**Table of content**

[Mô tả bài toán](#ProblemDescription)

[Các lớp cần thiết](#Classes)

[Flowchart diagram](#FlowchartDiagram)

[Hierarchy](#Herarchy)

[Sequence diagram](#SequenceDiagram)

1. Mô tả bài toán

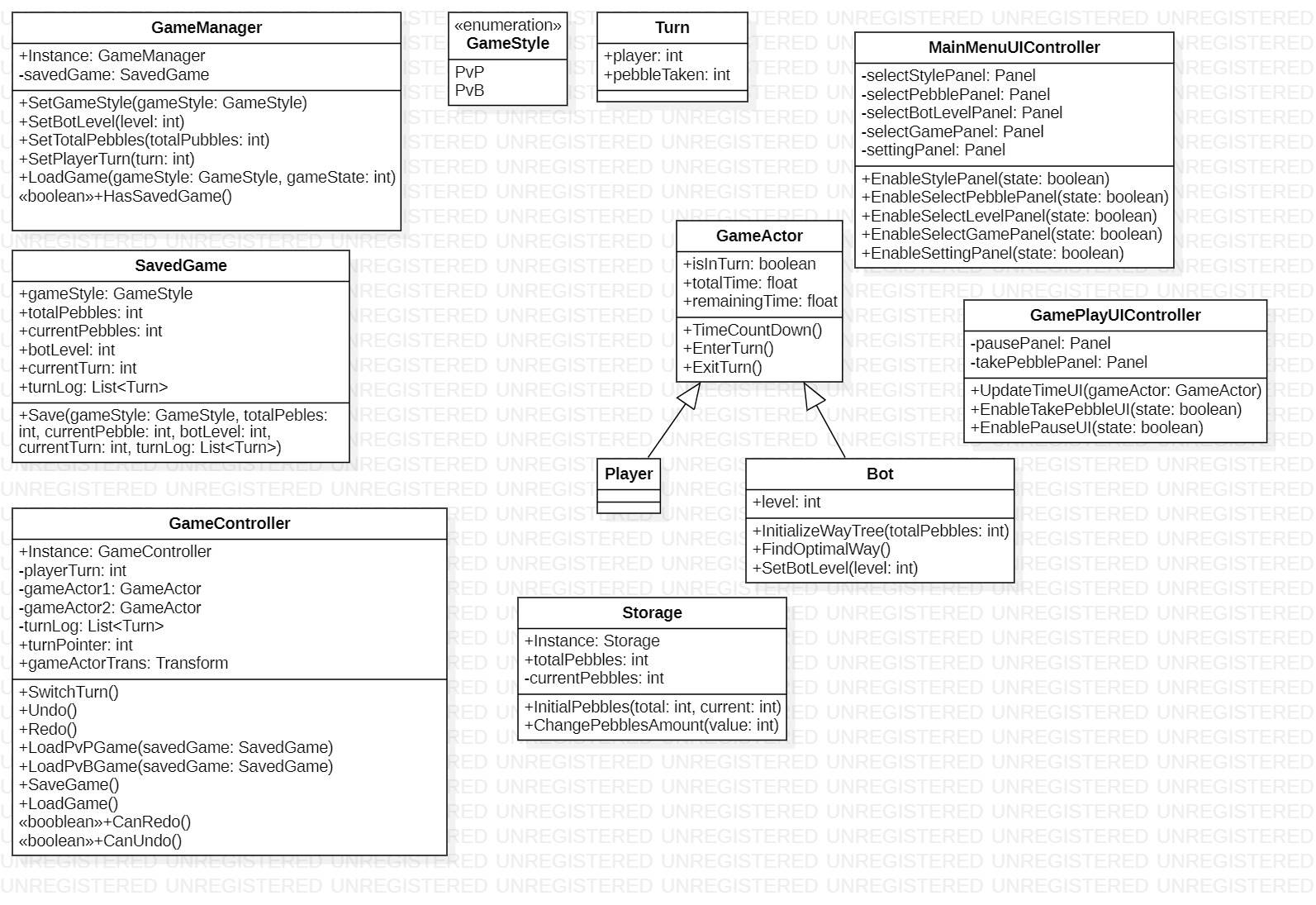
Phát triển một trò chơi cho hai người chơi hoặc một người chơi với bot. Trò chơi có cách chơi như sau, có một đống sỏi với số lượng sỏi nhất định, hai người chơi lần lượt bốc một số lượng sỏi (trong khoảng từ 1 đến một số đã được quy định trước) cho tới khi có một người không còn sỏi để bốc thì người đó sẽ bị xử thua. Bot có thêm chức năng tìm lựa chọn tối ưu. Ngoài thao tác bốc sỏi thì trình điều khiển trò chơi còn hỗ trợ các thao tác khác như undo, redo, dừng trò chơi, tiếp tục trò chơi, lưu trò chơi, …. Hệ thống quản lý trò chơi còn có thêm các chức năng như tạo màn chơi mới, tiếp tục màn chơi cũ, điều chỉnh âm lượng trò chơi, thay đổi ngôn ngữ hiển thị, chọn độ khó cho bot, …

Xác định các actor và use case:

