

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO TIỂU LUẬN
LẬP TRÌNH WEB VÀ ỨNG DỤNG
WEB GAME

Người hướng dẫn: GV Nguyễn Thanh Quân

Người thực hiện: Lê Minh Thành – 51900691

Trần Mỹ Hạnh - 51900794

Trần Tuấn An - 52100951

Khóa: 23,25

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO TIỂU LUẬN
LẬP TRÌNH WEB VÀ ỨNG DỤNG
WEB GAME

Người hướng dẫn: **GV Nguyễn Thanh Quân**

Người thực hiện: **Lê Minh Thành – 51900691**

Trần Mỹ Hạnh - 51900794

Trần Tuấn An - 52100951

Khóa: **23,25**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022

LỜI CẢM ƠN

Lời đầu tiên, chúng em xin gửi lời cảm ơn chân thành đến giảng viên TS. Nguyễn Thanh Quân đã hướng dẫn chúng em hoàn thành bài tiểu luận. Trong quá trình học tập và tìm hiểu môn Lập trình web và ứng dụng, chúng em không chỉ nhận được từ thầy sự hướng dẫn tận tình, tâm huyết mà còn nhận được sự quan tâm, giúp đỡ. Từ những kiến thức được thầy truyền đạt chúng em có thể áp dụng thực tế vào việc xây dựng các trang web. Và quan trọng hơn thế nữa, đây chính là nền tảng kiến thức giúp chúng em phát triển sự nghiệp sau này.

Bên cạnh đó, chúng em xin gửi lời cảm ơn đến khoa Công nghệ thông tin và trường đại học Tôn Đức Thắng đã tạo điều kiện cho chúng em học tập thật tốt.

Thông qua bài tiểu luận này, chúng em xin trình bày lại những kiến thức mà chúng em đã tiếp thu và tìm hiểu được trong quá trình học. Do kiến thức là vô hạn mà sự tiếp thu của mỗi người luôn tồn tại những hạn chế nên trong quá trình làm bài, chúng em khó tránh khỏi những sai sót. Chúng em mong nhận được những góp ý đến từ thầy để bài tiểu luận của chúng em được hoàn thiện hơn.

Chúng em xin kính chúc thầy nhiều sức khỏe, hạnh phúc và thành công trên con đường sự nghiệp giảng dạy.

BÀI TIỂU LUẬN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Chúng em xin cam đoan đây là bài tiểu luận của riêng chúng em và được sự hướng dẫn của TS. Nguyễn Thanh Quân. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong bài tiểu luận còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào chúng em xin hoàn toàn chịu trách nhiệm về nội dung tiểu luận của mình. Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do chúng em gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 8 tháng 11 năm
2022*

*Tác giả
(ký tên và ghi rõ họ tên)*

Lê Minh Thành

Trần Mỹ Hạnh

Trần Tuấn An

Lê Minh Thành

Trần Mỹ Hạnh

Trần Tuấn An

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

Phần xác nhận của GV hướng dẫn

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

Phần đánh giá của GV chấm bài

Tp. Hồ Chí Minh, ngày tháng năm
(ký và ghi họ tên)

MỤC LỤC

CHƯƠNG 1: GIỚI THIỆU GAME	<u>42</u>
1. Thành phần game	<u>42</u>
2. Quy tắc chơi	<u>42</u>
3. Cách chơi	<u>43</u>
4. Tính năng	<u>43</u>
5. Sơ đồ tuần tự	<u>76</u>
CHƯƠNG 2: ỨNG DỤNG VÀ NGÔN NGỮ LẬP TRÌNH	<u>76</u>
1. PHP	<u>76</u>
2. HTML	<u>87</u>
3. Javascript	<u>87</u>
4. Bootstrap 4	<u>87</u>
CHƯƠNG 3: CƠ SỞ DỮ LIỆU	<u>87</u>
1. Bảng cơ sở dữ liệu	<u>87</u>
2. Quản lý dữ liệu	<u>98</u>
2.1. Quản lý dữ liệu trò chơi và người dùng với phpMyAdmin	<u>98</u>
2.2. Quản lý Ball	<u>109</u>
2.3. Quản lý Brick	<u>119</u>
2.4. Quản lý Paddle	<u>1140</u>
2.5. Quản lý Score	<u>1140</u>
2.6. Quản lý tài khoản	<u>1240</u>
CHƯƠNG 4: PHÁT TRIỂN GAME	<u>1240</u>
1. Xây dựng giao diện Game	<u>1340</u>
1.1. Trang chủ	<u>1340</u>
1.2. Trang đăng nhập	<u>1542</u>
1.3. Trang Đăng ký	<u>1543</u>
1.4. Trang Getdata	<u>1543</u>
1.5. Trang Uploaddata	<u>1644</u>
1.6. Trang Submit	<u>1644</u>
1.7. Trang utils	<u>1846</u>
2. Xử lý sự kiện với JavaScript	<u>2149</u>
2.1. Hàm SetupBricks()	<u>2120</u>
2.2. Hàm ResetPaddle	<u>2221</u>

2.3.Hàm resetBall	<u>23</u>
2.4.Hàm xử lý sự kiện trạng thái của game	<u>23</u>
2.5.Hàm bắt Sự kiện di chuyển Paddle	<u>25</u>
2.6.Hàm vẽ Paddle trong game	<u>25</u>
2.7.Hàm vẽ ball trong game	<u>26</u>
2.8.Hàm vẽ bảng tính điểm	<u>26</u>
2.9.Hàm vẽ gạch	<u>27</u>
2.10.Hàm xử lý va chạm ball và wall	<u>27</u>
2.11.Hàm xử lý va chạm ball và paddle	<u>28</u>
2.12.Hàm xử lý va chạm với gạch	<u>29</u>
2.13.Hàm tạo mới thêm gạch	<u>30</u>
2.14.Hàm xử lý sự kiện Lưu lại ván chơi	<u>30</u>
2.15.Hàm bắt đầu trò chơi:	<u>32</u>
TÀI LIỆU THAM KHẢO	<u>33</u>

