**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**



**ĐỒ ÁN MÔN HỌC**

**ĐỒ ÁN CÔNG NGHỆ THÔNG TIN**

**ĐỀ TÀI: LẬP TRÌNH WEB GAME**

GVHD: Thầy Trần Công Tú

SVTH: Trần Nhật Hùng 19110213

Nguyễn Hoàng Phúc 19110268

TP. HCM, tháng 11 năm 2021

# **LỜI NÓI ĐẦU**

Đầu tiên, chúng em xin chân thành cảm ơn thầy Trần Công Tú đã phụ trách môn học và hướng dẫn chúng em thực hiện dự án này. Thầy đã đóng góp những ý kiến và hướng dẫn có giá trị để chúng em hoàn thành đồ án môn học và giúp hiểu được những vấn đề phức tạp liên quan đến việc lập kế hoạch để hoàn thành đồ án một cách hiệu quả. Giúp chúng em ôn lại những kiến thức đã học trong các năm qua. Chúng em cũng đã có thể tự học những kiến thức mới và áp dụng vào đồ án môn học, giúp chúng em có những cái nhìn mới về kiến thức ngành CNTT.

Đồ án môn học được thực hiện trong vòng mười lăm tuần. Tuy nhiên do kiến ​​thức mới nhiều cũng như thời gian chúng em làm qua mỗi tuần chưa tối ưu nên đồ án sẽ có nhiều sai sót là điều không thể tránh khỏi. Rất mong nhận được mọi sự đóng góp ý kiến ​​của thầy để giúp cho những kiến ​​thức còn hạn chế của chúng em được hoàn thiện hơn.

Chân thành cảm ơn.

**MỤC LỤC**

[**LỜI NÓI ĐẦU** 2](#_Toc88932602)

[**PHẦN 1. MỞ ĐẦU** 4](#_Toc88932603)

[**1.1.** **Khảo sát** 4](#_Toc88932604)

[**1.2.** **Đặt vấn đề** 5](#_Toc88932605)

[**1.2.1.** **Lý do chọn đề tài** 5](#_Toc88932606)

[**1.2.2.** **Mục tiêu đề tài** 5](#_Toc88932607)

[**PHẦN 2. NỘI DUNG** 6](#_Toc88932608)

[**2.1.** **Giới thiệu** 6](#_Toc88932609)

[**2.2.** **Phân tích yêu cầu** 6](#_Toc88932610)

[**2.3.** **Kiến thức liên quan** 6](#_Toc88932611)

[**2.4.** **Giao diện** 6](#_Toc88932612)

[**2.4.1.** **Header** 6](#_Toc88932613)

[**2.4.2.** **Body** 6](#_Toc88932614)

[**2.4.3.** **Footer** 6](#_Toc88932615)

[**2.5.** **Mã nguồn** 6](#_Toc88932616)

[**2.5.1.** **Client** 6](#_Toc88932617)

[**2.5.2.** **Server** 6](#_Toc88932618)

[**2.6.** **Khó khăn** 6](#_Toc88932619)

[**PHẦN 3. KẾT LUẬN** 7](#_Toc88932620)

[**3.1.** **Kết quả đạt được** 7](#_Toc88932621)

[**3.2.** **Ưu nhược điểm** 7](#_Toc88932622)

[**TÀI LIỆU THAM KHẢO** 8](#_Toc88932623)

# **PHẦN 1. MỞ ĐẦU**

## **Khảo sát**

### **Tham khảo chủ đề trò chơi**

* Dựa trên game khủng long vượt chướng ngại vật nổi tiếng khi không có internet trên trình duyệt.

Chart

Description automatically generated

* Cơ chế chủ yếu của game là người chơi sẽ nhấn nút để chú khủng long nhảy khi có vật cản lao tới, nếu đụng trúng vật cản sẽ thua. Mục tiêu của người chơi là vượt vật cản trong thời gian lâu nhất có thể.

**Ưu điểm:**

* Có thể giải trí khi không có internet.

**Nhược điểm:**

* Game thô sơ, dễ gây chán khi chơi lâu.
* Mục tiêu của game đơn giản, không thú vị.
* Giao diện game không đẹp mắt.

### **Tham khảo giao diện**

Tham khảo từ trang web: <https://play2048.co>.

Calendar

Description automatically generated

Bố cục: màn chơi ở giữa khung hình. Thông tin, điểm số ở trên và dưới màn chơi.

**Ưu điểm:**

* Dễ nhìn, không làm người chơi xao nhãng khỏi trò chơi.
* Lưu lại trạng thái trò chơi sau khi thoát khỏi trang web.
* Lưu lại điểm cao của người chơi.