* Player (người chơi)
  + Take pebbles
  + Redo
  + Undo
* Bot
  + Take pabbles
  + Find optimal way
* Storage (đống sỏi)
* Game Manager
* Game Controller
* App user
  + Start game
    - Create new game
    - Continue game
  + Modify volume
  + Change game’s language
  + Select bot’s level
  + Pause game
  + Resume game
  + Save game

1. Các class cần thiết

Sơ đồ lớp



GameManager: class quản lý chung của toàn bộ ứng dụng.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Parameters** | **Purpose** | **Solution** | **Referenced** |
| StartGame |  | Bắt đầu game | Bắt đầu một màn chơi | MainMenuUI  Controller |
| HasSavedGame |  | Kiểm tra xem có màn chơi nào đã được lưu hay không | Trả về thuộc tính savedGame | MainMenuUI  Controller  dùng khi người chơi nhấn Start game |
| SetGameStyle | gameStyle | * Dùng khi người chơi tạo mới màn chơi * Thiết lập giá trị gameStyle cho thuộc tính savedGame | Gán gameStyle cho thuộc tính gameStyle của savedGame | MainMenuUI  Controller dùng khi người chơi chọn game style |
| SetTotalPebbles | totalPebble | * Dùng khi người chơi tạo mới màn chơi * Thiết lập giá trị totalPebbles cho thuộc tính savedGame | Gán totalPebbles cho thuộc tính totalPebbles và currentPebbles | MainMenuUI  Controller dùng khi người chơi chọn số lượng sỏi |
| SetBotLevel | botLevel | * Dùng khi người chơi tạo mới màn chơi * Thiết lập giá trị botLevel cho thuộc tính savedGame | Gán giá trị botLevel cho thuộc tính botLevel của savedGame | MainMenuUI  Controller dùng khi người chơi chọn độ khó |
| SetPlayerTurn | turn | * Dùng khi người chơi tạo mới màn chơi * Thiết lập giá trị currentTurn cho thuộc tính savedGame | Gán giá trị turn cho thuộc tính currentTurn của savedGame | MainMenuUI  Controller dùng khi người chơi chọn lượt đi của người chơi |
| ChangeLanguage | language | Thay đổi ngôn ngữ của ứng dụng |  |  |
| ChangeVolume | deltaVolume | Điều chỉnh âm thanh của ứng dụng |  |  |

GameController: class quản lý những gì liên quan đến game play.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Parameters** | **Purpose** | **Solution** | **Referenced** |
| LoadGame | savedGame | Tải màn chơi (cũ hoặc mới, PvP hoặc PvB) | Dựa vào game style của savedGame để gọi hàm LoadPvPGame hay LoadPvBGame | Phương thức Awake của GameController |
| LoadPvPGame | savedGame | Tải màn chơi người vs người | Lưu đồ bên dưới | Phương thức LoadGame của GameController |
| LoadPvBGame | savedGame | Tải màn chơi người vs máy | Lưu đồ bên dưới | Phương thức LoadGame của GameController |
| SwitchTurn |  | Thay đổi lượt của hai người chơi | Lấy giá trị isInTurn của gameActor1  Gọi hàm SetInTurnState của gameActor1 với tham số truyền vào là !isInTurn của gameActor1  Gọi hàm SetInTurnState của gameActor2 với tham số truyền vào là isInTurn của gameActor1 |  |
| UpdateTurnLog | pebble-  Amount | Cập nhật turn log | * Kiểm tra xem phía sau turn pointer còn có turn nào được lưu không * Nếu có thì xóa bỏ các turn đó khỏi turn log * Thêm turn vừa thực hiện vào turn log | GamePlay  UIController |
| Undo |  | Quay lại bước thực hiện trước đó | Lưu đồ bên dưới |  |
| Redo |  | Thực hiện bước thực hiện sau đó (nếu có) | Lưu đồ bên dưới |  |

GameActor: đại diện cho đối tượng chơi game (player hoặc bot)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Parameters** | **Purpose** | **Solution** | **Referenced** |
| TimeCountDown |  | Đếm ngược thời gian của một lượt của đối tượng chơi | Giảm giá trị của thuộc tính remainingTime nếu isInTurn là true |  |
| EnterTurn |  | Bắt đầu lượt của đối tượng chơi | Đặt giá trị của thuộc tính isInTurn bằng true |  |
| ExitTurn |  | Kết thúc lượt của đối tượng chơi | * Đặt giá trị isInTurn bằng false * Đặt giá trị remainingTime bằng totalTime |  |

Bot

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Parameters** | **Purpose** | **Solution** | **Referenced** |
| InitializeDeterminedTree |  | Khởi tạo cây trò chơi |  |  |
| FindOptimalWay |  | Tìm đường đi nhiều khả năng thắng trên cây trò chơi dựa trên botLevel |  |  |
| SetBotLevel | value | Gán giá trị cho thuộc tính botLevel |  |  |

GameDeterminedTree: biểu diễn cho cây trò chơi, các nút trong cây có chứa 1 giá trị để bot có thể dễ dàng tìm ra chiến thắng.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Parameters** | **Purpose** | **Solution** | **Referenced** |
| CreateTree | root | Tạo cây trò chơi |  | Constructor của GameDeterminedTree |
| DetermineNode | node  playerTurn | Định trị cho một nút | Lưu đồ bên dưới | * Hàm CreateTree * Hàm DetermineChild |
| DetermineChild | playerTurn  value  node | Định trị cho một nút con | Lưu đồ bên dưới | Hàm DetermineNode |

