

**Table of contents**

[Class: Node 3](#_Toc163120569)

[Thuộc tính 3](#_Toc163120570)

[Phương thức 4](#_Toc163120571)

[Class: GameDeterminedTree 7](#_Toc163120572)

[Thuộc tính 7](#_Toc163120573)

[Phương thức 7](#_Toc163120574)

[Class: GameActor 8](#_Toc163120575)

[Thuộc tính 8](#_Toc163120576)

[Phương thức 8](#_Toc163120577)

[Class: Bot 9](#_Toc163120578)

[Thuộc tính 9](#_Toc163120579)

[Phương thức 9](#_Toc163120580)

[Mẫu: 13](#_Toc163120581)

[Thuộc tính 13](#_Toc163120582)

[Phương thức 13](#_Toc163120583)

# Class: Node

* Mô tả: Đại diện cho một node trên cây định trị trò chơi.
* Ràng buộc: Không.

## Thuộc tính

* value
  + Mô tả: Đại diện cho số lượng sỏi của kho qua các lượt đi khác nhau.
  + Kiểu dữ liệu: int
  + Ràng buộc: Có giá trị nhỏ hơn số lượng sỏi khởi đầu của trò chơi và lớn hơn 0.
* determineValue
  + Mô tả: Giá trị định trị của một node.
  + Kiểu dữ liệu: int
  + Ràng buộc: Giá trị 0 thể hiện nước đi mà bot có khả năng thắng cao, giá trị 1 thể hiện nước đi mà khả năng thắng của bot thấp.
* children
  + Mô tả: Đại diện cho các node con của node hiện tại.
  + Kiểu dữ liệu: List<Node>
  + Ràng buộc: Mỗi node chứa tối đa 3 node con.
* isLeaf
  + Mô tả: Thể hiện một node có phải là node lá hay không.
  + Kiểu dữ liệu: bool
  + Ràng buộc: Một node là node lá khi thuộc tính value của nó có giá trị từ 1 đến 3.
* playerTurn
  + Mô tả: Thể hiện node này đại diện cho lượt đi của ai.
  + Kiểu dữ liệu: int
  + Ràng buộc: Thể hiện rằng node hiện tại đại diện cho lượt đi của ai. Là 1 thì đại diện cho lượt đi game actor 1, 2 tương tự.
* parent
  + Mô tả: Đại diện cho node cha của node hiện tại.
  + Kiểu dữ liệu: Node
  + Ràng buộc: Giá trị parent của node gốc là null.

## Phương thức

* Node (constructor)
  + Mục đích: Thiết lập giá trị thuộc tính value khi khởi tạo một node, khi vừa tạo thì node mặt định là node lá.
  + Kiểu trả về: Không có.
  + Tham số truyền vào:
    - value

Mô tả: Giá trị đại diện cho số lượng sỏi trong kho.

Kiểu dữ liệu: int

Ràng buộc: Lớn hơn 0 và nhỏ hơn giá trị sỏi khởi tạo của kho.

* + Giá trị trả về: Không có.
  + Tiền điều kiện: Không có.
  + Hậu điều kiện: Node được tạo ra có giá trị của value bằng với tham số truyền vào.
  + Các thuộc tính được gọi/ được dùng: value.
  + Các phương thức được gọi: Không có.
  + Quá trình xử lý: Gán thuộc tính value bằng giá trị của tham số được truyền vào, gán giá trị thuộc tính isLeaf bằng true, khời tạo thuộc tính children là một List rỗng.
* SetPlayerTurn
  + Mục đích: Thiết lập giá trị thuộc tính playerTurn cho node.
  + Kiểu trả về: Không có.
  + Tham số truyền vào:
    - turn

Mô tả: Giá trị sẽ được gán cho playerTurn.

Kiểu dữ liệu: int

Ràng buộc: Giá trị 1 hoặc 2.