DANH MỤC HÌNH ẢNH

Hình 1 Giao diện chơi game	5
Hình 2 Form đăng ký	5
Hình 3 Form đăng nhập	6
Hình 4 Game đang chơi	6
Hình 5 Dừng game.....	8
Hình 6 Sơ đồ tuần tự	7
Hình 7 Cơ sở dữ liệu.....	9
Hình 8 Quản lý dữ liệu	10
Hình 9 Cấu trúc dữ liệu bảng ball	10
Hình 10 Cấu trúc dữ liệu bảng brick.....	11
Hình 11 Cấu trúc dữ liệu bảng Paddle	11
Hình 12 Cấu trúc dữ liệu bảng Score.....	11
Hình 13 Giao diện chơi game – Xây dựng giao diện	13
Hình 14 Thư viện sử dụng	14
Hình 15 Xây dựng trang chơi game chính	14
Hình 16 Css trang trí game	16
Hình 17 Xây dựng trang login	15
Hình 18 Lấy dữ liệu từ SQL	16
Hình 19 Up dữ liệu lên SQL	16
Hình 20 Xử lý sự kiện đăng ký, đăng nhập - 1	17
Hình 21 Xử lý sự kiện đăng ký, đăng nhập - 2	18
Hình 22 Hàm con thực hiện chức năng với database - 1	19
Hình 23 Hàm con thực hiện chức năng với database - 2	20
Hình 24 Khai báo các biến.....	21
Hình 25 Setup gạch.....	22
Hình 26 Reset paddle	23
Hình 27 Reset ball.....	23
Hình 28 Sự kiện dừng game	24
Hình 29 Sự kiện bắt đầu game.....	24
Hình 30 Sự kiện di chuyển paddle.....	25
Hình 31 Vẽ paddle	25
Hình 32 Vẽ ball.....	26
Hình 33 Tính điểm	26
Hình 34 Vẽ gạch	27
Hình 35 Xử lý va chạm banh và tường.....	27
Hình 36 Xử lý va chạm banh và paddle.....	28
Hình 37 Xử lý va chạm banh và gạch.....	29
Hình 38 Tạo mới, thêm gạch	30
Hình 39 Xử lý sự kiện chơi game	31
Hình 40 Bắt đầu trò chơi.....	32

CHƯƠNG 1: GIỚI THIỆU GAME

1. Thành phần game

Tập hợp các khối màu xanh tạo thành 1 hàng rào

1 chấm vuông màu đen

1 thanh màu đỏ.

2. Quy tắc chơi

Chấm đen sẽ di chuyển từ dưới lên phía trên chạm vào khối màu xanh nào sẽ làm khối, màu xanh đó biến mất. Sau đó chấm đen này lại rơi xuống dưới theo quỹ đạo lên xuống tự do không theo 1 quy luật nào.

Trong quá trình chơi bạn có thể tạm dừng và tiếp tục sau đó.

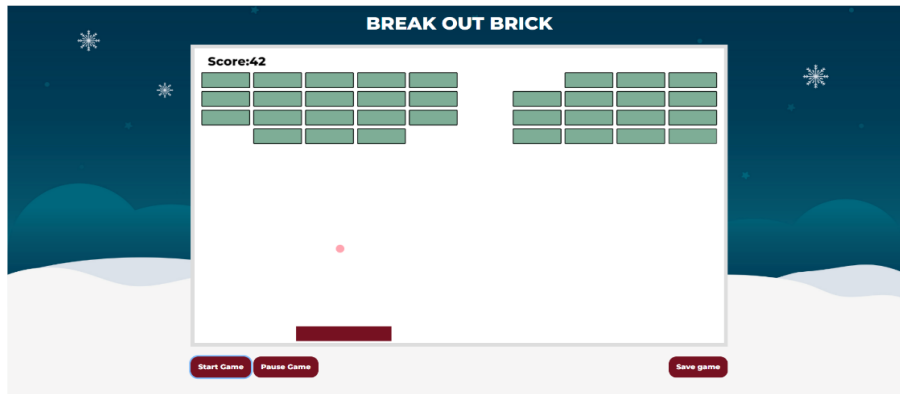
3. Cách chơi

Di chuyển thanh màu đỏ (sang trái hoặc phải) để chắn chấm màu đen không cho nó tiếp xúc với viền biên phía dưới.

Nếu bạn không chắn được chấm màu đen rơi xuống thì sẽ "Game over" - trò chơi kết thúc

4. Tính năng

- Chơi game



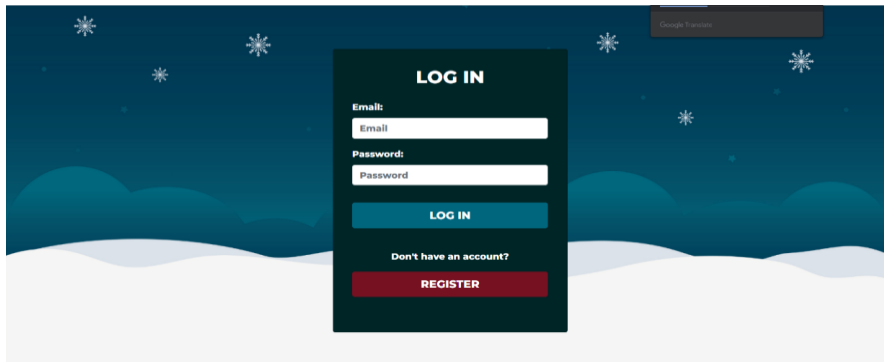
Hình 1 Giao diện chơi game

- Tạo tài khoản
 - + Nhập tên tài khoản
 - + Địa Chỉ email
 - + Nhập Mật khẩu
 - + Nhấn Đăng ký và Tạo Tài khoản Thành công

