ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

BÁO CÁO DỰ ÁN

NGÀNH: CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI: XÂY DỰNG CÁC TRÒ CHƠI NIM GAMES

Giảng viên đánh giá: **Th.S Hồ Đắc Phương**

Giảng viên đánh giá: **TS. Nguyễn Hà Nam**

Sinh viên: **Nguyễn Đức Hoàng**

Mã sinh viên: **13020713**

Lớp: **K58CB**

Hà Nội, tháng 4 năm 2017

Mục lục

I. LỜI MỞ

Trước hết, em xin chân thành cảm ơn tới thầy Nguyễn Hà Nam đã hết sức tạo điều kiện cho em hoàn thành dự án này.

Em cũng gửi lời cảm ơn sâu sắc tới thầy Hồ Đắc Phương đã tận tình chỉ bảo và hướng dẫn em trong suốt quá trình làm dự án.

Bước đầu làm một dự án thực sự, lần đầu trải nghiệm môi trường thực tế, kiến thức còn nhiều hạn chế nên không tránh khỏi những sai sót trong bản báo cáo. Em rất mong nhậnương được những ý kiến đóng góp quý giá của quý thầy cô để em có thể hoàn thiện báo cáo dự án này.

Em xin chân thành cảm ơn!

II. GIỚI THIỆU CHUNG

1. Giới thiệu công việc

1.1. Thời gian làm dự án

Từ 01/02/2017 – 01/05/2017

1.2. Yêu cầu công việc

Mỗi feature tuân thủ làm trong một tuần, sáng thứ 6 mỗi tuần báo cáo tiến độ với thầy Hồ Đắc Phương.

1.3. Công việc chính

Tìm hiểu các trò chơi nim.

Tìm hiểu cách làm game bằng PhaserJs.

Phát triển các trò chơi nim.

2. Giới thiệu bài toán

2.1. Tên sản phẩm

Nim games

Mã nguồn có thể tải về từ:

Github: https://github.com/hoangdevelopers/nim_game

2.2. Công nghệ chính sử dụng

NodeJs.

PhaserJs.

Webpack.

ES6.

2.3. Ý nghĩa sản phẩm

Từ bao lâu nay việc chơi cùng và giáo dục trẻ nhỏ vẫn luôn là một đề tài được các bậc phụ huynh

hết sức quan tâm. Mỗi lứa tuổi sẽ có một mức độ nhận thức khác nhau, căn cứ vào đó người làm cha mẹ phải tìm kiếm và lựa chọn những trò chơi phù hợp.

Tuy vậy trong thực tế hiện nay các trò chơi cho trẻ em còn chưa được những nhà làm game quan tâm. Những trò chơi chỉ mới hoàn thành nhiệm vụ giải trí chứ chưa có nhiều trò chơi giáo dục nhất là đối tượng người dùng nhỏ tuổi. Không những vậy những nhiều trò chơi còn gây tác hại không nhỏ cho người chơi nhỏ tuổi chưa có nhiều kinh nghiệm cũng như hiểu biết. Nhìn nhận được thách thức đó, tôi quyết định tìm hiểu về các trò chơi có tính xây dựng, định hướng và kích thức phát triển nơi trẻ. Một trong các trò đó là Nim.

Với mục tiêu nhằm tạo ra một công cụ mở và miễn phí dễ dàng thay đổi luật chơi, chế độ chơi giúp cho việc tạo ra các trò chơi các bậc phụ huynh giáo dục con em mình.

Nim là một game chiến thuật toán học có xuất xứ từ châu Âu. Nguyên lý của trò chơi dựa vào phép toán trừ. Mỗi lượt đi của mình ảnh hưởng tới lượt tiếp theo của đối phương và ảnh hưởng đến toàn cuộc. Trò chơi giúp các bé rèn luyện khả năng tính toán, tư duy logic và phán đoán.

2.4. Cách chơi

Có 3 khu vực chứa quân với số lượng lần lượt 1-3-5.

Khi tới lượt mình mỗi người sẽ lấy ít nhất một quân, nhiều nhất là tất cả quân của cùng một khu. Cứ thế lần lượt cho đến khi có người thắng.

Có 2 cách để phân biệt thắng - thua:

Người lấy thanh cuối cùng sẽ thắng

Người lấy thanh cuối cùng sẽ thua

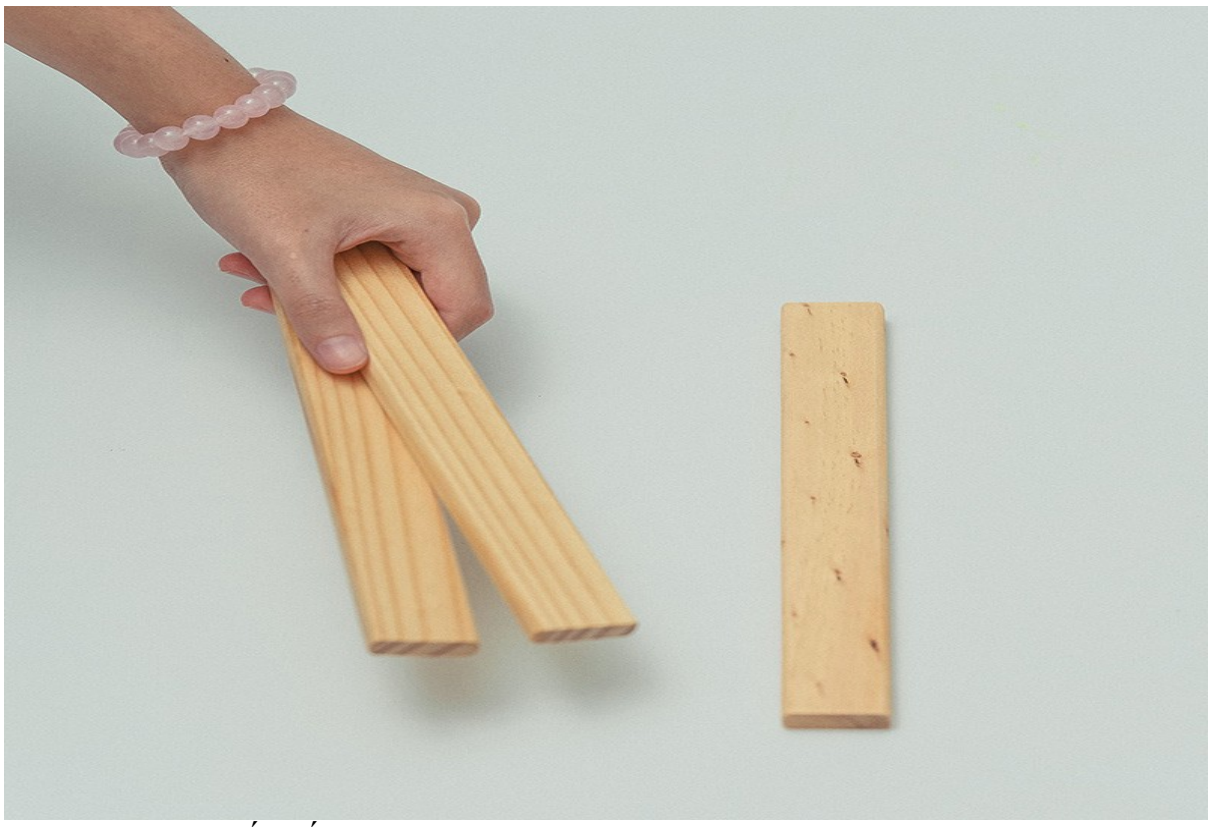
Vật liệu cần thiết: 9 hòn sỏi / thanh gỗ / cây bút chì… để làm quân cờ. Có thể linh hoạt các vật liệu. Tôi dùng gỗ để làm ví dụ cho luật chơi Nim.

Ví dụ: *Đây là một ván Nim cơ bản.*

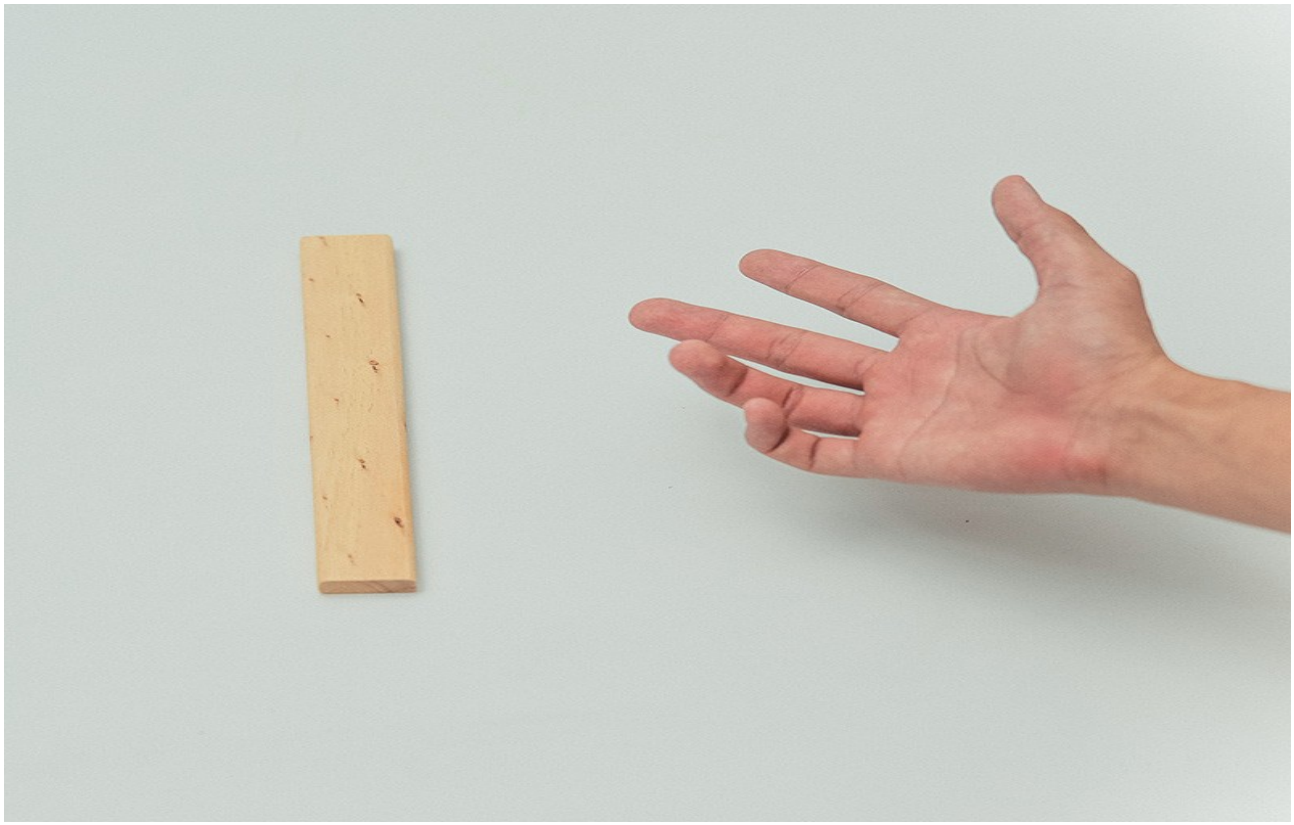
Hình

1: Bắt đầu ván Nim. Có 3 khu vực chứa quân từ trái qua phải: I,II,III.

