

Player.CS

Properties:

Private hand : *Hand*

Reference to the *Hand* component of this gameobject.

Private board : *Board*

Reference to the *Board* gameobject in the scene.

Private deck : *Deck*

Reference to the *Deck* gameobject in the scene.

Serialized ally : *Player*

Reference to the *Player* gameobject this is allied to.

Private hasAlly : *bool*

Property that keeps track of whether the player has an ally. Intended to gate calls to GetAlly().

Public isAttacking : *bool*

Keeps track of whether the player is attacking this turn.

Public isDefending : *bool*

Keeps track of whether the player is defending this turn.

Public hasEndedTurn : *bool*

Keeps track of whether the player has finished playing cards for the turn.

Private enterPlayHandler : *EnterPlayHandler*

Reference to the *EnterPlayHandler* gameobject in the scene.

Private gameState : *GameState*

Reference to the *GameState* gameobject in the scene.

Private isAI : *bool*

Keeps track of whether the player is controlled by an AI. Intended to gate calls to GetAI().

Private ai : *AI*

Reference to the *AI* component of this gameobject.

Private isThinking : *bool*

Keeps track of whether the *AI* component is “thinking”.

Private statusUpdater : *StatusUpdater*

Reference to the *StatusUpdater* component of this gameobject.

Serialized playerName : *string*

Name to be displayed during game play.

Private audioSource : *AudioSource*

Reference to the *AudioSource* component of this gameobject. The associated *AudioClip* will be played whenever this player plays a *Card*.

Lifecycle Methods:

Awake:

The Awake() method is responsible for finding all the necessary references and initializing any gate-keeping properties related to those references.

Start:

Subscribes to the *EnterPlayHandler*.

Methods

Public GetHand

Parameters: None

Return: *Hand*

Returns a reference to the hand component of this object.

Public DrawACard

Parameters: None

Return: None

Draws a card from the deck object to the hand component of this object.

Public DrawCards

Parameters:

number : Int

Number of cards to be drawn.

Return: None

Draws multiple cards from the deck object to the hand component of this object

Private PlayCard

Parameters:

card : Card

Card that will be played.

Return: None

Moves card from the hand component of this object to the board object.

Public AttackWithCard

Parameters:

card : Card

Card that will be used to attack with.

Return: None

Public DefendWithCard

Parameters:

cardInHand : Card

Card that player will be defending with.

cardOnBoard : Card

Card that player will be defending against.

Return: None

Public GetAlly

Parameters: None

Return: *Player*

Returns a reference to the player object that this player is allied with. This function will return null if this player is not allied to another other player object.

Public EndTurn

Parameters: None

Return: None

Sets this player object ready to end the turn. Turn ending control is handled by *GameState* object.

Public IsAI

Parameters: None

Return: *bool*

Returns the value of the isAI property;

Public HasAlly

Parameters: None

Return: *bool*

Returns the value of the hasAlly property.

Public UpdateThinkingStatus

Parameters:

value : bool

Value that the property isThinking will take on.

Return: None

Updates the `isThinking` property.

Public `GetThinkingStatus`

Parameters: None

Return: *bool*

Returns the `isThinking` property.

Public `GetStatusUpdater`

Parameters: None

Return: *StatusUpdater*

Returns a reference to the `statusUpdater` component of this object.

Public `GetName`

Parameters: None

Return: *string*

Returns the `playerName` property.

Scene Settings

This component assumes that there a *Board* gameobject, a *Deck* gameobject, a *GameState* gameobject, and an *EnterPlayHandler* gameobject in the scene.