**Nhược điểm:**

* Nếu có nhiều người chơi luân phiên trên cùng một máy tính thì sẽ phải dùng chung profile.

## **Đặt vấn đề**

### **Lý do chọn đề tài**

Hiện nay có rất nhiều trang web về game online trên internet, chúng mang đến cho người dùng sự giải trí nhanh chóng, giết thời gian vào những lúc rảnh rỗi. Vì vậy chúng em muốn tích hợp lập trình giữa web và game để tạo ra một sản phẩm giải trí đơn giản cho người dùng bằng những kiến thức đã học trong những năm qua, kết hợp thêm những kiến thức mới mà bọn em đã tự tìm hiểu.

### **Mục tiêu đề tài**

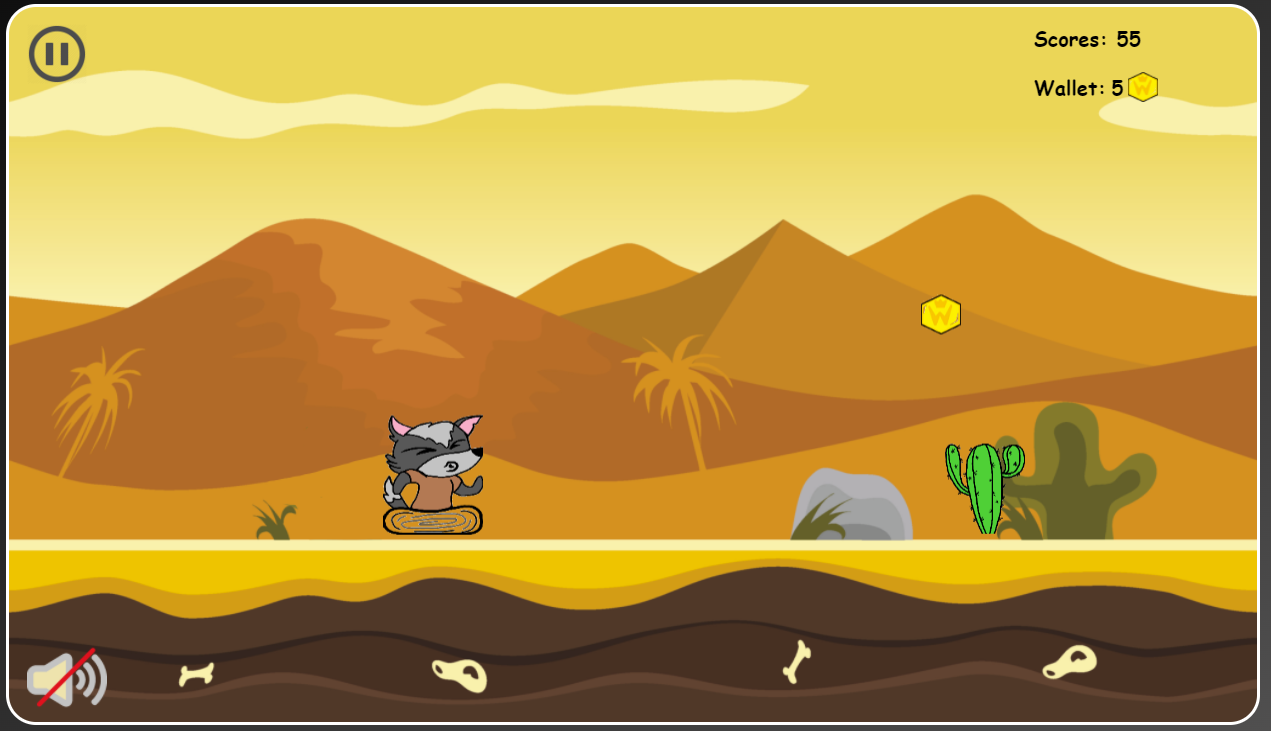
Thông qua việc thực hiện đồ án môn học này bọn em sẽ được học hỏi thêm nhiều kiến thức mới đồng thời cũng ôn lại và áp dụng thực tế những kiến thức đã học trước đó để tạo ra được sản phẩm hoàn chỉnh đáp ứng được yêu cầu của môn học này. Không chỉ vậy bọn em cũng cải thiện được khả năng làm việc nhóm, tự học và tìm kiếm tài liệu sao cho hiệu quả.

# **PHẦN 2. NỘI DUNG**

## **Giới thiệu**

Chúng em lập trình web game dựa trên các ngôn ngữ HTML, CSS, Javascript, NodeJS để tạo ra được giao diện hoàn chỉnh có kết hợp với dữ liệu người dùng.

