**井字棋函数用法手册**

**API**

**目录**

[一、GUIFunction 5](#_Toc163502146)

[1. GUIAnimatior 5](#_Toc163502147)

[·GuiAnimator.ShowElement(div) 5](#_Toc163502148)

[·GuiAnimator.HiddenElement(div) 5](#_Toc163502149)

[·GuiAnimator.TransformToMatrix(div) 5](#_Toc163502150)

[2. GameBoardController 5](#_Toc163502151)

[·GameBoardController.SetClickPositionEvent(gameBoard) 5](#_Toc163502152)

[·GameBoardController.ClickEvent(gameBoard) 6](#_Toc163502153)

[·GameBoardController.SetChessWith(position, gameBoard) 6](#_Toc163502154)

[·GameBoardController.ChessDiv(curr = DataBase.currentOrder - 1) 6](#_Toc163502155)

[·GameBoardController.CreateGameBoardLine(div) 6](#_Toc163502156)

[·GameBoardController.InitBoard(gameBoard) 6](#_Toc163502157)

[·GameBoardController.GameEnd() 6](#_Toc163502158)

[3. ScoreBoardController 7](#_Toc163502159)

[·ScoreBoardController.LoadScore(scoreboard = this.scoreboard) 7](#_Toc163502160)

[·ScoreBoardController.ToggleIconHighLight() 7](#_Toc163502161)

[4. SettlementController 7](#_Toc163502162)

[·SettlementController.SetSettlement(div) 7](#_Toc163502163)

[·SettlementController.ShowEndDiv() 7](#_Toc163502164)

[·SettlementController.HiddenEndDiv() 7](#_Toc163502165)

[5. CoinController 8](#_Toc163502166)

[·new CoinController() 8](#_Toc163502167)

[·CoinController.choose 8](#_Toc163502168)

[·CoinController.FixedUpdate() 8](#_Toc163502169)

[·CoinController.CreateMask() 8](#_Toc163502170)

[·CoinController.Destroy() 8](#_Toc163502171)

[二、Icon图标 8](#_Toc163502172)

[1. Unicode 8](#_Toc163502173)

[·Unicode.logo 8](#_Toc163502174)

[·Unicode.arrow\_down 9](#_Toc163502175)

[·Unicode.arrow\_up 9](#_Toc163502176)

[·Unicode.arrow\_left 9](#_Toc163502177)

[·Unicode.arrow\_right 9](#_Toc163502178)

[·Unicode.arrow\_ld 9](#_Toc163502179)

[·Unicode.arrow\_rd 9](#_Toc163502180)

[·Unicode.arrow\_ru 9](#_Toc163502181)

[·Unicode.arrow\_lu 9](#_Toc163502182)

[·Unicode.audio\_play 9](#_Toc163502183)

[·Unicode.audio\_mute 9](#_Toc163502184)

[·Unicode.setting 9](#_Toc163502185)

[·Unicode.reset 10](#_Toc163502186)

[·Unicode.coin 10](#_Toc163502187)

[·Unicode.chess\_cross 10](#_Toc163502188)

[·Unicode.chess\_nought 10](#_Toc163502189)

[三、FX特效 10](#_Toc163502190)

[1. ClickFX 10](#_Toc163502191)

[·ParticleNum 10](#_Toc163502192)

[·JitterSize 10](#_Toc163502193)

[·Color 10](#_Toc163502194)

[·MovingSpeed 10](#_Toc163502195)

[·CollapseVelocity 10](#_Toc163502196)

[·CollapseLimit 10](#_Toc163502197)

[·ClickFX.CreateCanvas() 11](#_Toc163502198)

[·ClickFX.FixedUpdate() 11](#_Toc163502199)

[·new Particle(position) 11](#_Toc163502200)

[·Particle.FixedUpdate() 11](#_Toc163502201)

[·Particle.Draw() 11](#_Toc163502202)

[四、Audio音效 11](#_Toc163502203)

[1. AudioController 11](#_Toc163502204)

[·AudioController.EnableAudio() 11](#_Toc163502205)

[·AudioController.PlaySound(file) 12](#_Toc163502206)

[·AudioController.InitBackGround() 12](#_Toc163502207)

[·AudioController.TryPlayBackGround() 12](#_Toc163502208)

[·MusicFiles.background 12](#_Toc163502209)

[·MusicFiles.book\_page 12](#_Toc163502210)

[·MusicFiles.blacksmith 12](#_Toc163502211)

[·MusicFiles.coin 12](#_Toc163502212)

[·MusicFiles.ui\_click 12](#_Toc163502213)

[五、Storage 12](#_Toc163502214)

[1. DataBase 12](#_Toc163502215)

[·DataBase.isAllowClick 13](#_Toc163502216)

[·DataBase.isAudioPlay 13](#_Toc163502217)