Hình 2 Form đăng ký

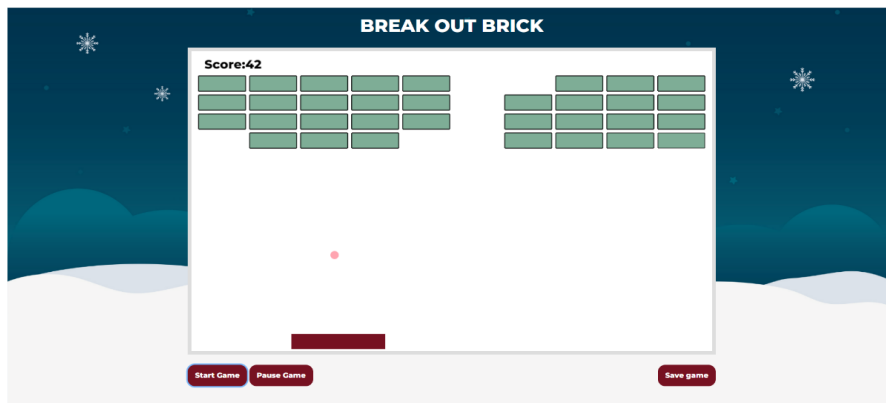
Formatted: Caption, Centered, Line spacing: single

- Đăng nhập
 - + Nhập Email đã tạo
 - + Nhập Mật khẩu



Hình 3 Form đăng nhập

- Lưu Trò chơi
 - + Lưu lại điểm số và trạng thái của trò chơi
 - + Đăng lập tài khoản và tiếp tục chơi với điểm số và trạng thái đã lưu

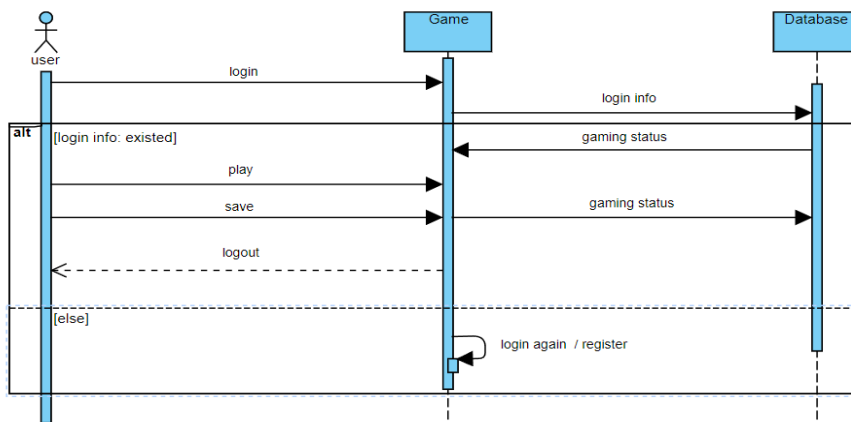


Hình 4 Game đang chơi

Formatted: Caption, Centered, Line spacing: single

- Pause
 - + Tạm dừng trò chơi đang chơi
 - + Click vào nút Start game để tiếp tục chơi tiếp

5. Sơ đồ tuần tự



Hình 6 Sơ đồ tuần tự

CHƯƠNG 2: ỨNG DỤNG VÀ NGÔN NGỮ LẬP TRÌNH

1. PHP

- Ngôn ngữ PHP là từ viết tắt của Personal Home Page (hiện nay là Hypertext Preprocessor). Thuật ngữ này chỉ chuỗi ngôn ngữ kịch bản hay mã lệnh, phù hợp để phát triển cho các ứng dụng nằm trên máy chủ.
- Khi viết phần mềm bằng ngôn ngữ PHP, chuỗi lệnh sẽ được xử lý trên server để từ đó sinh ra mã HTML trên client. Và dựa vào đó, các ứng dụng trên website của bạn sẽ hoạt động một cách dễ dàng.

2. HTML

- **HTML** là chữ viết tắt của **Hypertext Markup Language**. Nó giúp người dùng tạo và cấu trúc các thành phần trong trang web hoặc ứng dụng, phân chia các đoạn văn, heading, links, blockquotes, vâng vâng.
- **HTML** không phải là ngôn ngữ lập trình, đồng nghĩa với việc nó không thể tạo ra các chức năng “động” được. Nó chỉ giống như Microsoft Word, dùng để bố cục và định dạng trang web.

3. Javascript

- **JavaScript**, theo phiên bản hiện hành, là một ngôn ngữ lập trình thông dịch được phát triển từ các ý niệm nguyên mẫu. Ngôn ngữ này được dùng rộng rãi cho các trang web cũng như phía máy chủ.

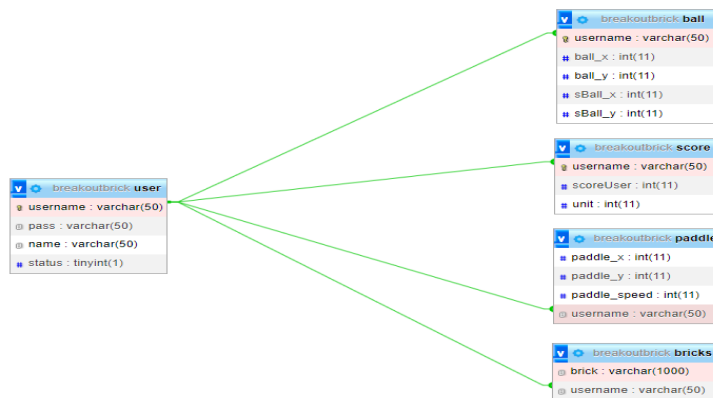
4. Bootstrap 4

- Bootstrap.css là một framework CSS sắp xếp và quản lý bố cục của trang web. Trong khi HTML quản lý nội dung và cấu trúc của trang web, CSS xử lý bố cục của trang web. Vì lý do đó, cả hai cấu trúc cần cùng tồn tại để thực hiện một hành động cụ thể
- Bootstrap.js là phần cốt lõi của Bootstrap. Nó bao gồm các file JavaScript chịu trách nhiệm cho việc tương tác của trang web.

