

```

小小演示

h-lua库sync演示

B站演示

魔兽争霸h-lua库sync演示

代码

```
local shop = hunit.create({
    whichPlayer = hplayer.players[1],
    id = hs1k.n2i("商店"),
    x = 500,
    y = 500
})
tb = {}
hshop.item.add(shop, hs1k.n2i("奖励"), 99)
--单位购买物品 触发UI
--注册出售物品事件
hevent.onItemSell(shop, function(evtData)
    hitem.del(evtData.soldItem)
    local p = hunit.getOwner(evtData.buyingUnit)
    tb[p] = {}
    local itTy = hs1k.typeIds({ "skill" })
    for i = 1, 3 do
        table.insert(tb[p], table.random(itTy))
    end
    if (p == hplayer.loc()) then
        --设置UI图标 名称 和描述
        for i = 1, 3 do
            local itSlk = hs1k.i2v(tb[p][i], "slk")
            hjapi.DzFrameSetTexture(hjapi.DzFrameFindByName("awar_jn_" .. i, 0),
            itSlk.Art, 0)
            hjapi.DzFrameSetText(hjapi.DzFrameFindByName("awar_jnname_" .. i, 0),
            itSlk.Name)
            hjapi.DzFrameSetText(hjapi.DzFrameFindByName("awar_jnsm_" .. i, 0),
            itSlk.Ubertip)
        end
        hjapi.DzFrameShow(awar, true)
    end
end)

UI = function()
    --加载toc
    hdzui.loadToc("UI\\frame.toc")
    --初始化
    awar = hdzui.frame("awar", hdzui.origin.game(), 0)
    --设置位置
    hjapi.DzFrameSetAbsolutePoint(awar, 4, 0.388, 0.350)
    hjapi.DzFrameShow(awar, false)
```

```

--注册3个按钮点击事件
hsync.onFrameMouse(hjapi.DzFrameFindByName("awar_1",0),1,function(syncData)
    --添加技能
    hskill.add(hero[syncData.triggerPlayer],tb[syncData.triggerPlayer][1],1)
    if (syncData.triggerPlayer ~= hplayer.loc()) then
        return
    end
    hjapi.DzFrameShow(awar,false)
end)
hsync.onFrameMouse(hjapi.DzFrameFindByName("awar_2",0),1,function(syncData)
    --添加技能
    hskill.add(hero[syncData.triggerPlayer],tb[syncData.triggerPlayer][2],1)
    if (syncData.triggerPlayer ~= hplayer.loc()) then
        return
    end
    hjapi.DzFrameShow(awar,false)
end)
hsync.onFrameMouse(hjapi.DzFrameFindByName("awar_3",0),1,function(syncData)
    --添加技能
    hskill.add(hero[syncData.triggerPlayer],tb[syncData.triggerPlayer][3],1)
    if (syncData.triggerPlayer ~= hplayer.loc()) then
        return
    end
    hjapi.DzFrameShow(awar,false)
end)

end

```

fdf文件

```

Frame "FRAME" "awar"{
    Width 0.400,
    Height 0.258,
    LayerStyle "IGNORETRACKEVENTS",
    Frame "BACKDROP" "awarbakc" {
        Width 0.400,
        Height 0.258,
        BackdropBackground "war3mapImported\awarTB.blp",
        BackdropBlendAll,
        SetPoint TOPLEFT, "awar", TOPLEFT, 0, 0,
        Frame "BUTTON" "awar_1" {
            Height 0.213,
            Width 0.125,
            ControlStyle "AUTOTRACK|HIGHLIGHTONMOUSEOVER",
            SetPoint LEFT, "awar", LEFT , 0.010, 0,
            //背景
            ControlBackdrop "awara",
            Frame "BACKDROP" "awara" {
                BackdropBackground "war3mapImported\awarAND.blp",

```