* + Giá trị trả về: Không có.
  + Tiền điều kiện: Không có.
  + Hậu điều kiện: Thuộc tính player của node hiện tại có giá trị như được gán.
  + Các thuộc tính được gọi/ được dùng: playerTurn.
  + Các phương thức được gọi: Không có.
  + Quá trình xử lý: Gán giá trị của tham số value cho thuộc tính playerTurn của node hiện tại.
* SetDeterminedValueForLeaf
  + Mục đích: Thiết lập giá trị determinedValue cho node là node lá.
  + Kiểu trả về: Không có.
  + Tham số truyền vào: Không có.
  + Giá trị trả về: Không có.
  + Tiền điều kiện: Không có.
  + Hậu điều kiện: Thuộc tính determinedValue của node có giá trị phù hợp.
  + Các thuộc tính được gọi/ được dùng: determinedValue, playerTurn.
  + Các phương thức được gọi: Không có.
  + Quá trình xử lý: Nếu thuộc tính playerTurn bằng 1 thì gán giá trị cho determinedValue là 1, ngược lại gán cho nó giá trị 0.
* SetDeterminedValue
  + Mục đích: Thiết lập giá trị của thuộc tính determinedValue ch một node không phải node lá.
  + Kiểu trả về: Không có.
  + Tham số truyền vào:
    - value

Mô tả: Giá trị sẽ được gán cho thuộc tính determinedValue.

Kiểu dữ liệu: int

Ràng buộc: Chỉ chấp nhân giá trị 1 hoặc 0.

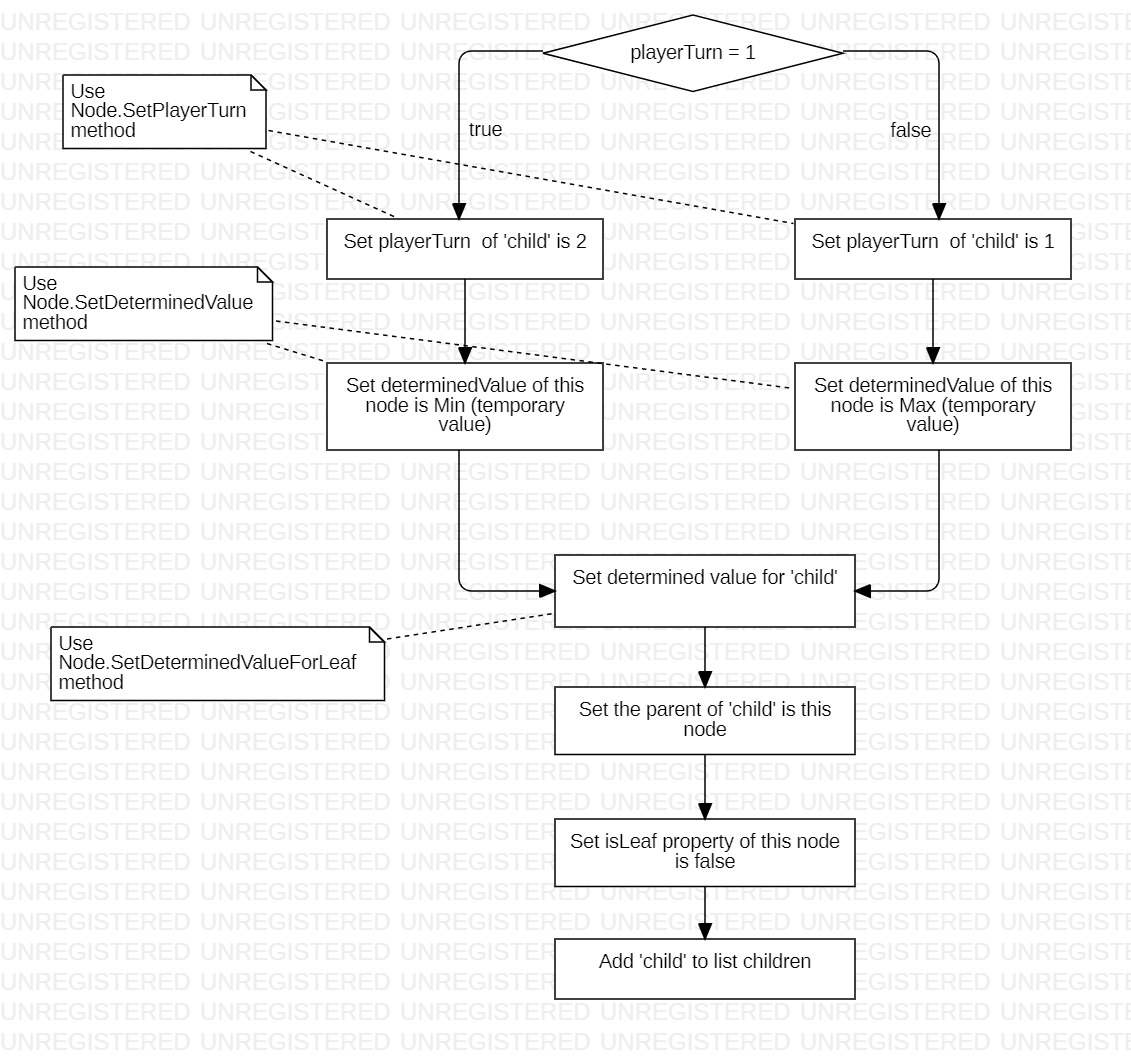
* + Giá trị trả về: Không có.
  + Tiền điều kiện: Không có.
  + Hậu điều kiện: Gái trị của determinedValue có giá trị bằng tham số được truyền vào.
  + Các thuộc tính được gọi/ được dùng: determinedValue.
  + Các phương thức được gọi: Không có.
  + Quá trình xử lý: Gán giá trị của thuộc tính determinedValue bằng giá trị của tham số truyền vào.
* AddChild
  + Mục đích: Thêm node con cho node hiện tại.
  + Kiểu trả về: Không có.
  + Tham số truyền vào:
    - child