CHƯƠNG 3: CƠ SỞ DỮ LIỆU

1. Bảng cơ sở dữ liệu

Cơ sở dữ liệu làm nhiệm vụ lưu trữ các dữ liệu về trạng thái chơi game của người dùng. CSDL gồm 5 bảng, mỗi bảng sẽ có một khóa chính và các bảng sẽ liên kết với nhau bằng khóa ngoại. Khi tìm kiếm dữ liệu, thay đổi dữ liệu người viết code sẽ dựa vào khóa username

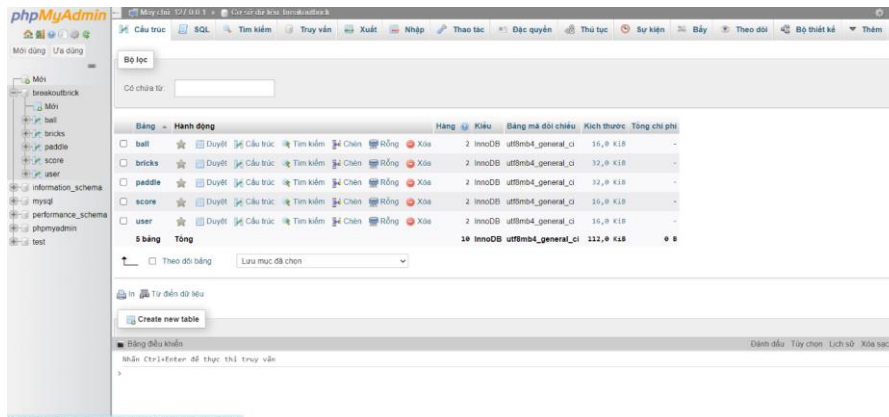


Hình 7 Cơ sở dữ liệu

2. Quản lý dữ liệu

2. 1. Quản lý dữ liệu trò chơi và người dùng với phpMyAdmin

- Quản lý user(người dùng): thêm, xóa, sửa(phân quyền).
- Quản lý cơ sở dữ liệu: tạo mới, xóa, sửa, thêm bảng, hàng, trường, tìm kiếm đối tượng.
- Nhập xuất dữ liệu(Import/Export): hỗ trợ các định dạng SQL, XML và CSV.
- Thực hiện các truy vấn MySQL, giám sát quá trình và theo dõi.
- Sao lưu và khôi phục(Backup/Restore): Thao tác thủ công.



Hình 8 Quản lý dữ liệu

Formatted: Caption, Centered, Line spacing: single

2. 2. Quản lý Ball

Quản lý cơ sở dữ liệu Ball: tạo mới, xóa.Sửa thông tin cột (tên cột, kiểu giá trị, khóa chính, ...)

#	Tên	Kiểu	Bảng mã đối chiếu	Thuộc tính	Null	Mặc định	Ghi chú	Thêm	Hành động
<input type="checkbox"/>	1	username	varchar(50)	utf8mb4_general_ci	Không	Không		Thay đổi	Xóa Thêm
<input type="checkbox"/>	2	ball_x	int(11)		Không	0		Thay đổi	Xóa Thêm
<input type="checkbox"/>	3	ball_y	int(11)		Không	0		Thay đổi	Xóa Thêm
<input type="checkbox"/>	4	sBall_x	int(11)		Không	4		Thay đổi	Xóa Thêm
<input type="checkbox"/>	5	sBall_y	int(11)		Không	4		Thay đổi	Xóa Thêm

Hình 9 Cấu trúc dữ liệu bảng ball

Formatted: Caption, Centered, Line spacing: single

2.3. Quản lý Brick

Quản lý cơ sở dữ liệu Brick: tạo mới, xóa. Sửa thông tin cột (tên cột, kiểu giá trị, khóa chính, ...)

Cấu trúc bảng

Hiện thị quan hệ

	#	Tên	Kiểu	Bảng mã đối chiếu	Thuộc tính	Null	Mặc định	Ghi chú	Thêm	Hành động
<input type="checkbox"/>	1	brick	varchar(1000)	utf8mb4_general_ci		Có	NULL		 Thay đổi	 Xóa  Thêm
<input type="checkbox"/>	2	username 	varchar(50)	utf8mb4_general_ci		Có	NULL		 Thay đổi	 Xóa  Thêm

Hình 10 Cấu trúc dữ liệu bảng brick

Formatted: Caption, Centered, Line spacing: single

2.4. Quản lý Paddle

Quản lý cơ sở dữ liệu Paddle: tạo mới, xóa. Sửa thông tin cột (tên cột, kiểu giá trị, khóa chính, ...)

	#	Tên	Kiểu	Bảng mã đối chiếu	Thuộc tính	Null	Mặc định	Ghi chú	Thêm	Hành động
<input type="checkbox"/>	1	paddle_x	int(11)			Không	0		 Thay đổi	 Xóa  Thêm
<input type="checkbox"/>	2	paddle_y	int(11)			Không	0		 Thay đổi	 Xóa  Thêm
<input type="checkbox"/>	3	paddle_speed	int(11)			Không	14		 Thay đổi	 Xóa  Thêm
<input type="checkbox"/>	4	username 	varchar(50)	utf8mb4_general_ci		Không	Không		 Thay đổi	 Xóa  Thêm

Hình 11 Cấu trúc dữ liệu bảng Paddle

Formatted: Caption, Centered

2.5. Quản lý Score

Quản lý cơ sở dữ liệu Score: tạo mới, xóa. Sửa thông tin cột (tên cột, kiểu giá trị, khóa chính, ...)