Hình 2: Bạn nữ (tay đeo vòng) bắt đầu trước với một thanh ở khu I. Lượt bạn nam (tay không đeo vòng) lấy hết quân ở khu III.



Hình 3: Bạn nữ lấy tiếp 2 quân ở khu II.



Hình 4: Bạn nam thua theo luật thông thường vì chỉ còn một quân ở khu II.

Chỉ với một vài thanh gỗ, hòn sỏi...là chúng ta đã có thể vừa chơi vừa giáo dục định hướng cho con. Thật đơn giản phải không các bố, các mẹ? Trò chơi không chỉ dừng lại với trẻ con, đối với người

lớn các ván đấu Nim còn căng thẳng và hấp dẫn hơn. Một trò chơi nhẹ để luyện tập bộ não vào cuối tuần. Thật dễ dàng phải không nào?

III. CƠ SỞ LÝ THUYẾT

1. NodeJs

1.1. Giới thiệu chung

NodeJS là một nền tảng Server side được xây dựng dựa trên Javascript Engine (V8 Engine). Node.js được phát triển bởi Ryan Dahl năm 2009 và phiên bản cuối cùng là v0.10.36.

Định nghĩa NodeJs bởi tài liệu chính thức như sau:

Node.js là một nền tảng dựa vào Chrome Javascript runtime để xây dựng các ứng dụng nhanh, có độ lớn. Node.js sử dụng các phần phát sinh các sự kiện (event-driven), mô hình non-blocking I/O để tạo ra các ứng dụng nhẹ và hiệu quả cho các ứng dụng về dữ liệu thời gian thực chạy trên các thiết bị phân tán.

NodeJs là một mã nguồn mở, đa nền tảng cho phát triển các ứng dụng phía Server và các ứng dụng liên quan đến mạng.

Ứng dụng Node.js được viết bằng Javascript và có thể chạy trong môi trường Node.js trên hệ điều hành Window, Linux...Node.js cũng cung cấp cho chúng ta các module Javascript đa dạng, có thể đơn giản hóa sự phát triển của các ứng dụng web sử dụng Node.js với các phần mở rộng.

1.2. Ưu điểm

Dưới đây là vài đặc điểm quan trọng biến Node.js trở thành sự lựa chọn hàng đầu trong phát triển phần mềm:

Không đồng bộ và Phát sinh sự kiện (Event Driven):

Tất các các APIs của thư viện Node.js đều không đồng bộ, nghĩa là không blocking (khóa). Nó rất cần thiết vì Node.js không bao giờ đợi một API trả về dự liệu. Server chuyển sang một API sau khi

gọi nó và có cơ chế thông báo về Sự kiện của Node.js giúp Server nhận đựa phản hồi từ các API gọi trước đó.

Chạy rất nhanh: Dựa trên V8 Javascript Engine của Google Chrome, thư viện Node.js rất nhanh trong các quá trình thực hiện code.Các tiến trình đơn giản nhưng hiệu năng cao: Node.js sử dụng một mô hình luồng đơn (single thread) với các sự kiện lặp. Các cơ chế sự kiện giúp Server trả lại các phản hồi với một cách không khóa và tạo cho Server hiệu quả cao ngược lại với các cách truyền thống tạo ra một số lượng luồng hữu hạn để quản lý request. Nodejs sử dụng các chương trình đơn luồng và các chương trình này cung cấp các dịch vụ cho số lượng request nhiều hơn so với các Server truyền thống như Apache HTTP Server.Không đệm: Ứng dụng Node.js không lưu trữ các dữ liệu buffer.Có giấy phép: Node.js được phát hành dựa vào MIT License.

Với Node.js, bạn phải làm mọi thứ.

Node.js chỉ là một môi trường – điều này có nghĩa bạn tự phải làm mọi thứ. Sẽ chẳng có bất kỳ máy chủ mặc định nào cả !!! Một đoạn script xử lý tất cả các kết nối với Client. Điều này làm giảm đáng kể số lượng tài nguyên được sử dụng trong ứng dụng.

1.3. Nhược điểm

Ứng dụng nặng tốn tài nguyên. Nếu bạn cần xử lý các ứng dụng tốn tài nguyên CPU như encoding video, convert file, decoding encryption... hoặc các ứng dụng tương tự như vậy thì không nên dùng NodeJS (Lý do: NodeJS được viết bằng C++ & Javascript, nên phải thông qua thêm 1 trình biên dịch của NodeJS sẽ lâu hơn 1 chút). Trường hợp này bạn hãy viết 1 Addon C++ để tích hợp với NodeJS để tăng hiệu suất tối đa !(Việc tích hợp rất thân thiện và nhanh chóng)!

