**Tổng quan về game 1**

**1) Tổng quan về cấu trúc file**

- File chứa game nằm trong mục ./cnpm/phaser3-react-template

- File chứa các java script và image nằm trong mục src

- bên trong src có các folder :

* **asset:** Chứa ảnh và 2 html inputform
* **Config:** Chứa file java script (viết tắt là js) có chức năng cấu hình màn hình chơi game
* **gameObject:** Folder chứa Object của game
* **helper:** folder chứa các chức năng của game
* **Scene:** chứa các scene của game

**1\*) Chi tiết về các folder**

a) **Asset** : Chứa hình ảnh và html inputform

b**) Config**:

* **config.js**: file dùng để cấu hình game

c) **gameObject**:

* **Ball.js**: (**Ball class**) dùng để tạo Ball object
* **Block class**: dùng để tạo Block object

d) **Helper**:

* **Audio** class: dùng để tạo âm thanh
* **BubbleBox** class: dùng để tạo hình chữ nhật đựng chữ
* **MessageBox**: nó cũng giống BubbleBox thôi nhưng có hình tam giác ở trên
* **CheckInputText** class: dùng để Check Input mình nhập vào có đúng với kết quả mà mình muôn ko
* **DisplayBox** class: dùng để khởi tạo và hiển thị BubbleBox và MessageBox (khơi tạo luôn bubble vì mình để class khởi tạo là BubbleBox class và MessageBox class trong class này luôn)
* **DragManage**r: dùng để drag and drop vật thể
* **FdInFdOut**: Tạo hiêu ứng fadeIn và fadeOut

2) **cấu trúc của game**

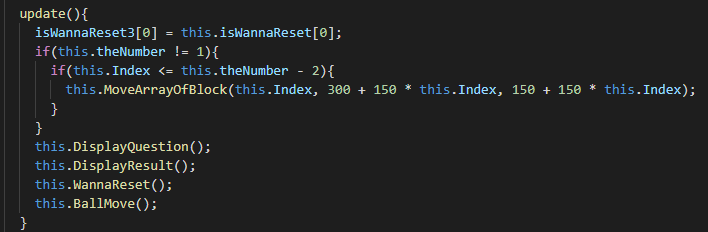
**(Scene1 là GameScene.js,**

**Scene2 là BootScene.js,**

**Scene3 là Game1Scene3.js)**

a) **Scene1**: Scene1 khá dễ nên phần này bỏ qua vì chỉ có 2 giai đoạn trong scene đấy

b) **Scene2**:



Game sẽ chạy theo luồng là:

1) Hàm **MoveArrayBlock**: dùng để check InputText, nếu đúng thì sẽ di chuyển Block

2) Hàm **DisplayQuestion**: Sau khi trả lời hết các câu hỏi nhỏ ở trên, Hàm này sẽ hiển thị câu hỏi lớn ở phía dưới

3) **DisplayResult**: trả lời câu hỏi lớn xong thì sẽ hiện kết quả của nó

4) **WannaReset**: khi trả lời sai thì sẽ có yêu cầu muôn làm lại k

5) **Ball Move**: nếu bước 4 chon không reset hoặc trả lời đúng hết thì bóng sẽ di chuyển và chuyển sang màn mới

Mỗi màn chơi có tích hợp random ngẫu nhiên. Tức là khi reset thì số lượng các object sẽ thay đổi chứ k cố định số lượng

c) **Scene3**: Tương tự như Scene2

3) **Các chức năng Game cần lưu ý**

a) **Về chức năng drag:**

- Nhớ là phải set physic cho các vật thể cần drag, và setInteractive cho chúng

b) **Về chức năng FadeIn FadeOut**

- Nhớ là phải setAlpha cho object muốn tạo hiệu ứng fade. setAlpha(0) là làm vật thể k nhìn thấy, setAlpha(1) là làm hiện vật thể. FadeIn là làm vật thể biến mất còn FadeOut là làm hiện ra.

c) Về hàm this.time.addEvent

- dùng để delay thời gian

*Sẽ cập nhật tiếp…*

**3) Biến**

a) **Scene1:**

- biến **ball** dùng để khởi tạo Ball, biến lastball dùng để trỏ vào ball cuối được khởi tạo

- **holder**: là nơi để chứa block sau khi drag đúng

- **distance[]** : khi drag 1 block thì mảng distance sẽ lưu khoảng cách từ block mình đang drag đến các holder 10 holder ứng với 10 phần tử của mảng)

- **numberOfBox**: sr đáng ra nên đặt là numberOfBlock.

b) **Scene2**:

- Ở Scene 2 thì có các vật thể là:

+ **Block**

+ **BubbleBox**

+ **MessageBox**

+ **Chữ trên Bubble**

+ **InputText**

* Biến **checkInput**: dùng để khởi tạo **CheckInputText** class và để check Input có đúng ko
* Biến **fade**: Dùng để khởi tạo **FdInFdOut** class để tạo hiệu ứng **FadeIn** **FadeOut**
* Biến **box**: Dùng để khởi tạo biến **DisplayBox** để Display **BubbleBox** và **MessageBox**
* Biến **button**:Dùng để tạo Button
* Biến **theNumber**: Dùng để tạo số lượng **Block** ban đầu
* Biến **Index:** Kết hợp với biến isStayCheck[], tí nói luôn 1 thể
* Biến **inputform**: tạo inputText từ html
* Biến **inputText**: lấy giá trị từ inputText
* Biến **txtArr**: biến này mình k biết nên đặt thế nào cho phù hợp, cơ mà nó dùng để thay thế inputform khi trả lời đúng. Ví dụ khi nhập đúng số 10, thì inputform sẽ biến mất và hiện lên text ghi ‘10’
* Biến **bubble**: đay là object có 2 thành phần là **contentArr** với **graphicArr**. **contentArr** để chứa chữ trong 1 Bubble và **graphicArr** là hình chữ nhật để chứa chữ. Gọi là graphic bới vì cái bubble đó được vẽ bằng graphic của Phaser
* Biến **Input**: **TextArr**: Chứa giá trị của inputext, **formArr**: là ô để nhập input
* Biến **msg**(viết tắt của Message): tương tự như bubble
* Các biến bool:
* Giả sử Scene mình có 3 Blocks cùng với 3 InputText cần phải trả lời.
* Nhưng phải trả lời đúng 1 cái thì mới hiện sang cái thứ 2.
* Biến **isStayCheck[]** dùng để check xem mình đang ở câu hỏi nào. **isStayCheck[Index = 0]** thì đang ở cau hỏi 1, 1 thì ở câu hỏi 2… nếu là true thì tức là mình vẫn chưa trả lời xong câu hỏi đấy, còn false tức là đã trả lời xong và cho cái **isStayCheck[Index+1]** là true.
* Biến **isMoving[]** dùng để check xem block đã được di chuyển chưa
* (\* cái biến **isMoving** với cả **isStayCheck** có vẻ same same nhau. Lúc đấy do chauw làm rõ tổng thể đã code nên h hơi rối, sau này sẽ sửa lại)
* Biến **isDisplayResult:** check xem đã được gọi hàm **DisplayResult** chưa
* Biến **intinalizedCompleted:** dùng để check xem là phải thay thế hết các **inputform** thành **text** rồi mới được chạy hàm **DisplayResult**
* Biến **isWannaReset**: dùng để check xem đã được gọi hàm **WannaReset** chưa
* Biến **isPlaytilEnd:** check xem là đã chạy hết game của scene này hay chưa (vì phải chạy hết game xong mới được reset xem có muốn chơi lại không)