[·DataBase.AIOrder 13](#_Toc163502218)

[·DataBase.gameBoard 13](#_Toc163502219)

[·DataBase.currentOrder 13](#_Toc163502220)

[·DataBase.isGameEnd 13](#_Toc163502221)

[·DataBase.score 13](#_Toc163502222)

[·DataBase.PlaceIsEmpty(position) 13](#_Toc163502223)

[·DataBase.SetChess(position) 13](#_Toc163502224)

[·DataBase.SwitchRounds() 13](#_Toc163502225)

[·DataBase.GameBoardHasEmpty() 14](#_Toc163502226)

[·DataBase.currIsWin(board = this.gameBoard, curr = this.currentOrder) 14](#_Toc163502227)

[·DataBase.CompareDifferences(calcBoard) 14](#_Toc163502228)

[2. Memory 14](#_Toc163502229)

[·Memory.MemSet() 14](#_Toc163502230)

[·Memory.MemLoad() 14](#_Toc163502231)

[·Memory.DataValidation(data) 15](#_Toc163502232)

[·Memory.isSaved 15](#_Toc163502233)

[·Memory.Destroy() 15](#_Toc163502234)

[·Memory.EncryptString(str) 15](#_Toc163502235)

[·Memory.DecryString(encryptedStr) 15](#_Toc163502236)

[六、AIEngine 16](#_Toc163502237)

[·Minimax(node, depth, isMaximizingPlayer, alpha, beta, player) 16](#_Toc163502238)

[·EvaluateBoard(board, player) 16](#_Toc163502239)

[·GetChildren(board, player) 16](#_Toc163502240)

[·IsGameOver(board, player) 17](#_Toc163502241)

[·BestMove(board, player) 17](#_Toc163502242)

[·RotateLeft(board, n) 17](#_Toc163502243)

[七、ToolBox 17](#_Toc163502244)

[1. Time 17](#_Toc163502245)

[·Time.deltaTime 17](#_Toc163502246)

[·Time.fixedTime 18](#_Toc163502247)

[·Time.GenrateUniqueId() 18](#_Toc163502248)

[2. Vector2 18](#_Toc163502249)

[·new Vector2(x, y) 18](#_Toc163502250)

[·Vector2.Normalized 18](#_Toc163502251)

[·Vector2.Distance(a, b = Vector2.zero) 18](#_Toc163502252)

[·Vector2.DotProduct(a, b) 19](#_Toc163502253)

[·Vector2.CrossProduct(a, b) 19](#_Toc163502254)

[·Vector2.Lerp(currentPosition, targetPosition, deltaTime) 19](#_Toc163502255)

[·Vector2.zero 19](#_Toc163502256)

[·Vector2.one 20](#_Toc163502257)

[·Vector2.left 20](#_Toc163502258)

[·Vector2.right 20](#_Toc163502259)

[·Vector2.up 20](#_Toc163502260)

[·Vector2.down 20](#_Toc163502261)

[·Vector2.Add(a, b) 20](#_Toc163502262)

[·Vector2.Sub(a, b) 21](#_Toc163502263)

[·Vector2.Mul(a, b) 21](#_Toc163502264)

[·Vector2.Div(a, b) 21](#_Toc163502265)

[3. UnionFind 21](#_Toc163502266)

[·UnionFind.find(x) 21](#_Toc163502267)

[·UnionFind.union(x, y) 21](#_Toc163502268)

[·UnionFind.connect(x, y) 21](#_Toc163502269)

[4. RandomMap 22](#_Toc163502270)

[·RandomMap.mapValuesToString(map) 22](#_Toc163502271)

[·RandomMap.stringToMapValues(str) 22](#_Toc163502272)

[·RandomMap.CreateRandomMapping() 22](#_Toc163502273)

一、GUIFunction

1. GUIAnimatior

用于动画控制。

·GuiAnimator.ShowElement(div)

使用高斯模糊显示div

@param {HTMLDivElement} div 要显示的div

·GuiAnimator.HiddenElement(div)

使用高斯模糊隐藏div

@param {HTMLDivElement} div 要隐藏的div

·GuiAnimator.TransformToMatrix(div)

将div里的transform属性解析为matrix

@param {HTMLDivElement} div 所需div

@return {Number[]} 矩阵

2. GameBoardController

　　用于控制棋盘内的相关逻辑。

·GameBoardController.SetClickPositionEvent(gameBoard)

在div上部署监听点击事件，以执行游戏更新

@param {HTMLDivElement} gameBoard

·GameBoardController.ClickEvent(gameBoard)

监听事件

@param {HTMLDivElement} gameBoard

·GameBoardController.SetChessWith(position, gameBoard)