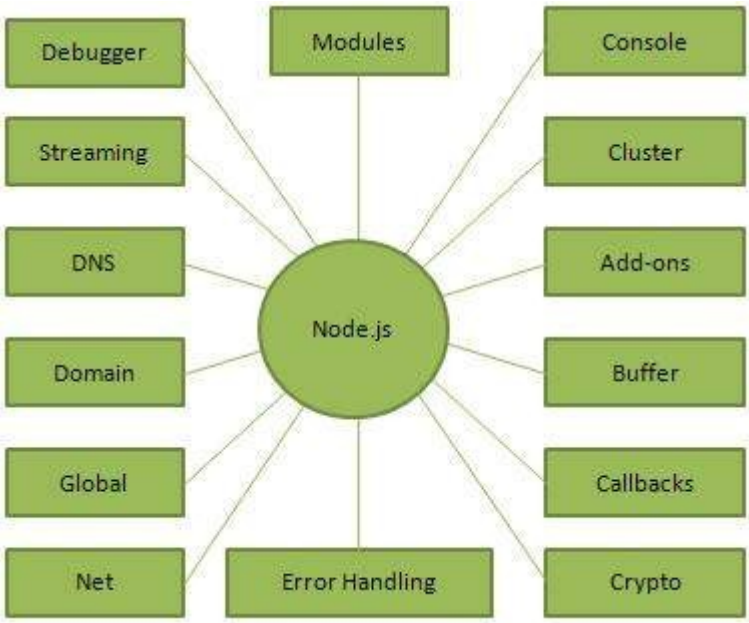
Với NodeJS, NoSQL thì là sự kết hợp hoàn hảo nhưng :
Bạn là người có kinh nghiệm với các ngôn ngữ lập trình để phát triển các dự án. Bạn biết được NodeJS qua tin tức, báo chí, bạn bè... Bạn quyết định xây dựng dự án bằng NodeJS. Nhưng khi gặp sự cố rủi ro xây dựng dự án với NodeJS đồng thời quay lưng luôn.

1.4. Ai sử dụng Node.js?

Dưới đây là link trên github wiki chứa danh sách các dự án, ứng dụng và các công ty sử dụng Node.js. Trong danh sách này bao gồm eBay, GE, GoDaddy, Microsoft, Paypal, Uber....

1.5. Các thành phần quan trọng trong Node.js

Lược đồ dưới đây mô tả các thành phần quan trọng của Node.js.



Hình 5: Các thành phần quan trọng trong Node.js

1.6. Cài đặt

Trên Ubuntu, chúng ta sẽ dùng apt để cài đặt NodeJS.

Đầu tiên, bạn nên update tất cả package của hệ điều hành để đảm bảo việc cài đặt NodeJS không gặp vấn đề.

sudo apt-get update
Cài NodeJS:

sudo apt-get install nodejs
Cài đặt NPM:

sudo apt-get install npm
Để kiểm tra NPM và NodeJS đã cài đặt được chưa:

nodejs -v
npm -v

2. PhaserJs

2.1. Giới thiệu chung

Phaser là một game framework cho desktop và mobile được xây dựng dựa trên Pixi.js. Nó thì nhanh, miễn phí và là mã nguồn mở. Phiên bản Phaser hiện tại là 2.1.3. Nó hỗ trợ cho cả WebGL và Canvas. Nó có một loạt các tính năng giúp bạn trong việc phát triển game. Nó cũng giống như game framework Flixer cho ActionScript 3. Trong bài viết này, tôi sẽ xây dựng một bộ khung trò chơi với Phaser.

2.2. Ưu điểm

Phaser là một engine mã nguồn mở JavaScript dùng để viết các trò chơi 2D đồng thời sử dụng Canvas hoặc WebGL nếu có.

Phaser hỗ trợ các hiệu ứng, particles, scale và rotare sprites. Phaser có sẵn bản đồ lưới và sử dụng 3 engines vật lý khác nhau, phụ thuộc vào nhu cầu của bạn. Nó cũng có built-in camera dùng để hỗ trợ điều hướng trong thế giới game.PhaserCó một vài plug-in cho Phaser, bao gồm particle system designer và Box2D plugin.

Nếu bạn ưa thích ngôn ngữ JavaScript, Phaser kết hợp với PhoneGap là một sự lựa chọn tốt cho bạn.

2.3. Nhược điểm

Phaser hỗ trợ các hiệu ứng vật lý nhưng nó làm không thực sư tốt như các framework khác như Physics2D

2.4. Ai sử dụng PhaserJs

Hàng ngàn sản phẩm được phát triển từ PhaserJs như Cartoon Candies của Numidia Studio hay burbles.io một tựa game được piohát triển dựa theo xu hướng game

2.5. Cài đặt

Tải thư viện PhaserJs về từ:
Github: https://github.com/photonstorm/phaser-ce/releases/download/v2.7.6/phaser.min.js

3. Webpack

3.1. Giới thiệu chung

Webpack là một công cụ hỗ trợ xây dựng JavaScript module trong các ứng dụng của bạn. Webpack đơn giản hóa các workflow bằng việc xây dựng một cách nhanh chóng một đồ thị tham chiếu (dependency graph) trong ứng dụng của bạn và sắp xếp nó một cách chính xác.

Webpack có thể được cấu hình theo tùy chọn code của bạn và tách biệt thành cách vendor/js/css trên production. Chạy với server với mode development có thể xem có thể những thay đổi code của bạn của bạn một cách nhanh nhất và có rất nhiều đặc điểm thú vị.

3.2. Ưu điểm

Ngày nay các website đang có xu hướng trở thành những web app với các đặc tính như:

Càng ngày càng sử dụng JS nhiều hơn

Những browser ngày càng hỗ trợ những công nghệ mới

Những trang full-page-reload ít đi, single page app lên ngôi

Dẫn đến phần code client-side ngày càng nhiều.

Điều đó có nghĩa chúng ta cần phải có một công cụ để quản lí chúng một cách hiệu quả. Và webpack là một công cụ rất mạnh để làm điều đó. Nó là một module bundler rất mới nhưng đã gây sốt gần đây. Nó nhận vào các module cùng với các dependencies và generate ra các static assets tương ứng.