Game bọn em tạo ra dựa trên đồ họa 2D, thể loại vượt chướng ngại vật có tính điểm và tiền. Cộng thêm các tính năng nổi bật như đăng ký, đăng nhập, hệ thống cửa hàng để người chơi có thể mua được những trang phục để tăng động lực cho người chơi không bị nhàm chán.



## **Kiến thức áp dụng**

Nhóm em đã áp dụng các yêu cầu về kiến thức mà thầy đã giao trong đồ án môn học này bao gồm các môn học mà chúng em đã học ở các học kỳ trước như: Lập trình web, hệ quản trị cơ sở dữ liệu, cấu trúc dữ liệu và giải thuật, kỹ thuật lập trình, lập trình hướng đối tượng.

Công cụ lập trình:

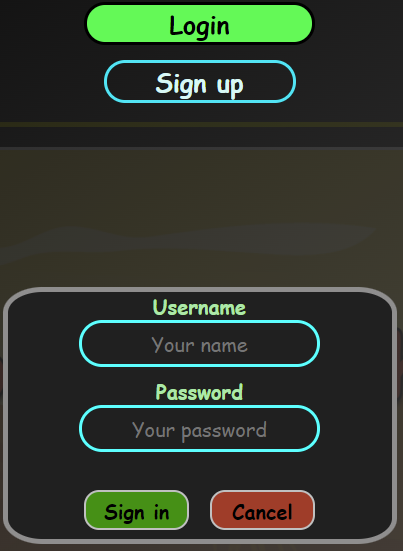
* Frontend: EJS, CSS, Javascript.
* Backend: NodeJS.
* Database: MySQL.
* VSCode

## **Giao diện**

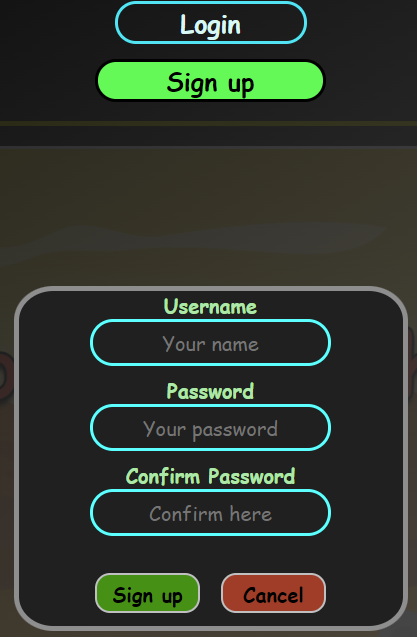
### **Header**

Phần đầu của giao diện bao gồm các chức năng như đăng ký, đăng nhập.

Đăng nhập:



Đăng ký:



Sau khi ta đăng nhập thành công:



### **Body**

Phần thân của trang web chúng em dùng để làm khu vực dành cho người chơi, mọi hoạt động ở đây bao gồm play, store.

Đây là giao diện của phần body trước khi user đăng nhập để chơi game.



Còn đây là giao diện sau khi đăng nhập.

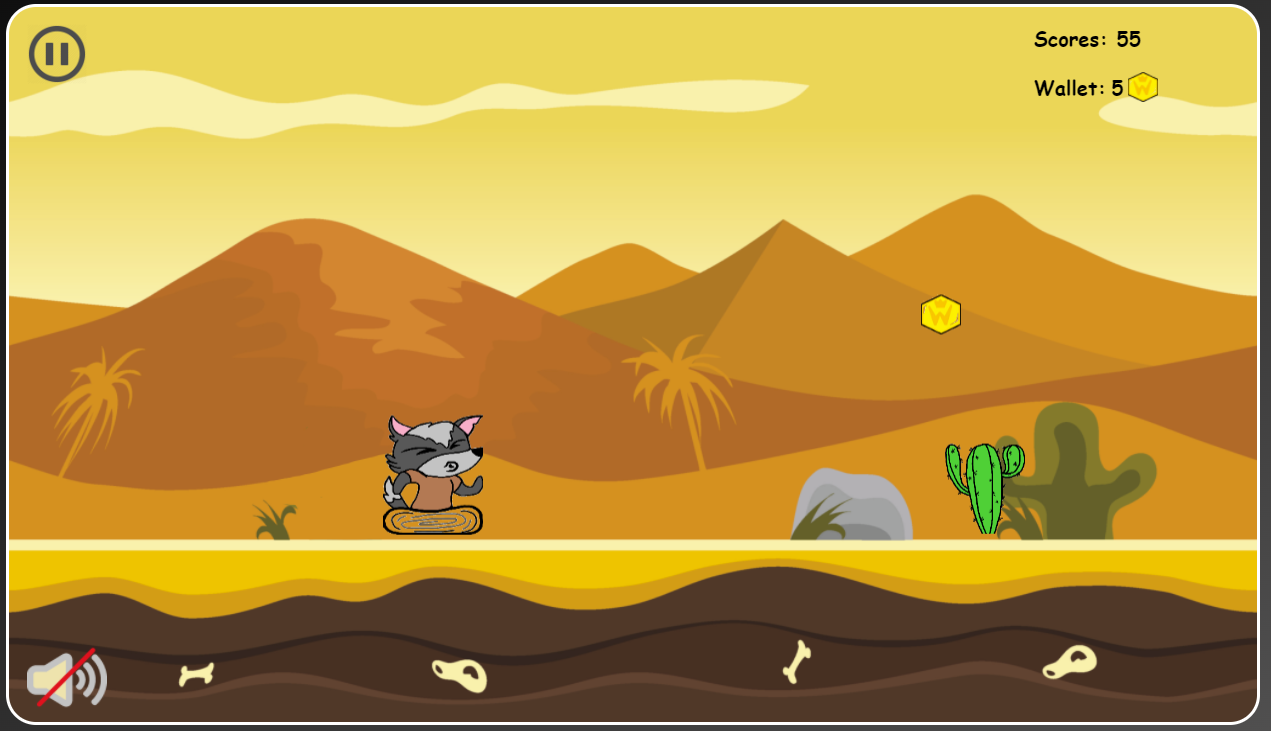
Ở phần này chúng em có tích hợp hai tính năng chính của trò chơi đó là chức năng play và cửa hàng.



