Duelyst API Documentation

Team Name: BetterGrade
Name Guid:
Zumin Li 2278100l
Linyue Zhang 2587334z
Keyi Miao 2544119k
Yifan Liu 2582395l
Yadi Tian 2546988t

● Class: Tile

Attribute

Name	Description	Type	Visibility	Default Value	Setter/Getter
tileState	The state of this tile	TileState	private	TileState.NORMAL	getTileState setTileState
unitOnTile	Show the unit on this tile	Unit	private	/	getUnitOnTile setUnitOnTile
moveableTiles	Store all the moveable tiles of the selected unit	HashSet <Tile>	private	/	getMoveableTiles

Method

Name	TileState Show the color of a tile			
Type	enum			
Visibility	default			
Value	**Value**	**Name**	**Mode**	**Description**
	NORMAL	normal	0	The color of tile is grey
	WHITE	white	1	1. Show the movable tile of selected friendly unit 2. Show the placeable tile of selected card 3. Show the target unit of selected spell
	RED	red	2	Show the attackable enemy unit of selected friendly unit
Name	Trigger Receive the broadcast and make corresponding actions			
Visibility	public			
Parameters	Class target Map<String, Object> parameters			
Return	/			
Parameters Type	**Other parameter**	**Value** **Y-compulsory** **N-optional**	**Description**	
searchUnit	range	enemy	Search all the enemy units, highlight white	
		all	Search all units, highlight white	

		non_avatar	Search all non-avatar units, highlight white
		your_avatar	Search the avatar of current player, highlight white
		all_friends	Search all units belong to AI Player, highlight white
validSummonRangeHighlight	airdrop	activate	For the card which can be summoned anywhere on the board, highlight white
		N	Calculate the tiles which should be highlight
validSummonRangeHighlight-checkNeighbour	tilex	Y	If there is no unit on tile, highlight the tile
	tiley	Y	
textureReset	/	/	Clear all the highlight
summon	tilex	Y	Sommon a unit on this tile
	tiley	Y	
firstClickTile	tilex	Y	When a player click a tile, if there is a unit on tile, set the state of unit, and highlight the movable and attackable targets of this unit
	tiley	Y	
moveHighlight	tilex	Y	If the tile is movable of the selected tile, highlight it to white
	tiley	Y	
attackHighlight	tilex	Y	If the tile is attackable of the selected tile, highlight it to red
	tiley	Y	
deleteUnit	tilex	Y	Delete the unit on this tile
	tiley	Y	
spell	tilex	Y	Play a spell card
	tiley	Y	
rangedUnitAttackHighlight	/	/	For the unit which can process ranged attack, highlight the target to red
operateUnit	tilex	Y	Operate the unit according to the selected tile state Only move - tile state is WHITE Only attack - tile state is RED and in attackable range Move and attack - tile state is RED and out of attackable range
	tiley	Y	
	originTileSelected	Y	
AI_FindOperateTile	/	/	For AI, find the tile which can operate
checkMoveVertically	tilex	Y	Check whether the unit need to move vertically first or horizontally
	tiley	Y	
	originTile	Y	
	AimTile	Y	

Name	moveHighlight Show the movable tiles and highlight it
Visibility	private
Parameters	Int count - record the times of the unit move, 0 - first step
Return	/

Name	attackHighlight Show the movable tiles and highlight it
Visibility	private
Parameters	/
Return	/

Name	attackedBroadcast

	Attack a unit
Visibility	private
Parameters	Unit attackerUnit - the attacker unit
Return	/
Name	move Move a unit to a tile
Visibility	private
Parameters	Unit unit - the unit needs to move Tile originTile - the unit’s original tile Boolean mode - decide horizontally move first, or vertically, true - vertically false - horizontally
Return	/
Name	distanceOfTiles Calculate the two tile’s square of distance
Visibility	private
Parameters	Tile tile1 - first tile Tile tile2 - second tile
Return	Int distance - two tile’s square of distance
Name	resetTileSelected Clear the highlight and reset the game state to READY
Visibility	private
Parameters	Tile tile1 - first tile Tile tile2 - second tile
Return	Int distance - two tile’s square of distance
Name	allBroadcast Broadcast to all tile object
Visibility	private
Parameters	String type
Return	/
Name	checkMoveVertically Generate a broadcast to check whether the unit need to move vertically first or horizontally
Visibility	private
Parameters	Tile originTile - the original selected tile
Return	/
Name	AdjacentBroadcast
Visibility	private
Parameters	String type
Return	/