```
        BackdropBlendAll,
    }
    //按下
    ControlPushedBackdrop "awaraa",
    Frame "BACKDROP" "awaraa" {
        BackdropBackground "war3mapImported\awarANX.blp",
        BackdropBlendAll,
    }
    //高亮
    ControlMouseOverHighlight "awarga",
    Frame "HIGHLIGHT" "awarga" {
        HighlightType "FILETEXTURE",
        HighlightAlphaFile "UI\Widgets\BattleNet\bnet-button01-highlight-
mouse.blp",
        HighlightAlphaMode "ADD",
    }
    Frame "BACKDROP" "awar_jn_1" {

        Width 0.034,
        Height 0.042,
        BackdropBackground
"ReplaceableTextures\PassiveButtons\PASBTNSscatterRockets.blp",
        BackdropBlendAll,
        SetPoint TOPLEFT, "awar_1", TOPLEFT, 0.040, -0.040,
    }
    Frame "TEXT" "awar_jnname_1" {
        Width 0.12,
        Height 0.1,
        DecorateFileNames,
        FrameFont "MasterFont", 0.015, "",
        FontJustificationH JUSTIFYCENTER,
        UseActiveContext,
        FontJustificationV JUSTIFYMIDDLE,
        LayerStyle "IGNORETRACKEVENTS",
        SetPoint CENTER, "awar_jn_1", CENTER, 0, -0.035,
        Text "弹幕攻击",
    }
    Frame "TEXT" "awar_jnsm_1" {
        Width 0.080,
        Height 0.1,
        DecorateFileNames,
        FrameFont "MasterFont", 0.01, "",
        FontJustificationH JUSTIFYCENTER,
        UseActiveContext,
        FontJustificationV JUSTIFYMIDDLE,
        LayerStyle "IGNORETRACKEVENTS",
        SetPoint CENTER, "awar_jnname_1", CENTER, 0, -0.035,
        Text "对周围的敌方空中单位进行强有力的弹幕攻击",
    }
}
Frame "BUTTON" "awar_2" {
    Height 0.213,
    Width 0.125,
    ControlStyle "AUTOTRACK|HIGHLIGHTONMOUSEOVER",
```

```
SetPoint LEFT, "awar_1", RIGHT , 0, 0,  
  
//背景  
ControlBackdrop "awarb",  
Frame "BACKDROP" "awarb" {  
    BackdropBackground "war3mapImported\awarAND.blp",  
    BackdropBlendAll,  
}  
//按下  
ControlPushedBackdrop "awarab",  
Frame "BACKDROP" "awarab" {  
    BackdropBackground "war3mapImported\awarANX.blp",  
    BackdropBlendAll,  
}  
//高亮  
ControlMouseOverHighlight "awargb",  
Frame "HIGHLIGHT" "awargb" {  
    HighlightType "FILETEXTURE",  
    HighlightAlphaFile "UI\Widgets\BattleNet\bnet-button01-highlight-  
mouse.blp",  
    HighlightAlphaMode "ADD",  
}  
Frame "BACKDROP" "awar_jn_2" {  
  
    Width 0.034,  
    Height 0.042,  
    BackdropBackground  
"ReplaceableTextures\PassiveButtons\PASBTNSscatterRockets.blp",  
    BackdropBlendAll,  
    SetPoint TOPLEFT, "awar_2", TOPLEFT, 0.040, -0.040,  
}  
Frame "TEXT" "awar_jnname_2" {  
    Width 0.12,  
    Height 0.1,  
    DecorateFileNames,  
    FrameFont "MasterFont", 0.015, "",  
    FontJustificationH JUSTIFYCENTER,  
    UseActiveContext,  
    FontJustificationV JUSTIFYMIDDLE,  
    LayerStyle "IGNORETRACKEVENTS",  
    SetPoint CENTER, "awar_jn_2", CENTER, 0, -0.035,  
    Text "弹幕攻击",  
}  
Frame "TEXT" "awar_jnsm_2" {  
    Width 0.080,  
    Height 0.1,  
    DecorateFileNames,  
    FrameFont "MasterFont", 0.01, "",  
    FontJustificationH JUSTIFYCENTER,  
    UseActiveContext,  
    FontJustificationV JUSTIFYMIDDLE,  
    LayerStyle "IGNORETRACKEVENTS",  
    SetPoint CENTER, "awar_jnname_2", CENTER, 0, -0.035,  
    Text "对周围的敌方空中单位进行强有力的弹幕攻击",
```