Node: biểu diễn cho một nút trên cây trò chơi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Parameters** | **Purpose** | **Solution** | **Referenced** |
| AddFirstChild | child | Gán nút con thứ nhất cho nút hiện tại | Gán ‘child’ cho thuộc tính firstChild của Node. |  |
| AddSecondChild | child | Gán nút con thứ hai cho nút hiện tại | Gán ‘child’ cho thuộc tính secondChild của Node. |  |
| AddThirdChild | child | Gán nút con thứ ba cho nút hiện tại | Gán ‘child’ cho thuộc tính thirdChild của Node. |  |
| SetPlayerTurn | turn | Gán giá trị cho playerTurn | Gán giá trị ‘turn’ cho thuộc tính playerTurn của Node. |  |
| SetDeterminedValue  ForLeaf |  | Định trị nút lá một nút lá.  Được gọi khi nút này mới vừa khởi tạo, chưa có nút con nào. | Xét giá trị của playerTurn để gán giá trị thích hợp cho thuộc tính determinedValue của Node. | * Hàm AddFirstChild * Hàm AddSecondChild * Hàm AddThirdChild |
| SetDeterminedValue | value | Định trị một nút không phải là nút lá. | Gán giá trị ‘value’ cho thuộc tính determinedValue của Node. | Hàm DetermineNode của GameDeterminedTree |

Storage: class đại diện cho kho chứa sỏi.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Parameters** | **Purpose** | **Solution** | **Referenced** |
| InitialTotalpebbles | totalPebble | Khởi tạo giá trị sỏi ban đầu | * Gán giá trị totalPebble của Storage là totalPebble * Gán giá trị currentPebbles của Storage là currentPebbles | * LoadPvPGame của GameController * LoadPvBGame của GameController |
| ChangePebbleAmount | value | Thay đổi số lượng số sỏi hiện tại dựa vào giá trị value | currentPebbles của Storage sẽ cộng thêm value |  |

GameStyle (enum): chứa các giá trị đại diện cho kiểu chơi (người vs người, người vs máy)

MainMenuUIController: class quản lý UI màn hình chính.

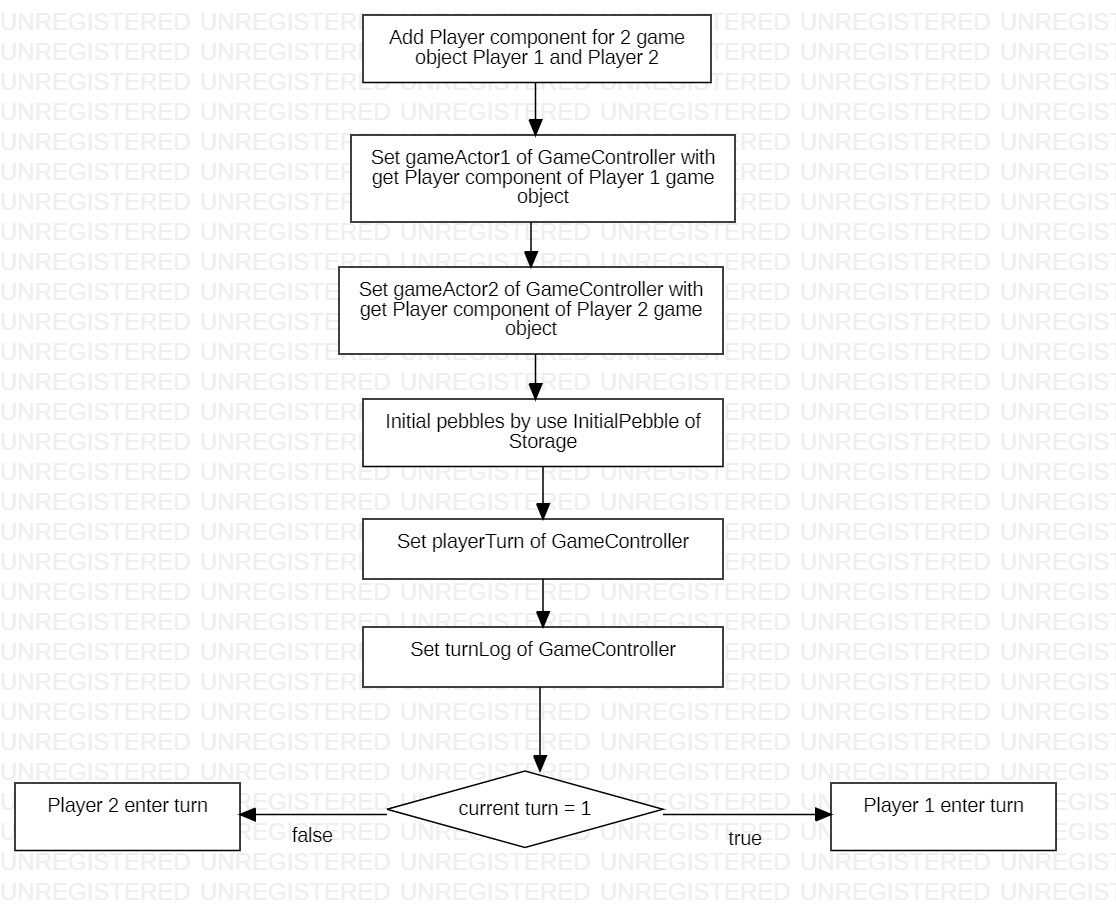
GamePlayUIController: class quản lý UI màn hình game play.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Parameters** | **Purpose** | **Solution** | **Referenced** |
| HightLightPebbleButton | pebbleAmount | Làm nổi bật phím chọn sỏi mà người chơi đã chọn |  |  |
| HightLightAvatar |  | Làm nổi bật avatar của người chơi hiện đang trong lượt |  | Hàm SwitchTurn của GameController |