- ## Class: Unit

Attribute

Name	Description	Type	Visibility	Default Value	Setter/Getter
maxHealth	The maximum health of unit, equals to the card's health	int	private	/	setMaxHealth()
currentState	The current state of unit	UnitState	private	/	setCurrentState()

owner	The unit's owner, human or AI player	Player	private	GameState.getInstance().getCurrentPlayer()	getOwner()
					setOwner()

attack	The unit's attack	int	private	0	getAttack()
					setAttack()

health	The unit's health	int	private	0	getHealth()
					setHealth()

rangedAttack	Decide whether a unit can ranged attack	boolean	default	false	
flying	Decide whether a unit can move to anywhere	boolean	default	false	
maxAttackNum	The maximum number of unit's attack	int	default	1	
maxMoveNum	The maximum number of unit's move	int	default	1	

attackNum	The number of unit's attack	int	default	1	getAttackNum()
					setAttackNum()

moveNum	The number of unit's move	int	default	1	getMoveNum()
					setMoveNum()

canProvoke	Decide whether a unit can provoke	boolean	default	false	getCanProvoke()
					setCanProvoke()

isProvoked	Decide whether a unit is provoked	boolean	default	false	isProvoked()
					setProvoked()

Method

<table>
<tr><td><b>Name</b></td><td colspan="2">UnitState
Show the state of a unit</td></tr>
<tr><td><b>Type</b></td><td colspan="2">enum</td></tr>
<tr><td><b>Visibility</b></td><td colspan="2">public</td></tr>
<tr><td rowspan="5"><b>Value</b></td><td><b>Value</b></td><td><b>Description</b></td></tr>
<tr><td>NOT_READY</td><td>When a unit is firstly summoned</td></tr>
<tr><td>READT</td><td>When a new turn starts, set every unit to READY</td></tr>
<tr><td>HAS_MOVED</td><td>After a unit move</td></tr>
<tr><td>HAS_ATTACKED</td><td>After a unit attack</td></tr>
</table>

Name	changeHealth Change the health of unit

Visibility	public
Parameters	Tile originTile - the original selected tile
Return	/

Name	changeAttack Change the attack of unit
Visibility	public
Parameters	Tile originTile - the original selected tile
Return	/

Name	Trigger Receive the broadcast and make corresponding actions
Visibility	public
Parameters	Class target Map<String, Object> parameters
Return	/

Parameters Type	**Other parameter**	**Value (Y-compulsory N-optional)**	**Description**
unitBeReady	/	/	Set the current player's friendly unit state READY, and set the enemy unit NOT_READY
attacked	attackedUnit	Y	Unit be attacked
modifyUnit	unitId	Y	Modify the health and attack of a unit
	health	Y	
	attack	Y	
	Limit	N	

Name	displayAttackAndHealth Display the attack and health in the front-end
Visibility	protected
Parameters	/
Return	/

Name	attacked
Visibility	private
Parameters	Unit attacker - the unit who launch attack Boolean allowCounterAttack – whether the unit is allowed to counter attack
Return	/

Name	targetIsInAttackRange
Visibility	private
Parameters	int tilex – the tilex of target int tiley – the tiley of target
Return	/

- ## Class: Player

Attribute

Name	Description	Type	Visibility	Default Value	Setter/Getter
deck	The deck of player	<Card> ArrayList	private	null	setDect
cardsOnHand	The cards in player's hand	Card[]	protected	new Card[6]	
health	The health of player	int	default	20	getHealth setHealth
mana	The player owns mana	int	default	0	getMana setMana

Method

Name	isHumanOrAI Get this player is human or AI player
Visibility	public
Parameters	/
Return	True - is human player False - is AI player
Name	drawCard Player draw card to its hand cards
Visibility	public
Parameters	/
Return	/
Name	removeCardFromHand After play the card, clear the the card from hand
Visibility	protected
Parameters	Card card - the discarded card
Return	/
Name	cardSelected Player click a card
Visibility	public
Parameters	Int handPosition - the card in the hand's position
Return	/
Name	clearSelected Clear the card selected
Visibility	public
Parameters	/
Return	/
Name	showValidRange

	Show the selected card can be place to which tile
Visibility	protected
Parameters	Card cardSelected - the selected card
Return	/

- ## Class: AIPlayer

Inherit to Player

Attribute

Name	Description	Type	Visibility	Default Value	Setter/Getter
optionalTiles	Store all the friendly unit's tile of AI player	<Tile> HashSet	default	null	/
whiteTileGroup	Store all the white tile	<Tile> HashSet	default	null	/
redTileGroup	Store all the red tile	<Tile> HashSet	default	null	/