Webpack hỗ trợ bạn làm những việc sau:

Chia các cây dependency thành các chunk được load khi cần thiết

Thời gian init ngắn hơn

Mỗi static asset đều có thể trở thành một module

Khả năng tích hợp 3rd-party library như module

Khả năng custom gần như mọi thành phần của module bundler

Phù hợp với các dự án lớn

3.3. Nhược điểm

Do Webpack có khá nhiều chức năng nên đối với người mới học nodejs sẽ có nhiều khó khăn, đối với những dự án nhỏ không cần thiết phải sử dụng thì cũng không cần dùng.

Có thể đọc thêm về so sánh các trình bundlers.

https://webpack.js.org/guides/comparison/

3.4. Cài đặt

Để bắt đầu, chúng ta sẽ cài webpack bằng lệnh sau:

npm install webpack -g
Sau đó, tạo ra 2 file: index.html và app.js

app.js:

document.write('welcome to my app');console.log('app loaded');
index.html

```
<html>
	<body>
		<script src="bundle.js"></script>
	</body>
</html>
```

Trong terminal, chúng ta chạy lệnh sau:

webpack ./app.js bundle.js
Lệnh trên webpack sẽ đọc vào file app.js và xuất ra file bundle.js cho chúng ta.

4. ECMAScript 6

4.1. Giới thiệu chung

Javascript, - ngôn ngữ lập trình web mà chúng ta vẫn sử dụng còn có một tên gọi khác là ECMAScript ?

ECMAScript hiện nay không phải là phiên bản đầu tiên.

Dưới đây là các mốc thời gian của ECMAScriptECMAScript

1:(ES1) ra đời năm*1997*ECMAScript

2:(ES2) được giới thiệu năm 1998ECMAScript

3:(ES3) được giới thiệu năm 1999ECMAScript

4:(ES4) được giới thiệu năm 2000

jQuery ra đời năm 2006

NodeJS ra đời năm 2009

ECMAScript 5:(ES5) ra đời năm 2011

ECMAScript 6:(ES6) tháng 6 năm 2015

Nodejs là một cách để chạy javascript trên phía server và rất thích hợp với trình duyệt Chrome. Hiện đang là cái tên rất hot hiện nay trong lĩnh vực phát triển web.

Node js hiện đã hỗ trợ ECMAScript6 (ES6) với cú pháp gọn gàng và mạnh mẽ hơn. Sau đây là một số những tiện ích thú vị mà ES6 đã thay đổi so với "người đàn anh" ES5 của mình.

4.2. Ưu điểm

ES6 có nhiều thay đổi so với các phiên bản trước như ví dụ như:

Function, variables hostting nghĩa là các hàm, biến được dùng trước khi định nghĩa.

Rest Parameters : các đối số được đối xử như một mảng

...vv

Hỗ trợ Class, extend để lập trình hướng đối tượng

Map, Export, import ...vv

Đọc thêm các tính năng của ES6

Gits: https://gist.github.com/vasco3/22b09ef0ca5e0f8c5996

4.3. Nhược điểm

Vì ES6 còn khá mới nên một số trình duyệt cũ không hỗ trợ nên muốn chạy trên các trình duyệt này ta cần sử dụng babel để convert sang ES5

IV. PHÂN TÍCH THIẾT KẾ

1. Giới thiệu sản phẩm

1.1. Sản phẩm: Nim games

Nim games là một trò chơi toán học nổi tiếng cho 2 người chơi. Đối tượng của phần mềm là người chơi hoặc lập trình viên. Đối với người chơi bình thường có thể chơi với máy hoặc chơi với người, có luật chơi có tùy chọn độ khó...vv Đối với lập trình viên có thể cài đặt luật chơi, cài đặt các giao diện người chơi, cài đặt chiến lược cho máy đánh..vv

Phần mềm cần phải đáp ứng được các yêu cầu về coding style, bảo mật cũng như các yêu cầu về chức năng, ngoài ra các module này còn phải hoạt động tốt với các theme sẵn có cũng như các plugin có của hệ thống. Về giao diện module phải có khả năng tùy biến cũng như thêm sửa xóa về sau.

1.2. Module

Phần mềm sẽ có ba module chính:

states: chứa các state trạng thái các trạng thái của game như trạng thái load font, load assets, và hiển thị tương ứng với mỗi trạng thái ra màn hình vv...

Mỗi State được thừa kế từ lớp State của phaser. Trong phaserJs lớp State cung cấp các thuộc tính có thể truy cập nhanh đến các file assets, camera, input, t vv... Ngoài ra các nó cung cấp các phương thức quản lý một trạng thái của game từ lúc khởi tạo state, tải các tệp resource, trình bày giao diện vv...

Dưới đây là một state cơ bản

```
import Phaser from 'phaser'
export default class extends Phaser.State {
     init () {
     }

     preload () {
          this.load.spritesheet( 'mummy' , './assets/mummy.png')
     }

     render () {
     }
     update () {
     }
}
```

player: chứa các lớp người chơi ví dụ như người chơi là Computer, người chơi là người, người chơi là server ...vv các lớp người chơi đều được thừa kế lớp player

Lớp player giữ thông tin người chơi cũng như trạng thái của người chơi ...vv

gameplay: chứa các phương thức luật chơi, kịch bản của máy tính cho người chơi là computer.

2. Phân tích thiết kế

2.1. Yêu cầu chức năng

Người chơi có thể chơi nim với máy tính

Lập trình viên có thể tạo một game nim khác với luật chơi do họ quy định

Lập trình viên có thể thay đổi các chi tiết game như số đống item số item trong 1 đống dễ dàng

Lập trình viên có thể thay phương thức kiểm tra thắng cuộc

2.2. Yêu cầu phi chức năng

Khả năng sử dụng:

Giao diện đẹp và dễ hiểu. Phải dễ sử dụng sao cho một người dùng mới có thể hiểu và sử dụng trong vòng 2 phút.