	#	Tên	Kiểu	Bảng mã đối chiếu	Thuộc tính	Null	Mặc định	Ghi chú	Thêm	Hành động
<input type="checkbox"/>	1	username	varchar(50)	utf8mb4_general_ci		Không	Không			 Thay đổi  Xóa  Thêm
<input type="checkbox"/>	2	scoreUser	int(11)			Không	0			 Thay đổi  Xóa  Thêm
<input type="checkbox"/>	3	unit	int(11)			Không	3			 Thay đổi  Xóa  Thêm

Hình 12 Cấu trúc dữ liệu bảng Score

Formatted: Caption, Centered, Line spacing: single

2. 6. Quản lý tài khoản

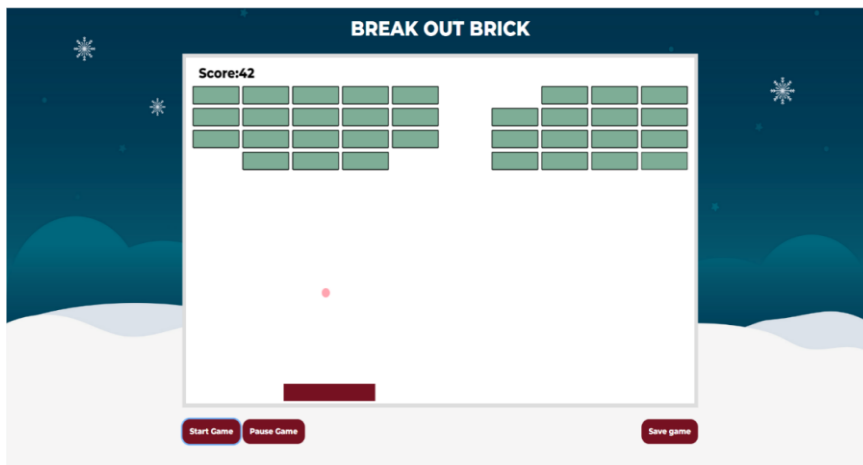
Quản lý cơ sở dữ liệu tài khoản : tạo mới, xóa.Sửa thông tin cột (tên cột, kiểu giá trị, khóa chính, ...)

CHƯƠNG 4: PHÁT TRIỂN GAME

1. Xây dựng giao diện Game

1.1. Trang chủ

- Giao diện Trang chủ gồm có các ô gạch, pallet và ball để người dùng tương tác trò chơi
- Pause: Tạm dừng trò chơi lại
- Save : Lưu lại điểm số và trạng thái chơi
- Start: Trở về trạng thái mới và chơi lại từ đầu
- Score: Điểm số người chơi



Hình 13 Giao diện chơi game – Xây dựng giao diện

- Xây dựng giao diện game bằng HTML5, JAVASCRIPT S, CSS
- Thư viện sử dụng BOOTSTRAP 4 VÀ JQUERY

```
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Break Out Brick with HTML5, JS</title>
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css" integrity="sha384-Gn5384xqQIAxXIA38V1gJE1rt8Ggg6MRQ98V6d6I16q7dbS21p2WXLsF1-v" crossorigin="anonymous">
<script src="https://code.jquery.com/jquery-3.3.1.min.js" integrity="sha256-FgpCb/KatQnnsZab6jbV1b9EpIQz9o1lw-POpdxt6d1CMv" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1Q2t3Z3A77N83Dh92vRt67l92Dp0+s7NIUB4j9384f4K8346" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/js/bootstrap.min.js" integrity="sha384-JZR6dg82UOZ98p2hzZhpVj1+z7i7ocV6s+L1720beut0f/vu10pLU0peioaY36" crossorigin="anonymous"></script>
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Montserrat:wght@800&display=swap" rel="stylesheet">
<link rel="stylesheet" href="/css/style.css">
</head>
```

— Trang trí bằng css

1.2.Trang đăng nhập

- Sử dụng các tài khoản đã lập trước đó và đăng nhập để bắt đầu mới hoặc tiếp tục chơi game đã lưu
- Nhập các thông tin bao gồm : Name, password
- Nếu chưa có có đăng ký tài khoản

```
<body>
<div class="container card">
  <h2 class="text-center mb-4">LOG IN</h2>
  <form method="post" action="">
    <div class="form-group">
      <label for="email">Email:</label>
      <input type="email" name="email" class="form-control" id="email" placeholder="Email" required>
    </div>
    <div class="form-group">
      <label for="password">Password:</label>
      <input type="password" name="pass" class="form-control" id="password" placeholder="Password" required>
    </div>
    <div class="form-group">
      <button id="login" type="submit" class="btn my-3 p-2" style="width: 100%;" name="login">LOG IN</button>
    </div>
    <p class="text-center mb-0 mt-4">Don't have an account?</p>
    <div class="form-group">
      <!-- <button id = "register" class="btn my-3 p-2" style="width: 100%;"><a href="register.php">REGISTER</a></button> -->
      <a href="register.php" id="register" class="btn my-3 p-2" style="width: 100%;">REGISTER</a>
    </div>
    <p class="announce"><?php echo $announce; ?></p>
  </form>
</div>
```

Hình 17 Xây dựng trang login

1.3.Trang Đăng ký

Thiết kế trang đăng ký với các thông tin yêu cầu cơ bản:

- Name
- Địa chỉ Email
- Password

1.4.Trang Getdata

- Trang thực thi lấy dữ liệu từ SQL

```

1  <?php
2  // session_start();
3  include_once('./submit.php');
4  include_once('./utils.php');
5  $username = $_SESSION['username'];
6  // echo json_encode($_SESSION);
7
8  $conn = connectData();
9  $statement = "SELECT *
10 FROM user
11     LEFT JOIN ball
12     ON ball.username = user.username
13     LEFT JOIN paddle
14     ON paddle.username = user.username
15     LEFT JOIN score
16     ON score.username = user.username
17     LEFT JOIN bricks
18     ON bricks.username = user.username
19     WHERE user.username = '$username'";
20 $query = mysqli_query($conn, $statement);
21 $data = mysqli_fetch_array($query);
22
23 die(json_encode($data));
24
25 // die(json_encode($data));
26