Method

Name	addToOptionalTile Add the tile to optional tile
Visibility	public
Parameters	Tile tile
Return	/
Name	addToWhiteGroup Store the white tile
Visibility	public
Parameters	Tile tile
Return	/
Name	addToRedGroup Store the red tile
Visibility	public
Parameters	Tile tile
Return	/
Name	startUpAIMode Start the AI
Visibility	public
Parameters	/
Return	/

- ## Class: GameState

Attribute

Name	Description	Type	Visibility	Default Value	Setter/Getter
cardSelectedCallbacks	A map stores a series of callbackcs(Function) called after selecting cards.	Map<String, Function>	private	null	getCardSelectedCallbacks()
beforeSummonCallbacks	A map stores a series of callbackcs(Function) called before a unit was summoned.	Map<String, Function>	private	null	getBeforeSummonCallbacks()
avatarAttackCallbacks	A map stores a series of callbackcs(Function) called when an avatar attack.	Map<String, Function>	private	null	getAvatarAttackCallbacks()
unitDeathCallbacks	A map stores a series of callbackcs(Function) called when a unit is dead.	Map<String, Function>	private	null	getUnitDeathCallbacks()
spellCastCallbacks	A map stores a series of callbackcs(Function) called when cast a spell.	Map<String, Function>	private	null	getSpellCastCallbacks()
turnCount	The count of turn	int	private	0	/
playerContainers	Contains the players of human and AI	Player[]	private	new Player[2]	getPlayerContainers()
currentPlayer	Record the current player	Player	private	null	getCurrentPlayer()

currentState	Show the current game state	CurrentState	private	CurrentState.READY	getCurrentState()
					setCurrentState()

cardSelected	Record the selected card, when this attribute be set null, game state switch to READY	Card	private	null	setCardSelected()
					getCardSelected()

tileSelected	Record the selected tile, when this attribute be set null, game state switch to READY	Tile	private	null	setTileSelected getTileSelected
out	Send message to the front-end	ActorRef	private	null	setOut getOut
instance	The singleton object of Gamestate.	GameState	private	New GameState()	getInstance()

Method

Name	CurrentState Show the current game state	
Type	enum	
Visibility	public	
Value	Value	Description
	READY	The initial state of the game

	CARD_SELECT	Select a card
	UNIT_SELECT	Select a unit

Name	addPlayers Add player to player container
Visibility	public
Parameters	Player humanPlayer Player AIPlayer
Return	/

Name	switchPlayer Switch current player, clear all state
Visibility	public
Parameters	/
Return	/

Name	clear Clear the game state
Visibility	public
Parameters	/
Return	/

Name	broadcastEvent Sent the event to class
Visibility	public
Parameters	Class target - the target class Map<String,Object> parameters - transmitted broadcast
Return	/

Name	registerCallbacks
Visibility	public
Parameters	/
Return	/

● Class: ToolBox

Attribute

Name	Description	Type	Visibility	Default Value	Setter/Getter
humanAvatarId	Unified ID of humanAvatar	int	public	99	/
AIAvatarID	Unified ID of AIAvatar	int	public	100	/
delay	Unified delay for thread delay	int	public	500	/

Method

Name	logNotification Display tips for human player
Visibility	public
Parameters	String message
Return	/
Name	findObjectInArray Find an object in array.
Visibility	public
Parameters	T [] _6Elements, T element
Return	/
Name	currentPlayerName return the current player name(human or AI)
Visibility	public
Parameters	/
Return	/
Name	getRules
Visibility	public
Parameters	Card card
Return	rules
Name	findNumberInStr
Visibility	public
Parameters	String str
Return	value

● Class: Card

Attribute

Name	Description	Type	Visibility	Default Value	Setter/Getter

id	ID of card	int	public	/	getId()
					setId()

cardname	name of card	String	public	/	getCardname()
					setCardname()

manacost	Cost of mana	int	public	/	getManacost()
					setManacost()

miniCard	Minicard resource	MiniCard	public	/	getMiniCard()
					setMiniCard()

bigCard	Bigcard resouce	BigCard	public	/	getBigCard()
					setBigCard()

Method

Name	cardToUnit
Visibility	public
Parameters	/
Return	/
Name	isCreatureOrSpell judge whether a card is spell or unit
Visibility	public
Parameters	/
Return	-1 - this card is spell ; 1 - this card is a unit
Name	creatureCardUsed When the creature card is going to use, call this method
Visibility	public
Parameters	int tilex, int tiley
Return	/