Độ tin cậy: Có khả năng hoạt động 24/7.

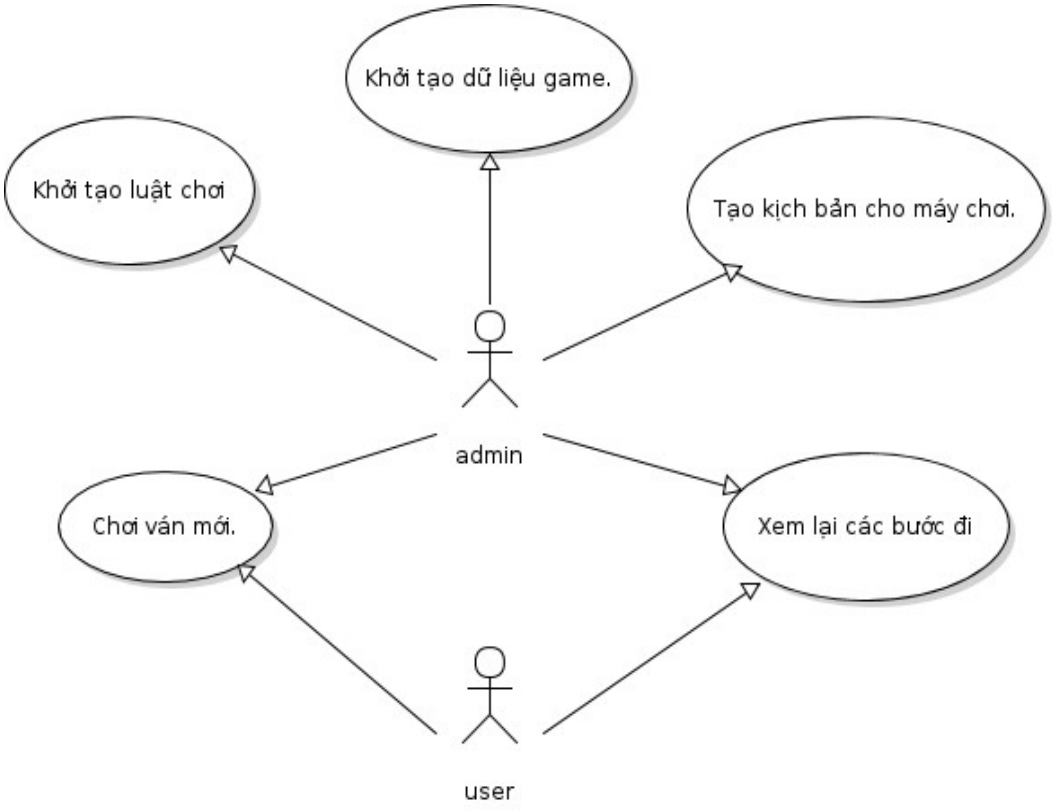
Hiệu năng: tốc độ thực thi không quá 2s.

Bảo mật: Không rò rỉ thông tin khác hàng.

Ràng buộc thiết kế: Cung cấp giao diện thích hợp cho cả máy tính và di động.

3. Phân tích yêu cầu

3.1. Biểu đồ UseCase



Hình 6: Biểu đồ UseCae

3.2. Phân tích use-case

3.2.1. Tạo luật chơi

Tên ca sử dụng: UC1_CreateRule.

Tác nhân: admin.

Tiền điều kiện: Người dùng đã truy cập hệ thống.

Hậu điều kiện: Người dùng xem được nội dung trang web.

Luồng điều khiển chính: Người dùng nhập luật chơi vào editor của page và gửi lên server.

Database lưu nội dung của luật chơi, bao gồm mô tả luật chơi (cho người chơi đọc) và file js (để máy đọc)

Luồng thay thế: Mất kết nối hệ thống hoặc dataserver. thông báo cho người dùng.

3.2.2. Tạo dữ liệu game

Tên ca sử dụng: UC2_InitGameData

Tác nhân: admin.

Tiền điều kiện: Người dùng đăng nhập hệ thống.

Hậu điều kiện: Người dùng xem được nội dung trang web.

Luồng điều khiển chính: Người dùng nhập dữ liệu game như ai chơi trước, map ..vv

Luồng thay thế: Mất kết nối hệ thống hoặc dataserver. thông báo cho người dùng.

3.2.3. Tạo kịch bản cho máy chơi

Tên ca sử dụng: UC3_InitScript.

Tác nhân: admin.

Tiền điều kiện: Người dùng đăng nhập hệ thống.

Hậu điều kiện: Người dùng xem được nội dung trang web.

Luồng điều khiển chính: Người dùng nhập kịch bản cho máy tính chơi ..vv

Người dùng submit kịch bản lên.

Server nhận dữ liệu và lưu trữ.

Luồng thay thế: Mất kết nối hệ thống hoặc dataserver. thông báo cho người dùng.

3.2.4. Chơi ván mới

Tên ca sử dụng: UC3_InitScript.

Tác nhân: admin.

Tiền điều kiện: Không có

Hậu điều kiện: Người dùng xem được nội dung trang web.

Luồng điều khiển chính: Người dùng chọn Chơi ván mới.

Luồng thay thế: Mất kết nối hệ thống hoặc dataserver. thông báo cho người dùng.

3.2.5. Xem lại các bước đi

Tên ca sử dụng: UC3_InitScript.

Tác nhân: admin.

Tiền điều kiện: Không có.

Hậu điều kiện: Người dùng xem được nội dung trang web.