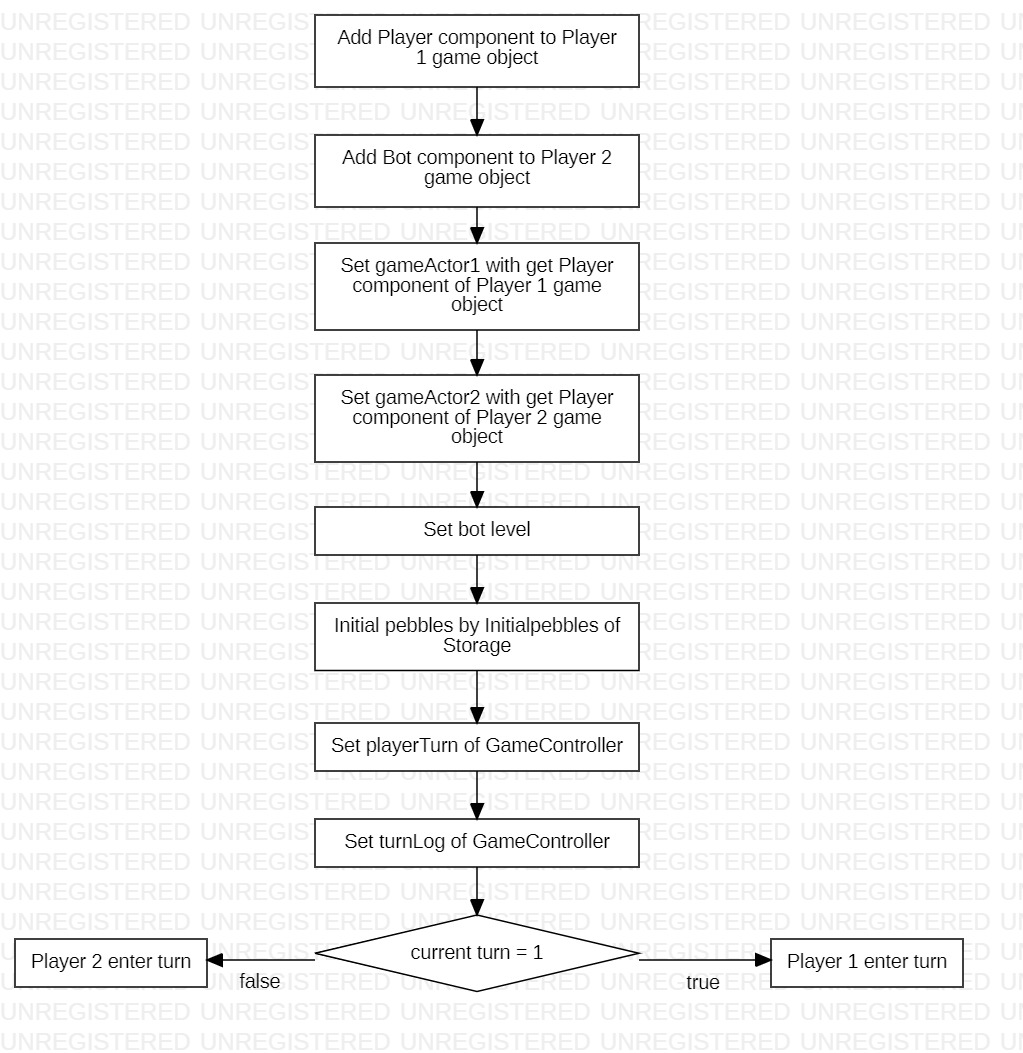
Thay đổi solution của SwitchTurn trong GameController (thêm hight light avatar)