```

Hình 18 Lấy dữ liệu từ SQL

Formatted: Caption, Centered, Line spacing: single

1.5.Trang Uploaddata

uploaddata.php là up dữ liệu của người dùng khi save game lên sql

```

1  <?php
2  include_once('./submit.php');
3  include_once('./utils.php');
4
5  $ball = getPOST('ball');
6  $paddle = getPOST('paddle');
7  $score = getPOST('dataScore');
8  $bricks = getPOST('brick');
9  $username = $_SESSION['username'];
10 $dataBrick = "";
11
12 if ($ball != ""){
13     $dataBall = "ball_x = ".$ball['x'].", ball_y = ".$ball['y'].", sball_x = ".$ball['speed']['x'].",
14     sball_y = ".$ball['speed']['y'];
15     $dataPaddle = "paddle_x = ".$paddle['x'].", paddle_y = ".$paddle['y'].", paddle_speed = ".$paddle
16     ['speed'];
17     $dataScore = "scoreUser = ".$score['s'] . ", unit = ".$score['unit'];
18 }
19
20 updateData("ball",$dataBall,$_SESSION['username']);
21 updateData("paddle",$dataPaddle,$_SESSION['username']);
22 updateData("score",$dataScore,$_SESSION['username']);
23 updateData("bricks","brick = '$bricks'",$_SESSION['username']);
24

```

Hình 19 Up dữ liệu lên SQL

Formatted: Caption, Centered, Indent: First line: 0 cm, Line spacing: single

1.6.Trang Submit

Submit.php là xử lý sự kiện click nút login và register lưu và xác nhận tài khoản game của người dùng.

```
<?php
session_start();
require_once('../utils.php');

$action = getPOST('action');

switch ($action) {
    case 'login':
        login();
        break;
    case 'register':
        register();
        break;
}

function register()
{
    $email = getPOST('email');
    $pass = getPOST('pass');
    $name = getPOST('name');

    $pass = md5($pass);

    $values = '$email','$pass','$name';
    $valuedefault = '$email';

    if (checkAccountRegis($email)) {
        addData('user', 'username,pass,name', $values);
        addData('ball', 'username', '$email');
        addData('paddle', 'username', '$email');
        addData('score', 'username', '$email');
    }

    $xBrick = 14;
```

Hình 20 Xử lý sự kiện đăng ký, đăng nhập - 1

```

    $yBrick = 50;
    $widthBrick = 90;
    $heightBrick = 30;
    $value = "";
    for ($number = 1; $number < 5; $number++) {
        while ($xBrick + $widthBrick < 1000) {
            $value = $value . "xBrick $yBrick ";
            $xBrick += $widthBrick + 8; //mỗi cục gạch cách nhau 8px
        }
        $xBrick = 14;
        $yBrick += $heightBrick + 8; //mỗi hàng gạch cách nhau 8px
    }
    addData("bricks", "brick,username", "'$value','$email'");
    return "You have successfully registered.<br/> Please login to start game.";
}
return "The account has been created.";
}
if (isset($_POST['register'])) {
    $announce = register();
}
function login()
{
    $email = $_POST['email'];
    $pass = $_POST['pass'];
    $pass = md5($pass);

    if (checkAccountLogin($email, $pass)) {
        $_SESSION['username'] = $email;
        return "";
    }
    return "Email or Password is incorrect.";
}
if (isset($_POST['login'])) {
    $announce = login();
}

```

Hình 21 Xử lý sự kiện đăng ký, đăng nhập - 2

1.7. Trang utils

Trang có nhiệm vụ xử lý từ php sang mysql chứa các hàm con thực hiện các chức năng nhỏ như thêm data , connect sql,...

```

<?php

function getPOST($key)
{
    if (isset($_POST[$key])) {
        $result = $_POST[$key];
    } else {
        $result = '';
    }
    return $result;
}

function connectData()
{
    $conn = mysqli_connect("localhost", "root", "", "breakoutbrick");
    mysqli_set_charset($conn, 'utf8');
    return $conn;
}

function getAllData($user)
{
    $conn = connectData();

    $data = [];

    $dbBall = "SELECT * FROM ball WHERE username = '$user'";
    $qBall = mysqli_query($conn, $dbBall);
    array_push($data, mysqli_fetch_array($qBall, MYSQLI_ASSOC));

    $dbPaddle = "SELECT * FROM paddle WHERE username = '$user'";
    $qPaddle = mysqli_query($conn, $dbPaddle);
    array_push($data, mysqli_fetch_array($qPaddle, MYSQLI_ASSOC));

    $dbScore = "SELECT * FROM score WHERE username = '$user'";
    $qScore = mysqli_query($conn, $dbScore);
    array_push($data, mysqli_fetch_array($qScore, MYSQLI_ASSOC));

    mysqli_close($conn);

    return $data;
}

function getData($statement, $getData)
{
    $conn = connectData();
    $data = "";

    $query = mysqli_query($conn, $statement);

```

Hình 22 Hàm con thực hiện chức năng với database - 1


```

    if ($getData == 1) {
        $data = mysqli_fetch_array($query, MYSQLI_ASSOC);
    }
    mysqli_close($conn);
    return $data;
}

function selectData($user, $table, $id,)
{
    $statement = "SELECT * FROM $table WHERE $id = '$user'";
    return getData($statement, 1);
}

function addData($table, $key, $values)
{
    $statement = "INSERT INTO $table ($key) VALUES ($values)";
    getData($statement, 0);
}

function updateData($table, $values, $user)
{
    $statement = "UPDATE $table SET " . $values . " WHERE username = '$user'";
    getData($statement, 0);
}

function checkAccountRegis($user)
{
    $data = selectData($user, 'user', 'username');
    if ($data == "") {
        return true;
    }

    return false;
}

function checkAccountLogin($user, $pass)
{
    $data = selectData($user, 'user', 'username');

    if ($data == null) {
        return false;
    } else {
        if ($data['pass'] == $pass) {
            header('location: index.php');
            return true;
        } else {
            return false;
        }
    }
}