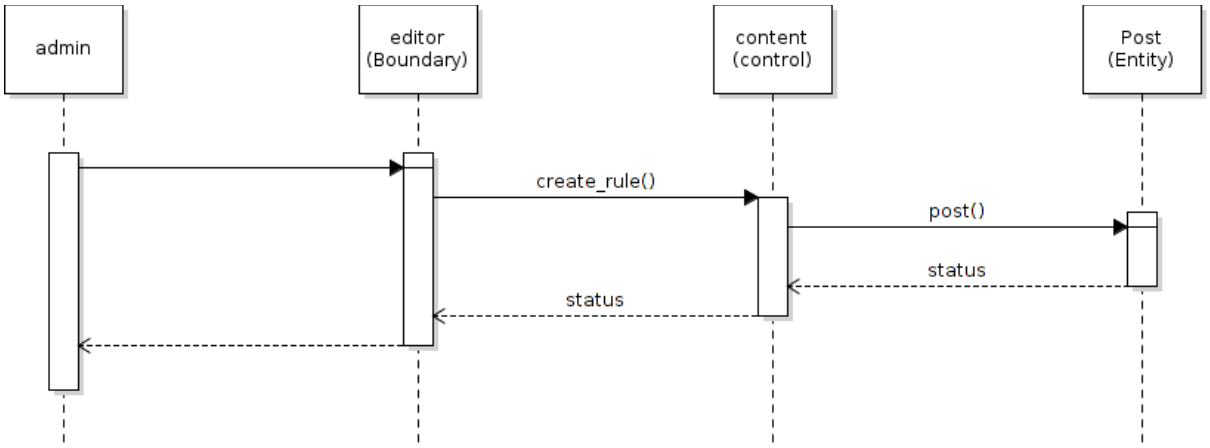
Luồng điều khiển chính: Người dùng chọn Chơi ván mới.

Hệ thống khởi tạo dữ liệu & trả lại giao diện cho người dùng.

Luồng thay thế: Không có.