在目标点创建棋子

@param {Vector2} position

@param {HTMLDivElement} gameBoard

·GameBoardController.ChessDiv(curr = DataBase.currentOrder - 1)

渲染棋子

@param {number} curr 默认渲染当前回合类型的棋子

@return {HTMLDivElement}

·GameBoardController.CreateGameBoardLine(div)

创建棋盘线

@param {HTMLDivElement} div 创建位置

·GameBoardController.InitBoard(gameBoard)

从存储中加载

@param {HTMLDivElement} gameBoard

·GameBoardController.GameEnd()

游戏结束重置数据库的逻辑

3. ScoreBoardController

　　用于控制分数相关逻辑。

·ScoreBoardController.LoadScore(scoreboard = this.scoreboard)

加载分数

@param {HTMLDivElement} scoreboard 第一次需要传参

·ScoreBoardController.ToggleIconHighLight()

切换回合强调

4. SettlementController

　　用于显示游戏结束提示

·SettlementController.SetSettlement(div)

设置Settlement的dom

@param {HTMLDivElement} div

·SettlementController.ShowEndDiv()

显示结算条

@param {Number} winner 胜利方，0平局、1先方、2后方

·SettlementController.HiddenEndDiv()

隐藏结算条

5. CoinController

　　硬币实例控制。

·new CoinController()

自动生成硬币转动，进行选择。

·CoinController.choose

选择的先后手，2 or 1

* 以下函数自动执行

·CoinController.FixedUpdate()

每Time.fixedTime执行一次。

·CoinController.CreateMask()

创建用于禁用点击的遮罩层。

·CoinController.Destroy()

删除硬币和遮罩层。

二、Icon图标

1. Unicode

　　使用MesloLGS NF Regular.ttf字体做图标

·Unicode.logo

LOGO

* 箭头相关

·Unicode.arrow\_down

下箭头

·Unicode.arrow\_up

上箭头

·Unicode.arrow\_left

左箭头

·Unicode.arrow\_right

右箭头

·Unicode.arrow\_ld

左下

·Unicode.arrow\_rd

右下

·Unicode.arrow\_ru

右上

·Unicode.arrow\_lu

左上

* 功能

·Unicode.audio\_play

播放声音

·Unicode.audio\_mute

静音

·Unicode.setting

设置

·Unicode.reset

重置

·Unicode.coin

硬币

·Unicode.chess\_cross

先手方

·Unicode.chess\_nought

后手方

三、FX特效

1. ClickFX

　　鼠标点击特效。

·ParticleNum

粒子数量，默认15

·JitterSize

粒子抖动大小（最大尺寸），默认5

·Color

粒子色彩，默认#FFCC00

·MovingSpeed

粒子移动速度，默认3

·CollapseVelocity

粒子缩小速度，默认0.1

·CollapseLimit

粒子消失界限（最小尺寸），默认0.2

* 以下函数自动执行

·ClickFX.CreateCanvas()

绘制画布

·ClickFX.FixedUpdate()

每Time.fixedTime执行一次

·new Particle(position)

在position处创建粒子

@param {Vector2} position

·Particle.FixedUpdate()

每Time.fixedTime执行一次

·Particle.Draw()

绘制粒子

四、Audio音效

1. AudioController

　　音效控制相关

·AudioController.EnableAudio()

使浏览器允许播放声音。

·AudioController.PlaySound(file)

播放音频文件

@param {string} file 从MusicFiles中选择

·AudioController.InitBackGround()

初始化背景音乐

·AudioController.TryPlayBackGround()

尝试播放背景音乐

·MusicFiles.background

背景音乐

·MusicFiles.book\_page

翻书音

·MusicFiles.blacksmith

打铁音

·MusicFiles.coin

硬币音

·MusicFiles.ui\_click

UI点击音

五、Storage

1. DataBase

　　缓存数据库，可视为内存。

·DataBase.isAllowClick

是否允许点击棋盘

·DataBase.isAudioPlay

是否允许播放声音

·DataBase.AIOrder

0表示没有AI，1为AI先手，2为AI后手

·DataBase.gameBoard

棋盘 Number[3][3]

·DataBase.currentOrder

当前回合，1为先手方，2为后手方

·DataBase.isGameEnd

游戏结束标记

·DataBase.score

比分 Number[2]

·DataBase.PlaceIsEmpty(position)

检查棋盘目标格是否为空

@param {Vector2} position

@return {boolean}

·DataBase.SetChess(position)

将插入位置写入gameBoard

@param {Vector2} position

·DataBase.SwitchRounds()

切换先后手

·DataBase.GameBoardHasEmpty()

检查棋盘上是否还有空位

@return {boolean}

·DataBase.currIsWin(board = this.gameBoard, curr = this.currentOrder)

检查某一方是否胜利

@param {Number} curr 默认为当前方

@param {Number[][]} board 默认为棋盘

@return {boolean}

·DataBase.CompareDifferences(calcBoard)