```

Hình 23 Hàm con thực hiện chức năng với database - 2

2. Xử lý sự kiện với JavaScript

- Sử dụng Canvas vẽ hình ảnh gạch, bàn, paddle chuyển động trong trò chơi
- Các chuyển động được tạo nên bởi hiệu ứng khung hình được xóa đi và vẽ lại
- id là thuộc tính để chúng ta có thể truy xuất được đối tượng canvas này từ mã JavaScript (thông qua phương thức `document.getElementById()`).
- Thiết lập trạng thái game qua hàm `Let statusGame`
- `let moveRight = false`; đánh dấu sự kiện nhấn phím phải di chuyển paddle sang phải
- `let moveLeft = false`; đánh dấu sự kiện nhấn phím trái di chuyển paddle sang trái

```
const canvas = document.getElementById("myCanvas");
const c = document.getElementById("myCanvas");
const ctx = c.getContext("2d");
const WIDTHPADDLE = 180;
const HEIGHTPADDLE = 30;
const RADIUSBALL = 8;
const UNITSCORE = 3;
const SPEEDBALLDEFAULT = 4;
let statusGame = {
  start: true,
  pause: false,
  end: false,
};
let paddle;
let ball;
let bricks = [];
let score = 0;
let moveRight = false; //đánh dấu sự kiện nhấn phím phải di chuyển paddle sang phải
let moveLeft = false; //đánh dấu sự kiện nhấn phím trái di chuyển paddle sang trái
```

Hình 24 Khai báo các biến

Formatted: Caption, Centered, Tab stops: Not at 11.25 cm

2.1.Hàm `SetupBricks()`

- Hàm được dùng để xử lý với gạch là một chuỗi số
- Lấy ra và cộng dồn đến khi nào gặp dấu /t thì chuyển sang num, đánh dấu chuyển x – y, nếu là x thì add vào x ngược lại add vào y

```

function setupBricks(){
  let lenBrickDatabase = data.brick.length;
  let j = 0; let flag = 1;
  while (j < lenBrickDatabase) {
    let num = "";
    while (data.brick[j] !== " "){
      num += data.brick[j];
      j++;
    }
    if (data.brick[j] === " ") {
      if (flag > 0) { // tọa độ của x
        bricksdatabase.push(...items:{
          x : Number(value: num),
          y : 0,
          width : 90,
          height : 30
        });
      }
      else{ //tọa độ y
        bricksdatabase[bricksdatabase.length-1].y = Number(value: num);
      }
      j++;
      flag = -flag;
    }
  }
  console.log(...data: "bricks database: " + bricksdatabase.length);
}

```

Hình 25 Setup gạch

2.2.Hàm ResetPaddle

Reset paddle trở về vị trí và kích thước ban đầu

```
function resetPaddle() {
  paddle = {
    x: Number(value: data.paddle_x),
    y: Number(value: data.paddle_y),
    speed: Number(value: data.paddle_speed),
  };
}
```

Hình 26 Reset paddle

2.3.Hàm resetBall

```
function resetBall() {
  ball = {
    x: Number(value: data.ball_x),
    y: Number(value: data.ball_y),
    speed: {
      x: Number(value: data.sBall_x),
      y: Number(value: data.sBall_y),
    },
  };

  if (data.ball_x == "0") {
    //check first play
    paddle = {
      x: canvas.width / 2,
      y: canvas.height - HEIGHTPADDLE - 4,
      speed: Number(value: data.paddle_speed),
    };
    ball.x = paddle.x + WIDTHPADDLE / 2;
    ball.y = paddle.y;
  }
}
```

Hình 27 Reset ball

2.4.Hàm xử lý sự kiện trạng thái của game

Sự kiện tạm dừng Game với hàm \$("#pause").on("click", function (event)

```

$("#pause").on("click", function (event) {
    statusGame.start = false;
    statusGame.pause = true;
    $("#myCanvas").addClass("pause");
    clearTimeout(start);
});

```

Hình 28 Sự kiện dừng game

Sự kiện bắt đầu chơi với : `$("#start").on("click", function (event)`

```

$("#start").on("click", function (event) {
    if ($("#myCanvas").hasClass("pause")) {
        $("#myCanvas").removeClass("pause");
    }
    if ($("#myCanvas").hasClass("endgame")) {
        $("#myCanvas").removeClass("endgame");
    }
    if (!statusGame.pause && !statusGame.end) {
        clearTimeout(start);
        resetGame();
    }
    statusGame.pause = false;
    startGame();
});

```

Hình 29 Sự kiện bắt đầu game

2.5.Hàm bắt Sự kiện di chuyển Paddle

```
window.addEventListener(  
  type: "keydown",  
  listener: function (event) {  
    if (event.defaultPrevented) {  
      return;  
    }  
    if (event.key === "ArrowRight") {  
      moveRight = true;  
    } else if (event.key === "ArrowLeft") {  
      moveLeft = true;  
    }  
  },  
  options: true  
);
```

Hình 30 Sự kiện di chuyển paddle

2.6.Hàm vẽ Paddle trong game

```
function drawPaddle() {  
  ctx.beginPath();  
  ctx.rect(paddle.x, paddle.y, WIDTHPADDLE, HEIGHTPADDLE);  
  ctx.fillStyle = "#761322";  
  ctx.fill();  
}
```

Hình 31 Vẽ paddle

2.7.Hàm vẽ ball tronng game

```
function drawBall() {  
  ctx.beginPath();  
  ctx.arc(ball.x, ball.y, RADIUSBALL, 0, 2 * Math.PI);  
  ctx.fillStyle = "#FEA4B0";  
  ctx.fill();  
}
```

Hình 32 Vẽ ball

2.8.Hàm vẽ bảng tính điểm

```
function drawScore() {  
  ctx.fillStyle = "black";  
  ctx.font = "1.5rem Montserrat";  
  ctx.fillText("Score:" + score, 25, 35);  
}
```

Hình 33 Tính điểm

2.9.Hàm vẽ gạch

```
function drawBrick() {
  ctx.beginPath();

  let lenBrick = bricks.length;
  for (number = 0; number < lenBrick; number++) {
    ctx.rect(
      bricks[number].x,
      bricks[number].y,
      bricks[number].width,
      bricks[number].height
    );
    ctx.fillStyle = "#7EAD96";
    ctx.fill();
    ctx.stroke();
  }
}
```

