

AI.cs

A basic AI that can play Durak. It always attacks if able and always defends if able. It chooses which card to attack or defend with randomly.

Properties

Private player : *Player*

Reference to the *Player* component of this gameobject.

Private board : *Board*

Reference to the *Board* gameobject in the scene.

Private hand : *Hand*

Reference to the *Hand* component of this gameobject.

Serialized maxThinkingTime : *float*

The maximum amount of time that the AI can spend thinking on any action.

Serialized minThinkingTime : *float*

The minimum amount of time that the AI will spend thinking on any action.

Lifecycle Methods

Awake:

Gets references for the player, board, and hand properties.

Methods

Private Reevaluate

Parameters: None

Return: None

After waiting for a random amount of time between minThinkingTime and maxThinkingTime, the AI evaluates the board state and decides what card(s) to play.

Private DefendAgainstBoard

Parameters:

playableCards : *List<Card>*

List of playable cards in the player's hand.

Return: None

This function causes the AI to attempt to defend against every attacking card on board.

Private PickRandomCardFromList

Parameters:

cards:: List<cCard>

List of cards to pick from.

Return: *Card*

Returns the card that was picked.

This function picks a card at random from the list.

Private GetDefendingCards

Parameters:

playableCards : List<Card>

List of playable cards in hand.

attackingCard : Card

card that is being defended against.

Return: *List<card>*

Returns the list of cards that can defend against the attacking card.

Private PlayCardInFirstSlot

Parameters:

card : Card

Return: None

This function plays the card in the first cardslot without another card in it.

Private FindCardToAttackWith

Parameters:

playableCards : List<Card>

List of playable cards in hand

Return: *Card*

Returns the chosen card to attack with.

Scene Settings:

This class expects there to be a *Board* gameobject in the scene.