```
        }

    }

Frame "BUTTON" "awar_3" {
    Height 0.213,
    Width 0.125,
    ControlStyle "AUTOTRACK|HIGHLIGHTONMOUSEOVER",
    SetPoint LEFT, "awar_2", RIGHT , 0, 0,

    //背景
    ControlBackdrop "awarc",
    Frame "BACKDROP" "awarc" {
        BackdropBackground "war3mapImported\awarAND.blp",
        BackdropBlendAll,
    }
    //按下
    ControlPushedBackdrop "awarac",
    Frame "BACKDROP" "awarac" {
        BackdropBackground "war3mapImported\awarANX.blp",
        BackdropBlendAll,
    }
    //高亮
    ControlMouseOverHighlight "awargc",
    Frame "HIGHLIGHT" "awargc" {
        HighlightType "FILETEXTURE",
        HighlightAlphaFile "UI\Widgets\BattleNet\bnet-button01-highlight-
mouse.blp",
        HighlightAlphaMode "ADD",
    }
    Frame "BACKDROP" "awar_jn_3" {

        Width 0.034,
        Height 0.042,
        BackdropBackground
"ReplaceableTextures\PassiveButtons\PASBTNSscatterRockets.blp",
        BackdropBlendAll,
        SetPoint TOPLEFT, "awar_3", TOPLEFT, 0.040, -0.040,
    }
    Frame "TEXT" "awar_jnname_3" {
        Width 0.12,
        Height 0.1,
        DecorateFileNames,
        FrameFont "MasterFont", 0.015, "",
        FontJustificationH JUSTIFYCENTER,
        UseActiveContext,
        FontJustificationV JUSTIFYMIDDLE,
        LayerStyle "IGNORETRACKEVENTS",
        SetPoint CENTER, "awar_jn_3", CENTER, 0, -0.035,
        Text "弹幕攻击",
    }
    Frame "TEXT" "awar_jnsm_3" {
        Width 0.080,
        Height 0.1,
        DecorateFileNames,
        FrameFont "MasterFont", 0.01, "",
    }
}
```

```
    FontJustificationH JUSTIFYCENTER,
    UseActiveContext,
    FontJustificationV JUSTIFYMIDDLE,
    LayerStyle "IGNORETRACKEVENTS",
    SetPoint CENTER, "awar_jnname_3", CENTER, 0, -0.035,
    Text "对周围的敌方空中单位进行强有力的弹幕攻击",
}
}
}
```

小小演示

构建酒馆与小地图UI显示头像

B站演示

魔兽争霸构建酒馆与小地图UI显示头像

小地图UI显示头像

h-lua已添加此完整功能接口，建议直接使用，不必像下方自己实现

```
--[[  
    绑定一个单位在"小地图"上显示你想要的贴图  
    options = {  
        frequency = 0.05, --刷新频率  
        whichUnit = nil, --某单位  
        texture = {file="",w=0.1,h=0.12,opacity}, --是用的单位标志: 贴图文件、宽、  
高、透明[0.0-1.0]  
        miniMap = {frame=0,w=0,h=0}, -- 小地图大小位置参数: UI-id、宽、高  
        action = function(trackOptions){}, -- 每周期行为  
    }  
]]  
---@param options pilotDZUIMiniMapTrack  
hdzui.miniMapTrack = function(options)  
    if (options.whichUnit == nil or his.deleted(options.whichUnit)) then  
        return  
    end  
    options.frequency = options.frequency or 0.05  
    options.texture = options.texture or {}  
    options.texture.file = options.texture.file or  
hunit.getAvatar(hunit.getId(options.whichUnit))  
    options.texture.w = options.texture.w or 0.016  
    options.texture.h = options.texture.h or 0.016  
    options.texture.opacity = options.texture.opacity or 1.0  
    options.miniMap = options.miniMap or {}  
    options.miniMap.frame = options.miniMap.frame or hdzui.origin.miniMap()  
    options.miniMap.w = options.miniMap.w or 0.13875  
    options.miniMap.h = options.miniMap.h or 0.1375  
    if (type(options.action) ~= "function") then  
        options.action = function( t)  
    end
```