Hình 34 Vẽ gạch

2.10.Hàm xử lý va chạm ball và wall

```
function ballCollisionWithWall() {
  if (ball.x ≥ canvas.width - RADIUSBALL || ball.x ≤ 0) {
    ball.speed.x = -ball.speed.x;
  }
  if (ball.y ≤ 0) {
    ball.speed.y = -ball.speed.y;
  }
  if (ball.y ≥ canvas.height - RADIUSBALL) {
    statusGame.end = true;
  }
}
```

Hình 35 Xử lý va chạm banh và tường

2.11. Hàm xử lý va chạm ball và paddle

- Ball chạm vào pallet thì chuyển hướng của y, tại vị trí hai bên paddle ball nảy đi với góc 60 độ.
- Áp dụng tỷ số lượng giác của góc nhọn trong tam giác vuông ta tính được khoảng cách cần cộng thêm cho ball

```
function ballCollisionWithPaddle() {  
  if (  
    ball.y + RADIUSBALL > paddle.y &&  
    ball.y - RADIUSBALL < paddle.y + HEIGHTPADDLE &&  
    ball.x + RADIUSBALL > paddle.x &&  
    ball.x - RADIUSBALL < paddle.x + WIDTHPADDLE  
  ) {  
    ball.speed.y = -ball.speed.y;  
    ball.y = paddle.y - RADIUSBALL;  
  }  
}
```

Hình 36 Xử lý va chạm banh và paddle

2.12. Hàm xử lý va chạm với gạch

```
function ballCollisionWithBrick() {  
  for (let b = 0; b < bricks.length; b++) {  
    if (  
      ball.x + RADIUSBALL > bricks[b].x &&  
      ball.x - RADIUSBALL < bricks[b].x + bricks[b].width &&  
      ball.y + RADIUSBALL > bricks[b].y &&  
      ball.y - RADIUSBALL < bricks[b].y + bricks[b].height  
    ) {  
      let numRan = Math.floor(Math.random() * 4) - 2;  
      ball.speed.x = -(SPEEDBALLDEFAULT + numRan);  
      ball.speed.y = -(SPEEDBALLDEFAULT - numRan);  
      bricks.splice(start: b, deleteCount: 1);  
      score += UNITSCORE;  
    }  
  }  
}
```

Hình 37: Xử lý va chạm bóng và gạch

2.13. Hàm tạo mới thêm gạch

```
function addBrick(){
    console.log(...data: "bricks database before: "+ bricksdatabase.length);
    bricks = bricksdatabase.slice(start: 0, end: bricksdatabase.length);
    console.log(...data: "bricks : "+ bricks.length);
    console.log(...data: "bricks database after: "+ bricksdatabase.length);

    if (bricks.length == 0){
        // Vẽ gạch từ tọa độ x=14, y=10. Mỗi cục gạch có kích thước 90x30px;
        // Có 4 hàng gạch, vẽ mỗi cục gạch đến khi cục gạch thứ n tràn ra ngoài khung game
        let xBrick = 14;
        let yBrick = 50;
        let heightBrick = 30;
        let widthBrick = 90;
        bricks = [];
        for (number = 1; number < 5; number++) {
            while (xBrick + widthBrick < 1000) {
                bricks.push(...items: {
                    x: xBrick,
                    y: yBrick,
                    height: heightBrick,
                    width: widthBrick,
                });
                xBrick += widthBrick + 8; //mỗi cục gạch cách nhau 8px
            }
            xBrick = 14;
            yBrick += heightBrick + 8; //mỗi hàng gạch cách nhau 8px
        }
    }
}
```

Hình 38 Tạo mới, thêm gạch

2.14. Hàm xử lý sự kiện Lưu lại ván chơi

Phương thức “ ajax() ” chính là một đối tượng (Object) gồm các thuộc tính cấu hình để.

Trong đó:

- url : chuỗi chứa đường dẫn tới “./uploaddata.php” cần lấy và trả về dữ liệu
- type : Sử dụng phương thức POST truyền dữ liệu vào
- data : truyền dữ liệu sang ball, paddle, dataScore, brick thực hiện xử lý và trả về dữ liệu
- success: Một hàm được gọi khi request thành công.

```
//Save game
$("#save").on("click", function (event) {
    dataScore = {
        s: score,
        unit: UNITSCORE,
    };
    let brick = "";
    for (let i = 0; i < bricks.length; i++) {
        brick += bricks[i].x.toString() + " " + bricks[i].y.toString() + " ";
    }

    $.ajax({
        type: "POST",
        url: "./uploaddata.php",
        data: { ball, paddle, dataScore, brick },
        success: function (data) {},
    });
});
}
```

Hình 39 Xử lý sự kiện chơi game

2.15.Hàm bắt đầu trò chơi:

```
function startGame() {
  if (statusGame.end && !statusGame.pause) {
    statusGame.end = false;
    resetGame();
  }

  ctx.clearRect(0, 0, canvas.width, canvas.height);
  drawScore();
  drawPaddle();
  drawBall();
  movePaddle();

  ballCollisionWithPaddle();
  ballCollisionWithWall();
  ballCollisionWithBrick();
  ball.x += ball.speed.x;
  ball.y -= ball.speed.y;

  drawBrick();
  // Xử lí pause và start game
  start = setTimeout(handler: startGame, timeout: 10);

  // Dừng game
  if (statusGame.end) {
    $("#myCanvas").addClass("endgame");
    statusGame.start = false;
    clearTimeout(id: start);
    resetGame();
  }
}
```

Hình 40 Bắt đầu trò chơi

TÀI LIỆU THAM KHẢO

1. Tiếng Việt
 - <https://glints.com/vn/blog/lap-trinh-php-la-gi/#.Y5DDUHbP3IU>
 - <https://viblo.asia/p/jquery-ajax-va-kien-thuc-co-ban-4dbZNxnv5YM>
2. Tiếng anh
 - <https://getbootstrap.com/docs/4.0/getting-started/introduction>