Mô tả: node con được thêm.

Kiểu dữ liệu: Node.

Ràng buộc: Có giá trị của thuộc tính value chênh lệch với giá trị value của node hiện tại từ 1 đến 3.

* + Giá trị trả về: Không có.
  + Tiền điều kiện: Node hiện tại thuộc tính value lớn hơn 3.
  + Hậu điều kiện: Thuộc tính children của node hiện tại có thêm 1 phần tử mới.
  + Các thuộc tính được gọi/ được dùng: children, playerTurn, determinedValue, isLeaf, parent.
  + Các phương thức được gọi: Node.SetPlayerTurn, Node.SetDeterminedValueForLeaf, Node.SetDeterminedValue.
  + Lưu đồ xử lý:



# Class: GameDeterminedTree

## Thuộc tính

## Phương thức

# Class: GameActor

* Mô tả: Biểu diễn cho thực thể chịu trách nhiệm chơi trò chơi, có thể là người chơi hoặc máy.
* Ràng buộc: Kế thừa lớp MonoBehaviour của Unity.

## Thuộc tính

* isInTurn
  + Mô tả: thể hiện trạng thái trong lượt của người chơi.
  + Kiểu dữ liệu: boolean.
  + Ràng buộc: bằng true nếu game actor này đang trong lượt và ngược lại.

## Phương thức

* SetIsTurnState
  + Mục đích: Thiết lập giá trị cho thuộc tính isInTurn.
  + Kiểu trả về: Không.
  + Tham số truyền vào:
    - state

Mô tả: Giá trị được gán cho thuộc tính isInTurn.

Kiểu dữ liệu: boolean.

Ràng buộc: Không.

* + Giá trị trả về: Không.
  + Tiền điền kiện: Không.
  + Hậu điều kiện:
  + Các thuộc tính được đọc/ được dùng: isInTurn.
  + Phương thức được gọi: Không.
  + Lưu đồ xữ lý: Gán giá trị state cho thuộc tính isInTurn.

# Class: Bot

* Mô tả: Biểu diển cho thực thể máy trong trò chơi.
* Ràng buộc: Kế thừa từ lớp GameActor.