1. FlowChart Diagrams

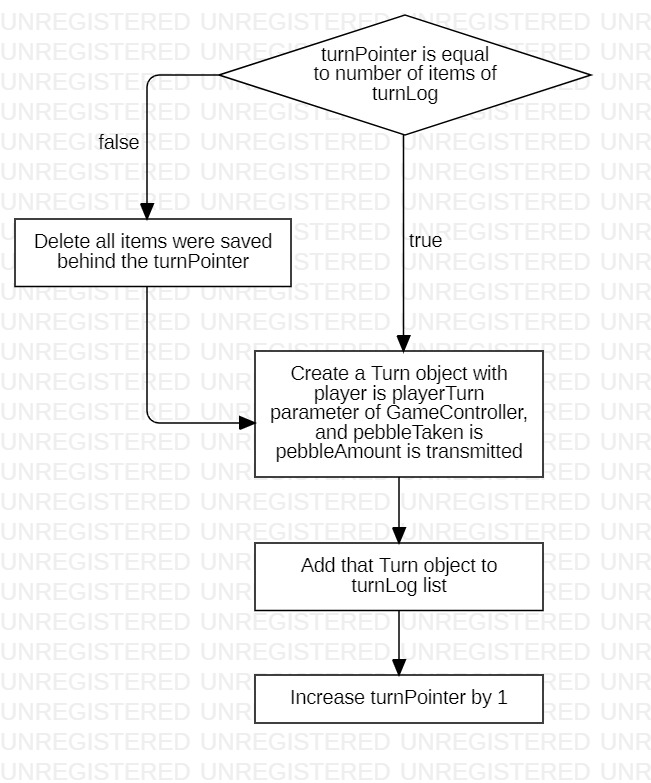
GameController.LoadPvPGame(savedGame)



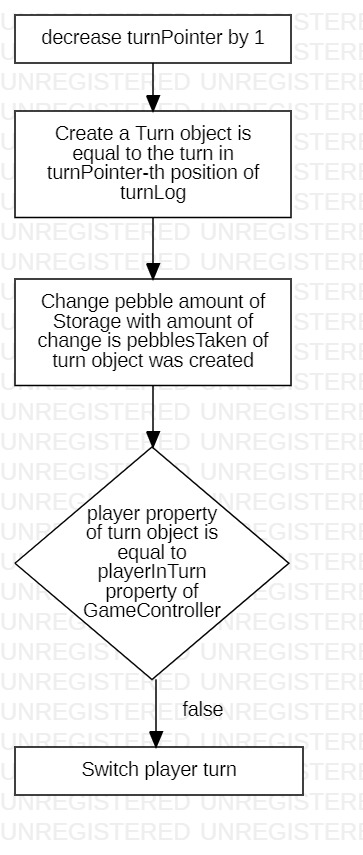
GameController.LoadPvBGame(savedGame)



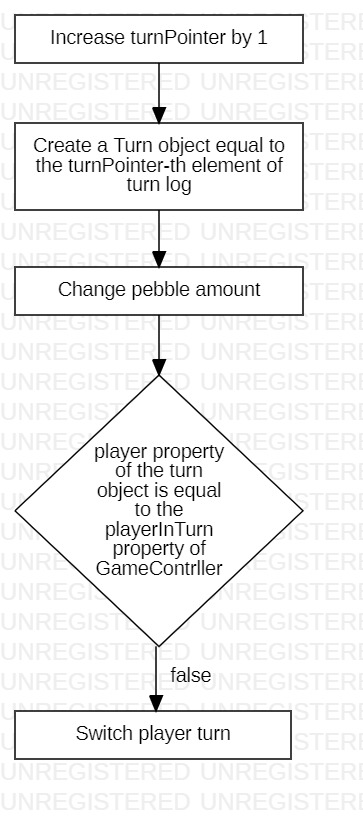
GameController.UpdateTurnLog(playerTurn)



GameController.Undo()