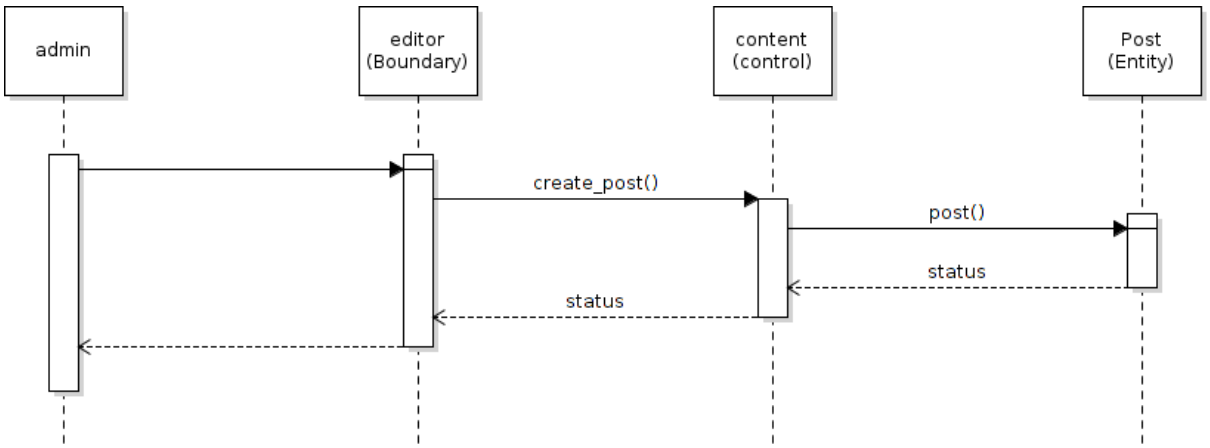
3.3. Sơ đồ UML cho từng Use-case

3.3.1. Tạo luật chơi



Hình 7: Sơ đồ UML khởi tạo dữ liệu game

3.3.2. Khởi tạo dữ liệu game



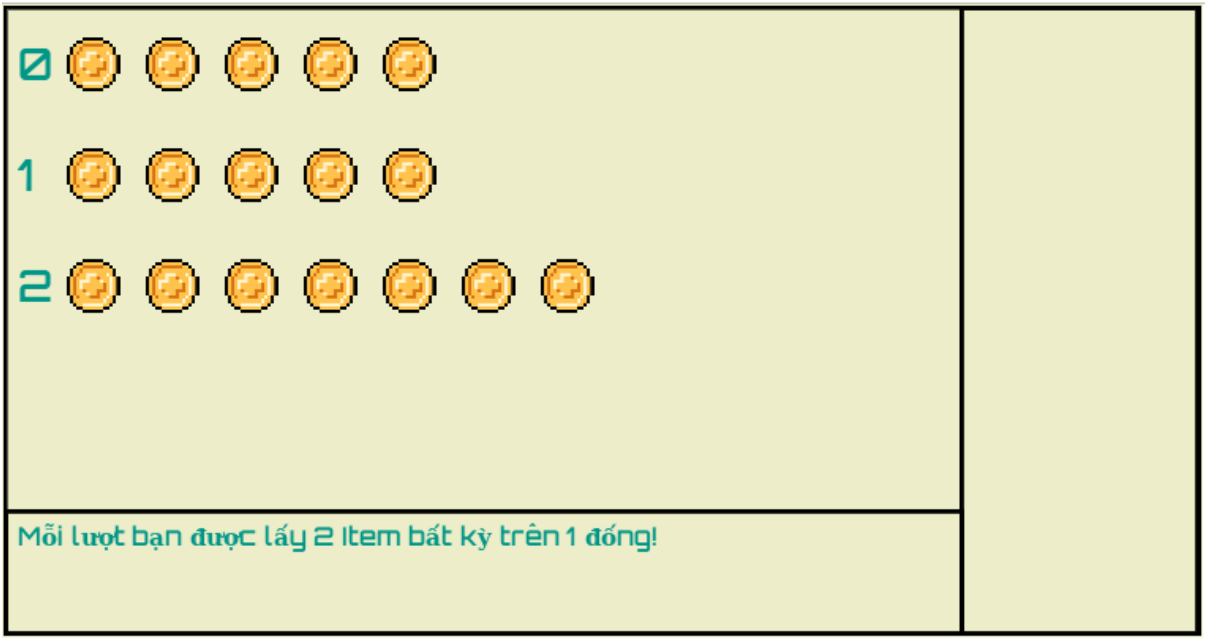
Hình 8: Sơ đồ UML khởi tạo dữ liệu game

V. HÌNH ẢNH DEMO SẢN PHẨM

1.1. Màn hình bắt đầu.

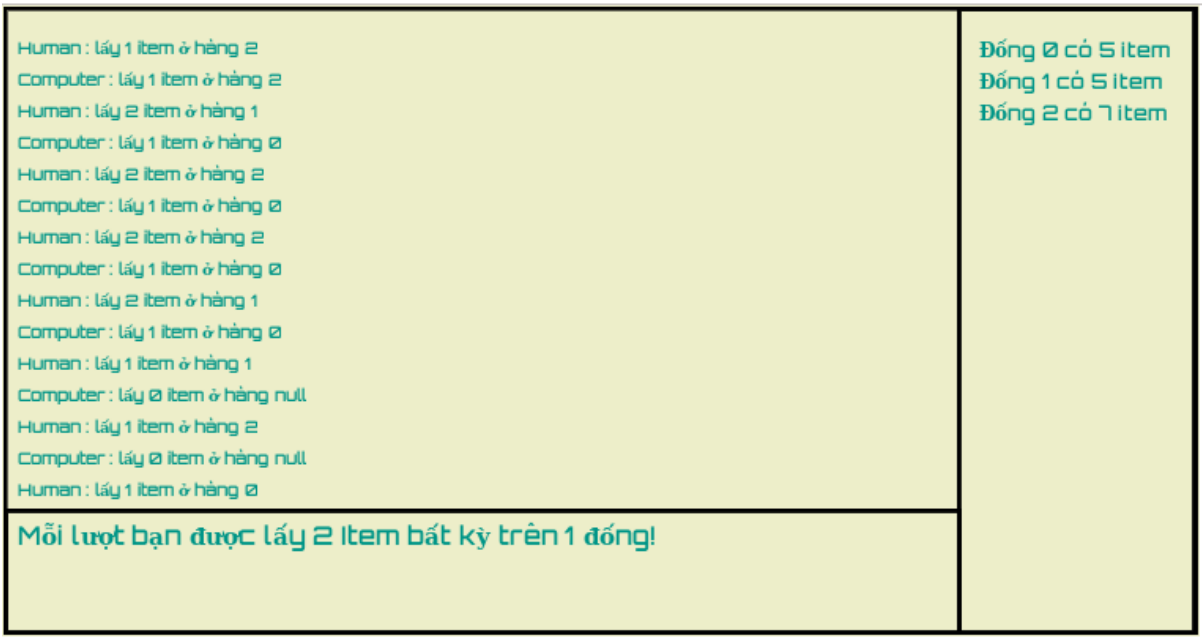
Hình 9: màn hình bắt đầu game

1.2. Màn hình chơi game.



Hình 10: Màn hình chơi game

1.3. Màn hình xem lại các bước chơi.



Hình 11: màn hình kết qua và các bước chơi

VI. KẾT QUẢ ĐẠT ĐƯỢC

- Hiểu được cách thức hoạt động của mô hình thiết kế phần mềm đơn giản.
- Biết được cách làm việc nhóm, thu thập và giải quyết các yêu cầu của cấp trên đưa ra.
- Sử dụng được các mã nguồn mở, các framework có sẵn để giải quyết yêu cầu đề bài.
- Được tìm hiểu framework nodejs...
- Được trang bị thêm kiến thức về PhaserJs, nodejs qua các buổi thực hành.
- Được tiếp xúc với môi trường làm việc chuyên nghiệp, có trách nhiệm.
- Hiểu được tâm lí khách và mong muốn của khách hàng với một sản phẩm công nghệ.
- Có được nhiều trải nghiệm mới về ngành nghề đang theo học.
- Sử dụng thành thục github để quản lí mã nguồn của đội.
- tìm hiểu các yêu câu của một trang giáo dục trực tuyến.

1. Hướng phát triển trong tương lai

- Tăng khả năng dễ sử dụng cho người dùng.

- Cải thiện UX/UI.

- Cải thiện tốc độ tải của trang.

V. Tài liệu tham khảo

https://phaser.io/learn

https://nodejs.org/en/docs/

https://gist.github.com/hoangdevelopers/60998bf4f4d3cc4d6325fdef2e850881

https://webpack.github.io/docs/

VII. Đánh giá

Ý kiến đánh giá:

……

……

……

……

……

……

……

……

Hà Nội, ngày tháng năm 20 .

Người hướng dẫn

(Ký, ghi rõ họ tên & dấu công ty)

Ý kiến đánh giá:

……

……

……

……

……

……

……

……

Điểm số: ……. Điểm chữ: ……

Hà Nội, ngày tháng năm 20 .

Giảng viên đánh giá

(Ký, ghi rõ họ tên)