## Thuộc tính

* level
  + Mô tả: cấp độ của máy.
  + Kiểu dữ liệu: int
  + Ràng buộc: giá trị kéo dài từ 1 đến 3, cấp càng cao thì độ khó càng cao.
* determidedTree
  + Mô tả: cây định trị trò chơi, giúp bot tìm ra đường đi dẫn đến chiến thắng.
  + Kiểu dữ liệu: DeterminedTree
  + Ràng buộc:
* currentNode
  + Mô tả: lưu một node trên determinedTree tương ứng với tình thế của trò chơi hiện tại.
  + Kiểu dữ liệu: Node.
  + Ràng buộc:

## Phương thức

* InitializeDeterminedTree
  + Mục đích: Định trị cho tất cả các node trên cây trò chơi.
  + Kiểu trả về: Không.
  + Tham số truyền vào:
    - totalPebble

Mô tả: Tổng số lượng sỏi của màn chơi.

Kiểu dữ liệu: int

Ràng buộc: Lớn hơn 0.

* + Giá trị trả về: Không.
  + Tiền điền kiện: Màn chơi phải được load hoàn thành.
  + Hậu điều kiện: Cây trò chơi được định trị tất cả các node.
  + Các thuộc tính được đọc/ được dùng: determinedTree (Bot).
  + Phương thức được gọi: DetermieNode (Tree), DetermineChild (Tree).
  + Lưu đồ xữ lý: Tạo một đối tượng DeterminedTree bằng constructor với tham số truyền vào là totalPebble, gọi phương thức CreateTree của DetermiedTree bằng đối tượng vừa mới tạo, sau đó gán đối tượng vừa tạo cho thuộc tính determinedTree của Bot.
* SetBotLevel
  + Mục đích: Thiết lập giá trị cho thuộc tính botLevel của Bot.
  + Kiểu trả về: Không.
  + Tham số truyền vàos:
    - level

Mô tả: cấp độ của bot.

Kiểu dữ liệu: int

Ràng buộc: Có giá trị từ 1 đến 3.

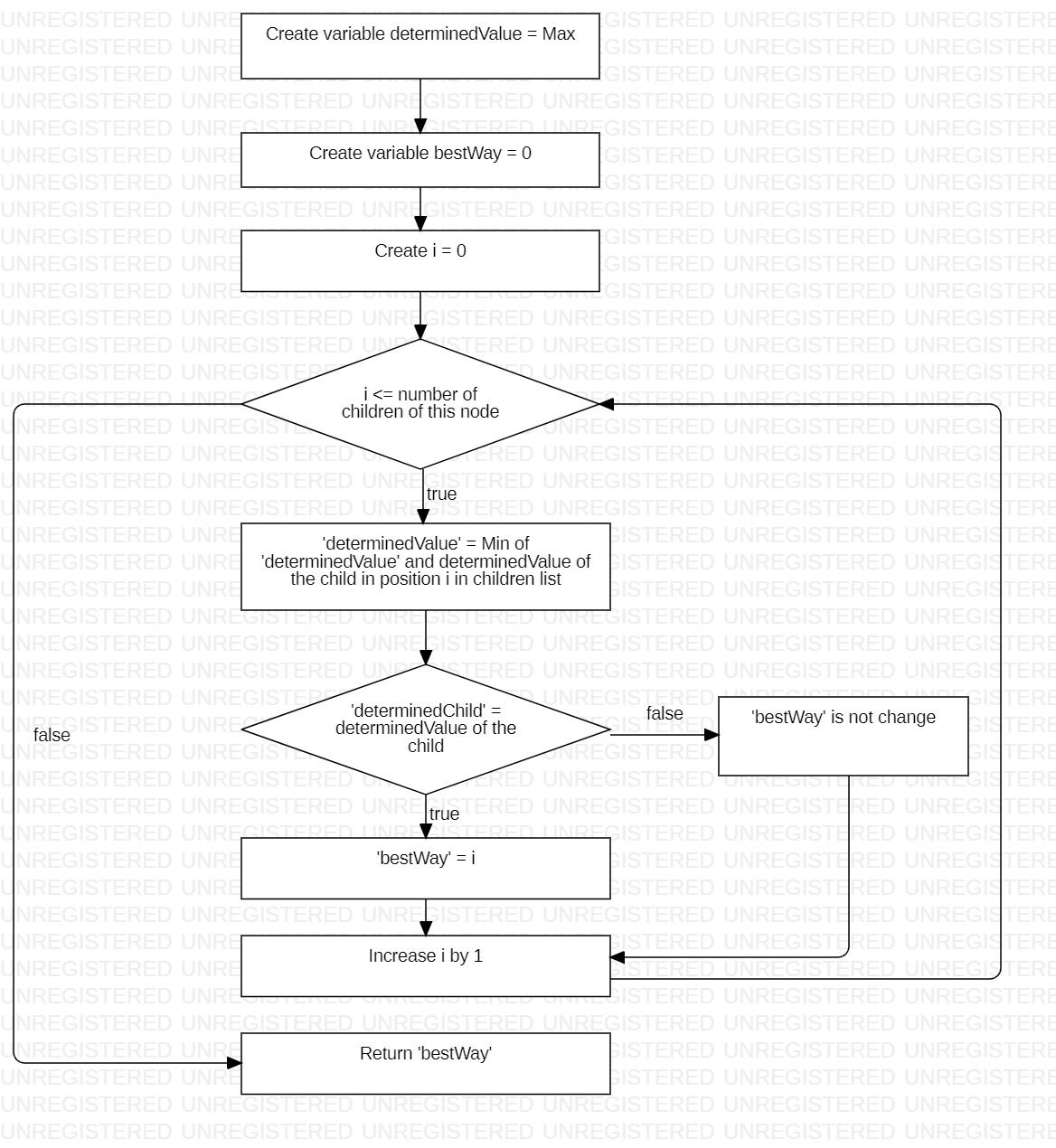
* + Giá trị trả về: Không.
  + Tiền điền kiện:
  + Hậu điều kiện: botLevel được thiết lập.
  + Các thuộc tính được đọc/ được dùng: botLevel (Bot).
  + Phương thức được gọi: Không.
* UpdateCurrentNode
  + Mục đích: Cập nhật lại thuộc tính currentNode khi có sự thay đổi về số sỏi trong màn chơi hiện tại (khi người chơi/ máy bóc sỏi, khi người chơi dùng tùy chọn undo, redo, …)
  + Kiểu trả về: Không.
  + Tham số truyền vàos:
    - currentPebble:

Mô tả: số sỏi còn lại hiện tại của màn chơi.

Kiểu dữ liệu: int

Ràng buộc: lớn hơn 3.

* + Giá trị trả về: Không.
  + Tiền điền kiện: Có sự thay đổi về số lượng sỏi hiện tại trong màn chơi.
  + Hậu điều kiện: Thuộc tính currentNode được cập nhật đúng với số lượng sỏi hiện tại.
  + Các thuộc tính được đọc/ được dùng: currentNode (Bot).
  + Phương thức được gọi: Không.
  + Lưu đồ xữ lý: Kiểm tra thuộc tính value của currentNode và giá trị của tham số currentPebble, nếu thuộc tính value của currentNode có giá trị nhỏ hơn currentPebble thì currentNode sẽ được gán bằng parent của nó trên cây trò chơi. Ngược lại, hiệu số của thuộc tính value của currentNode và currentPebble sẽ xác định giá trị cập nhật của nó (một trong các con của nó).
* FindBestWay
  + Mục đích: Tìm ra đường đi tối ưu nhất để có chiến thắng.
  + Kiểu trả về: int.
  + Tham số truyền vào: Không.
  + Giá trị trả về: Chỉ số node con của currentNode (Bot) mà là node tối ưu nhất.
  + Tiền điều kiện: Người chơi đã xong lượt của mình và GameController đã đổi lượt sang lượt của máy.
  + Hậu điều kiện: Tìm được và trả về node con tối ưu nhất (node con có giá trị định trị là nhỏ nhất).
  + Các thuộc tính được gọi/ được dùng: children.
  + Các phương thức được gọi: Không.
  + Lưu đồ xử lý



* FindWay

# Mẫu:

## Thuộc tính

* Test
  + Mô tả:
  + Kiểu dữ liệu:
  + Ràng buộc:

## Phương thức

* Test
  + Mục đích:
  + Kiểu trả về:
  + Tham số truyền vào:
  + Giá trị trả về:
  + Tiền điều kiện:
  + Hậu điều kiện:
  + Các thuộc tính được gọi/ được dùng:
  + Các phương thức được gọi:
  + Lưu đồ xử lý