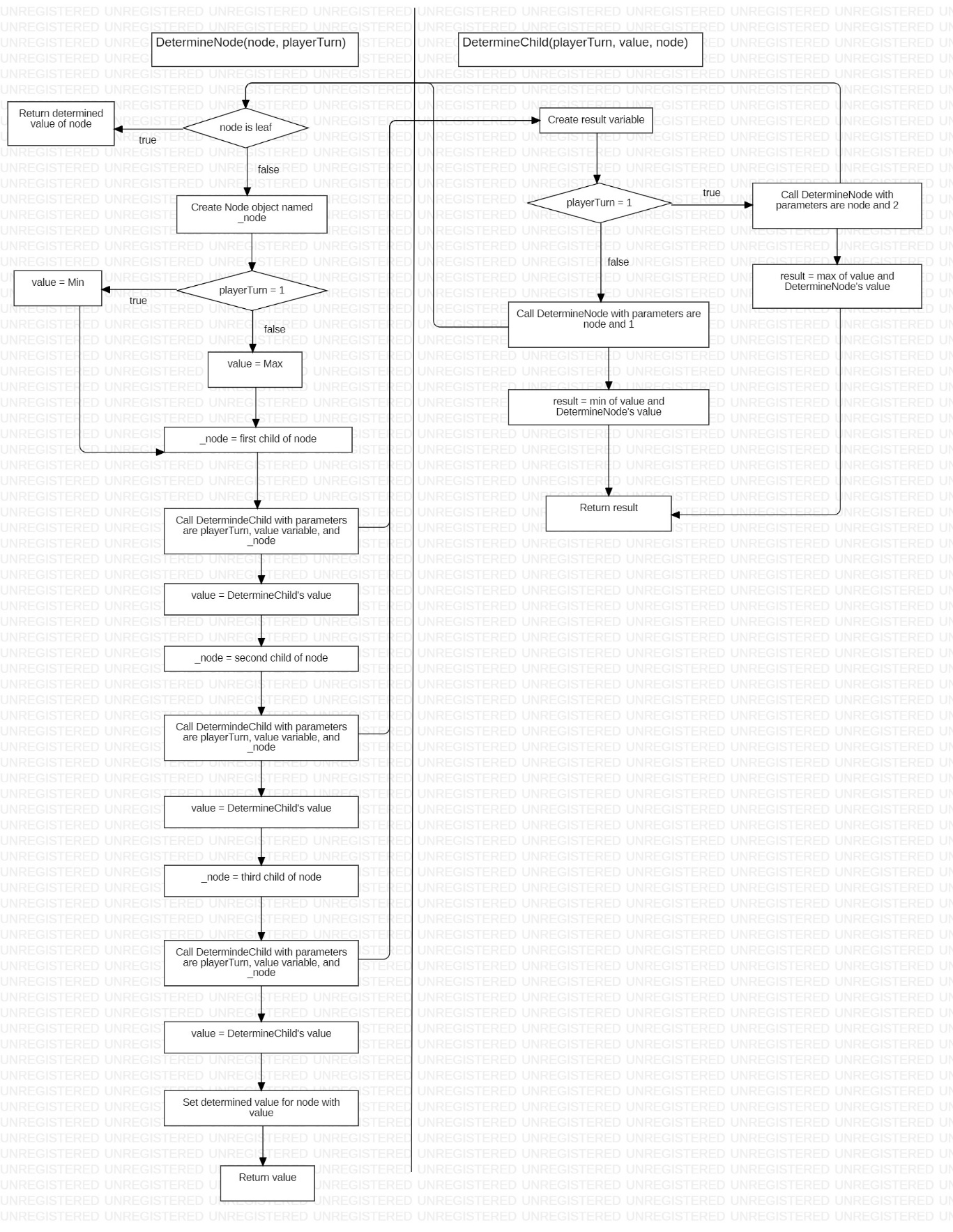
****

GameController.Redo()

****

GameDeterminedTree.DetermineNode(node, playerTurn)

& GameDeterminedTree.DetermineChild(playerTurn, value, node)