```
-- 设_t.deleted=true则中止追踪
if (his.dead(_t.whichUnit)) then
    hjapi.DzFrameShow(_t.frame, false)
    return
end
hplayer.forEach(function(enumPlayer, _)
    if (enumPlayer == hplayer.loc()) then
        hjapi.DzFrameShow(_t.frame, his.allyPlayer(_t.whichUnit,
enumPlayer))
    end
end)
end
local track = hdzui.frameTag("BACKDROP", "StandardMenuTinyButtonBaseBackdrop",
hdzui.origin.game())
if (track <= 0) then
    return
end
hjapi.DzFrameSetTexture(track, options.texture.file)
hjapi.DzFrameSetAlpha(track, 255 * options.texture.opacity)
hjapi.DzFrameSetSize(track, options.texture.w, options.texture.h)
hjapi.DzFrameShow(track, false)
options.frame = track
options.deleted = false
htime.setInterval(options.frequency, function(curTimer)
    if (options.deleted == true or his.deleted(options.whichUnit)) then
        htime.delTimer(curTimer)
        hjapi.DzFrameShow(options.frame, false)
        hjapi.DzFrameSetEnable(options.frame, false)
        return
    end
    local x = (hunit.x(options.whichUnit) - hrect.getMinX(hrect.world())) /
hrect.getWidth(hrect.world()) * options.miniMap.w
    local y = (hunit.y(options.whichUnit) - hrect.getMinY(hrect.world())) /
hrect.getHeight(hrect.world()) * options.miniMap.h
    hdzui.framePoint(options.frame, options.miniMap.frame, FRAME_ALIGN_CENTER,
FRAME_ALIGN_LEFT_BOTTOM, x, y)
    options.action(options)
end)
end
```

小小演示

伤害排行榜

B站演示

魔兽争霸伤害排行榜

伤害排行榜UI显示

```
UI = function()
    --加载toc
    hdzui.loadToc("UI\\frame.toc")
    --初始化
    awar = hdzui.frame("awar",hdzui.origin.game(),0)
    --设置位置
    hjapi.DzFrameSetAbsolutePoint(awar,4,0.388,0.350)
    hjapi.DzFrameShow(awar,false)

    --伤害榜初始化
    damUI = {}
    txtUI = {}
    barUI = {}
    --logo
    damUI[1] = hdzui.frameTag("BACKDROP","x1",hdzui.origin.game(),"")
    hjapi.DzFrameSetTexture(damUI[1],"war3mapImported\\dam.tga",false)
    hjapi.DzFrameSetAbsolutePoint(damUI[1],4,0.744,0.411)
    hjapi.DzFrameSetSize(damUI[1],0.108,0.026)
    hjapi.DzFrameShow(damUI[1],true)
    --忘记放我可爱的音乐了
    hplayer.forEach(function(enumPlayer, idx)
        --头像
        damUI[idx + 1] = hdzui.frameTag("BACKDROP","tx1" ..
idx,hdzui.origin.game(),"")
        hjapi.DzFrameSetTexture(damUI[idx + 1],"war3mapImported\\tx.tga",false)
        hjapi.DzFrameSetPoint(damUI[idx + 1],2,damUI[1],8,0,0)
        hjapi.DzFrameSetSize(damUI[idx + 1],0.015,0.019)
        hjapi.DzFrameShow(damUI[idx + 1],false)