查找棋盘与当前棋盘的差异（AI用）

@param {number[][]} calcBoard

@return {Vector2}

2. Memory

　　持久化存储数据。以类似消息队列模式存储数据。

·Memory.MemSet()

将缓存数据库写入存储

·Memory.MemLoad()

将存储的数据读入缓存

·Memory.DataValidation(data)

数据校验

@param {{}} data

@return {boolean}

·Memory.isSaved

检查是否有存储内容

·Memory.Destroy()

清除存储内容

·Memory.EncryptString(str)

加密buffer

@param {string} str

@return {string}

·Memory.DecryString(encryptedStr)

解密buffer

@param {string} encryptedStr

@return {string}

六、AIEngine

AI相关逻辑，采用极大极小树方法进行判别。

·Minimax(node, depth, isMaximizingPlayer, alpha, beta, player)

极大极小树

@param {number[][]} node 节点

@param {number} depth 遍历深度

@param {boolean} isMaximizingPlayer 是否为极大化

@param {number} alpha

@param {number} beta

@param {number} player 需要判别的先后方，1或2

@return {number} 分数

·EvaluateBoard(board, player)

计算分数

@param {number[][]} board 判别盘面

@param {number} player 需要判别的先后方，1或2

@return {number} 分数

·GetChildren(board, player)

获取所有可能的子盘

@param {number[][]} board 所需盘面

@param {number} player 下一步的先后方，1或2

@return {[number[][]]} 所有下一步可以走的盘面

·IsGameOver(board, player)

判断游戏是否结束

@param {Number[][]} board 判断盘面

@param {number} player

@return {boolean}

·BestMove(board, player)

供外部调用，计算最佳落子

@param {number[][]} board 当前盘面

@param player 需要计算的先后方，1或2

@return {number[][]} 最佳位置的盘面

·RotateLeft(board, n)

左旋整个棋盘

@param {number[][]} board

@param {number} n 旋转次数

@return {number[][]} 旋转后的棋盘

七、ToolBox

1. Time

　　时间相关。

·Time.deltaTime

每一帧的更新时间（毫秒）

·Time.fixedTime

固定更新时间，默认20毫秒（用于物理效果更新）

·Time.GenrateUniqueId()

生成随机ID

@returns {string}

2. Vector2

　　二维向量

·new Vector2(x, y)

创建二维向量

@param {number} x

@param {number} y

·Vector2.Normalized

向量归一化

@returns {Vector2}

·Vector2.Distance(a, b = Vector2.zero)

计算两点间的距离

@param {Vector2} a

@param {Vector2} b

@returns {number}

·Vector2.DotProduct(a, b)

计算点积:

@param {Vector2} a

@param {Vector2} b

@returns {number}

·Vector2.CrossProduct(a, b)

计算叉乘:

@param {Vector2} a

@param {Vector2} b

@returns {number}

·Vector2.Lerp(currentPosition, targetPosition, deltaTime)

向目标点线性插值(delta)

@param {Vector2} currentPosition 当前坐标

@param {Vector2} targetPosition 目标坐标

@param {number} deltaTime 毫秒数

@returns {Vector2}

* 构造函数

·Vector2.zero

Vector2 (0, 0)

@returns {Vector2}

·Vector2.one

Vector2 (1, 1)

@returns {Vector2}

·Vector2.left

Vector2 (-1, 0)

@returns {Vector2}

·Vector2.right

Vector2 (1, 0)

@returns {Vector2}

·Vector2.up

Vector2 (0, -1)

@returns {Vector2}

·Vector2.down

Vector2 (0, 1)

@returns {Vector2}

* 向量运算

·Vector2.Add(a, b)

坐标相加

@param {Vector2} a

@param {Vector2} b

@returns {Vector2}

·Vector2.Sub(a, b)

坐标相减

@param {Vector2} a

@param {Vector2} b

@returns {Vector2}

·Vector2.Mul(a, b)

向量乘值

@param {Vector2} v Vector2

@param {number} n Number

@returns {Vector2}

·Vector2.Div(a, b)

向量除值

@param {Vector2} v Vector2

@param {number} n Number

@returns {Vector2}

3. UnionFind

　　并查集(Disjoint Set Union, DSU)

·UnionFind.find(x)

·UnionFind.union(x, y)

·UnionFind.connect(x, y)

4. RandomMap

随机映射表（存储加密用，将[0-9]随机映射到[0-9]上）

·RandomMap.mapValuesToString(map)

map to string

@param {Map} map

@return {string}

·RandomMap.stringToMapValues(str)

string to map

@param {string} str

@return {Map}

·RandomMap.CreateRandomMapping()

创建一个[0-9]的随机映射表

@return {Map}