****

1. Hierarchy

MainMenu Scene

MainMenuUI

Main Menu Panel

Start Game Button

Setting Button

Exit Button

Setting Panel

Back Button

Sound Text

Sound Editor

Language Text

Language Combobox

Select Game Panel

Back Button

New Game Button

Continue Button

Select Level Panel

Select Pebbel Panel

Don’t Destroy On Load

Game Manager

Gameplay Scene

Game Play UI

Pause Panel

Resume Button

Back To Main Menu Button

Game Play Panel

Player 1 UI

Avatar

Avatar Border

Time Counter

Player 2 UI

Avatar

Avatar Border

Time Counter

Get Pebbles Panel

Get 1 Pebble

Button Background

Button Border

Get 2 Pebbles

Button Background

Button Border

Get 3 Pebbles

Button Background

Button Border

Game Actors

Player 1

Player 2

Storage

Game Controller

Don’t Destroy On Load

Game Manager

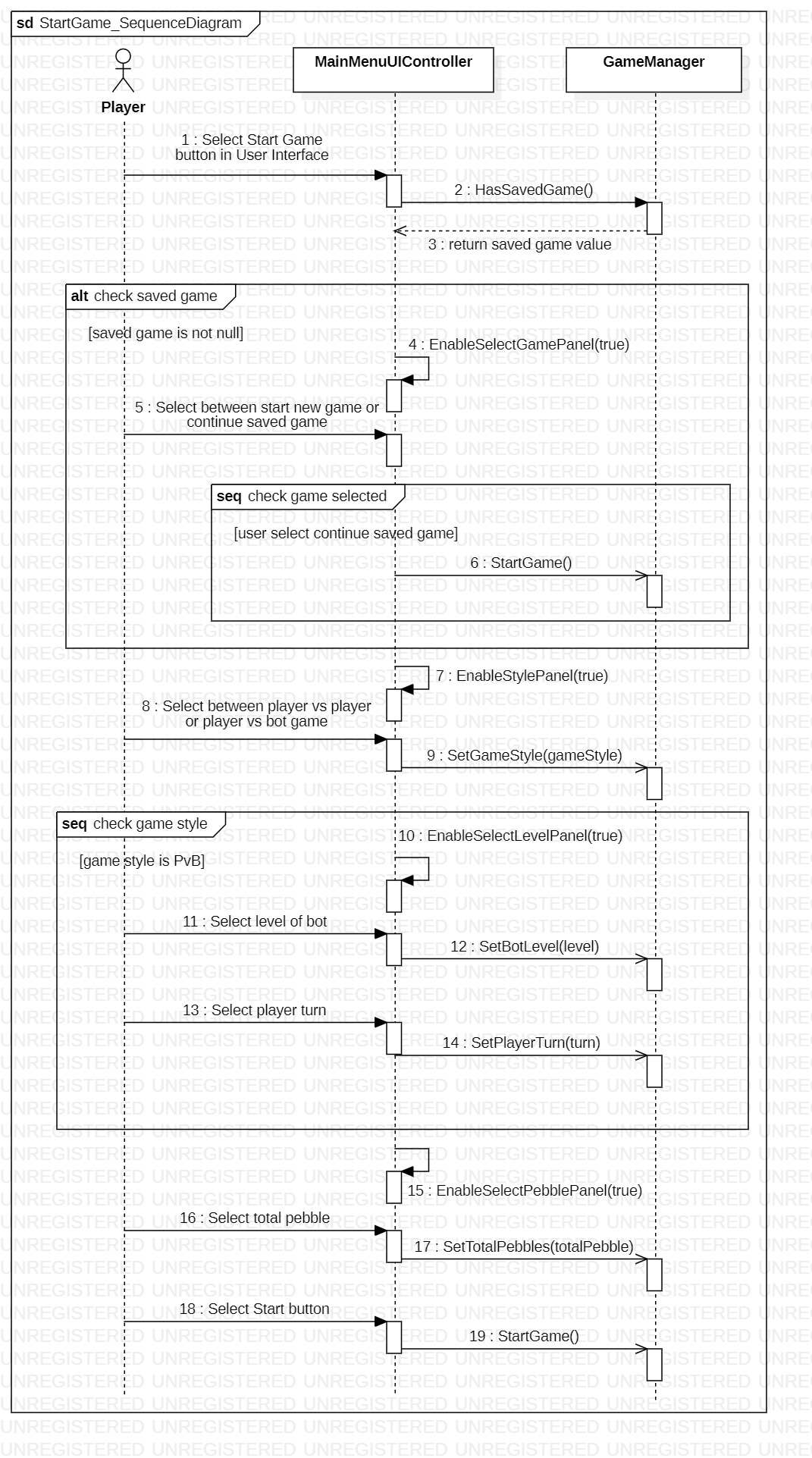
1. Sequence Diagram

Start Game

Kịch bản

1. Người dùng chọn nút Start Game trên màn hình
2. MainMenuUIController gọi hàm HasSavedGame của GameManager
3. GameManager trả về giá trị của savedGame
4. MainMenuUIController kiểm tra giá trị của savedGame, nếu khác null thì sẽ hiển thị giao diện lựa chọn giữa tạo màn chơi mới và tiếp tục màn chơi cũ
5. Người chơi chọn giữa tạo màn chơi mới và tiếp tục màn chơi đã lưu
6. Nếu người chơi chọn chơi tiếp màn chơi cũ thì MainMenuUIContrller gọi hàm StartGame của GameManager
7. Nếu người chơi chọn tạo màn chơi mới, MainMenuUIController sẽ hiển thị giao diện chọn kiểu game (người vs người hay người vs máy)
8. Người chơi chọn kiểu game
9. MainMenuUIController gọi hàm SetGameStyle từ GameManager
10. MainMenuUIController so sánh gameStyle, nếu gameStyle là PvB thì sẽ hiển thị giao diện chọn độ khó của máy
    1. Người chơi chọn độ khó
    2. MainManuUIController gọi hàm SetBotLevel của GameManager
    3. Người chơi chọn lượt đi
    4. MainMenuUIController gọi hàm SetPlayerTurn của GameManager
11. MainMenuUIController hiển thị giao diện chọn số lượng sỏi
12. Người chơi chọn số lượng sỏi
13. MainMenuUIController gọi hàm SetTotalPebbles của GameManager
14. Người chơi chọn nút Start
15. MainMenuUIController gọi hàm StartGame của GameManager