        --条条
        barUI[idx] = hdzui.frameTag("BACKDROP","txx" .. idx,damUI[idx + 1],"")
        hjapi.DzFrameSetTexture(barUI[idx],"war3mapImported\\green.tga",false)
        hjapi.DzFrameSetPoint(barUI[idx],5,damUI[idx + 1],3,0,0)
        hjapi.DzFrameSetSize(barUI[idx],0.107,0.019)
        hjapi.DzFrameShow(barUI[idx],true)

        --伤害文字
        txtUI[idx] = hdzui.frameTag("TEXT","tx" .. idx,damUI[idx + 1],"")
        hjapi.DzFrameSetText(txtUI[idx],"0")
        hjapi.DzFrameSetPoint(txtUI[idx],5,barUI[idx],3,0,0)
        hjapi.DzFrameSetSize(txtUI[idx],0.030,0.017)
        hjapi.DzFrameShow(txtUI[idx],true)
    end)
    --注册3个按钮点击事件
    hsync.onFrameMouse(hjapi.DzFrameFindByName("awar_1",0),1,function(syncData)
        --添加技能
        hskill.add(hero[syncData.triggerPlayer],tb[syncData.triggerPlayer][1],1)
        if (syncData.triggerPlayer ~= hplayer.loc()) then
            return
        end
        hjapi.DzFrameShow(awar,false)
    end)
    hsync.onFrameMouse(hjapi.DzFrameFindByName("awar_2",0),1,function(syncData)
```

```

--添加技能
hskill.add(hero[syncData.triggerPlayer],tb[syncData.triggerPlayer][2],1)
if (syncData.triggerPlayer ~= hplayer.loc()) then
    return
end
hjapi.DzFrameShow(awar,false)
end)
hsync.onFrameMouse(hjapi.DzFrameFindByName("awar_3",0),1,function(syncData)
--添加技能
hskill.add(hero[syncData.triggerPlayer],tb[syncData.triggerPlayer][3],1)
if (syncData.triggerPlayer ~= hplayer.loc()) then
    return
end
hjapi.DzFrameShow(awar,false)
end)

--然后计时器改变UI位置

htime.setInterval(0.05,function(curTimer)
local sort = {}
hplayer.forEach(function(enumPlayer, idx)
    local s = {idx = idx , val = hplayer.getDamage(enumPlayer)}
    table.insert(sort,s)
end)
--排序
table.sort(sort,function(a, b)
    if (a.val == b.val) then
        return b.idx > a.idx
    end
    return a.val > b.val
end)
--根据排序更改头像UI位置
local dam = sort[1].val
local i = 0
for k,v in ipairs(sort)do
    if (k > 1) then
        --除了排第一的 其他条条根据第一名伤害来确定大小
        i = v.val / dam
        hjapi.DzFrameSetSize(barUI[v.idx],i * 0.107,0.019)
    else
        hjapi.DzFrameSetSize(barUI[v.idx],0.107,0.019)
    end
    --修改伤害文字

hjapi.DzFrameSetText(txtUI[v.idx],math.round(hplayer.getDamage(hplayer.players[v.idx])))
    --修改头像图标
    hjapi.DzFrameSetTexture(damUI[v.idx +
1],hslk.i2v(hunit.getId(hero[v.idx]),"slk","Art"),false)
    --修改头像UI位置
    hjapi.DzFrameSetPoint(damUI[v.idx + 1],2,damUI[1],8,0,(k - 1) * -0.02)
    hjapi.DzFrameShow(damUI[v.idx + 1],true)
end

```

```
    end)  
end
```

高端演示

B站演示

[魔兽争霸lua作图 - 完美动态箭矢模拟实际效果演示](#)

[魔兽争霸lua作图 - 七十二变模型箭矢效果演示](#)

[魔兽争霸lua作图 - hot实时热更新，调试从未如此轻松](#)

[魔兽争霸lua作图 - 技能实时动态选区](#)

[魔兽争霸lua作图 - 非凡技能栏](#)