Sơ đồ

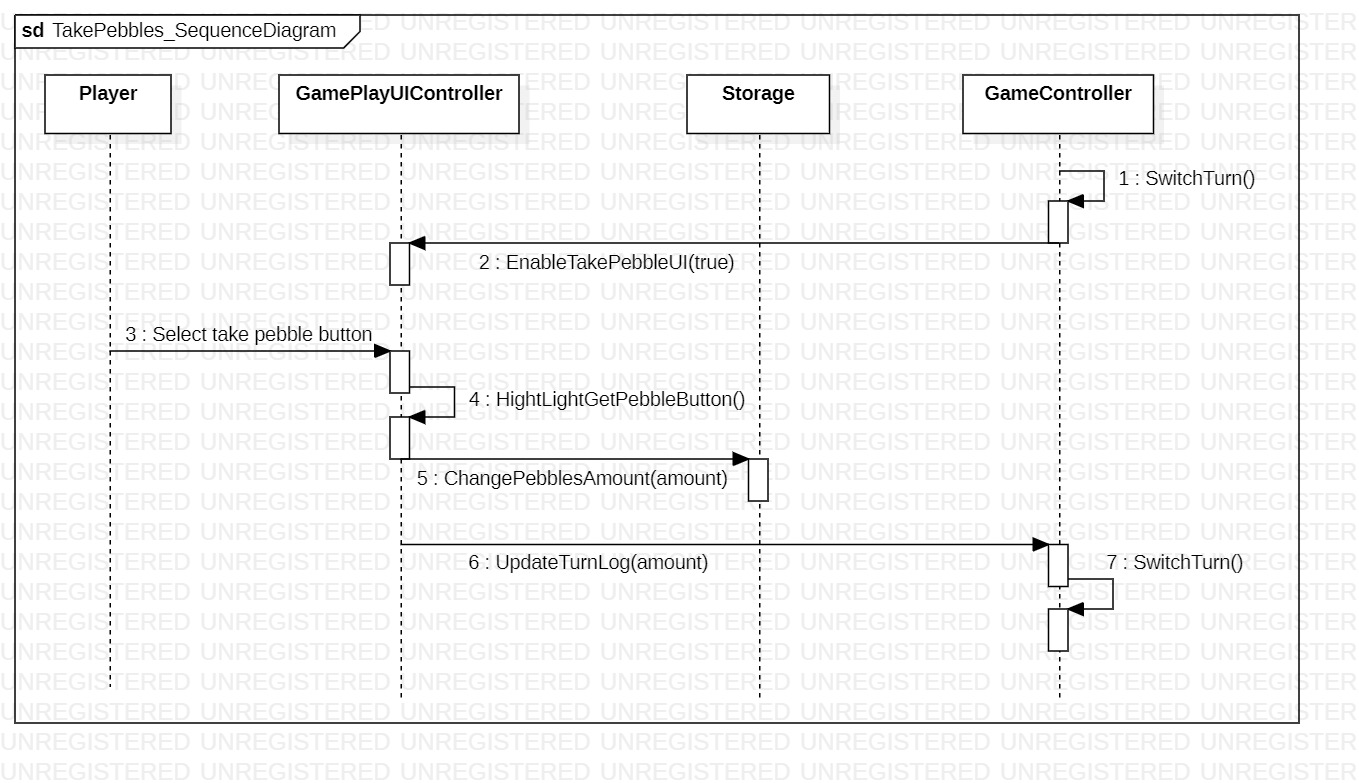


Take Pebbles

Kịch bản

1. Lượt của người chơi bắt đầu
2. GameplayUIController hiển thị giao diện chọn số lượng sỏi
3. Người chơi nhấn vào button có số lượng sỏi muốn bóc
4. GameplayUIController làm nổi bật button vừa chọn
5. Storage thay đổi số lượng sỏi hiện tại
6. GameController cập nhật turn log
7. GameController thay đổi lượt của hai người chơi

Sơ đồ

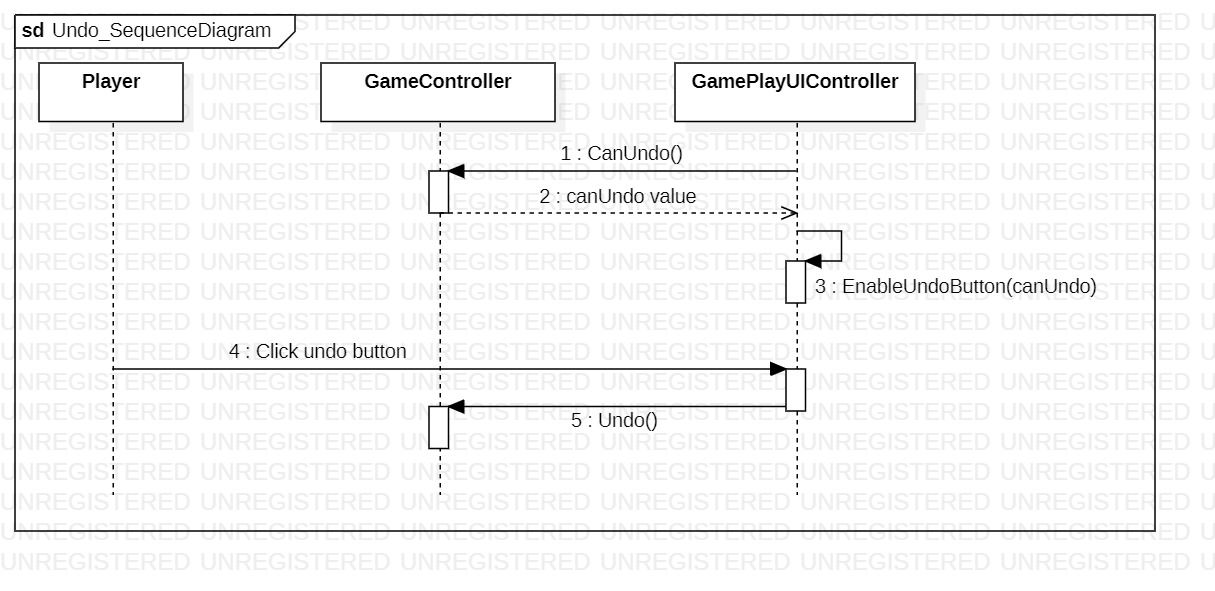


Undo

Kịch bản

1. Kiểm tra xem người chơi có thể undo hay không
2. Nếu không, vô hiệu hóa nút undo
3. Nếu có, hợp lệ hóa nút undo
4. Người chơi nhấn vào nút undo
5. GameController gọi hàm Undo

Sơ đồ



Redo

Kịch bản

1. GamePlayUIController gọi hàm CanRedo của GameController
2. GameController trả về giá trị canRedo
3. GamePlayUIController gọi hàm EnableRedoButton để vô hiệu hóa nút redo hoặc không
4. Người chơi nhấn nút redo
5. GameContrller gọi hàm redo

Sơ đồ