Nhóm em cũng đã tích hợp thêm tính năng tắt/mở nhạc cho người chơi.



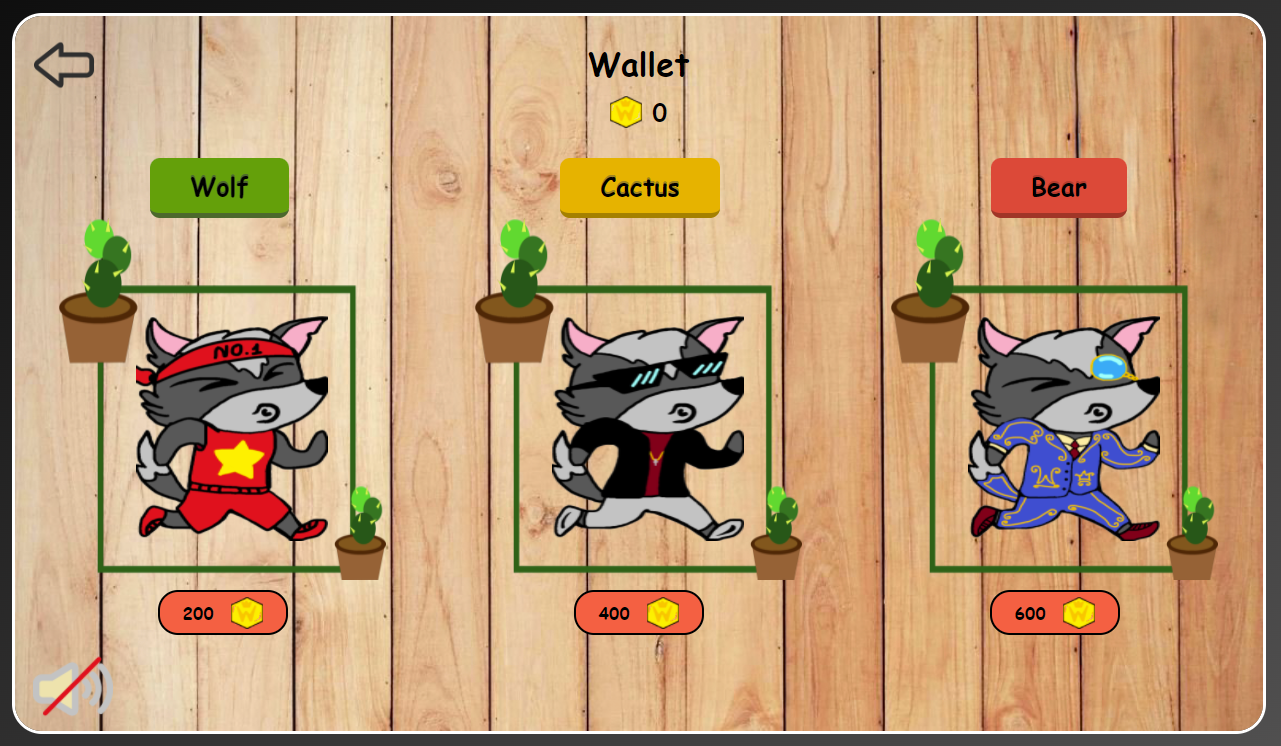
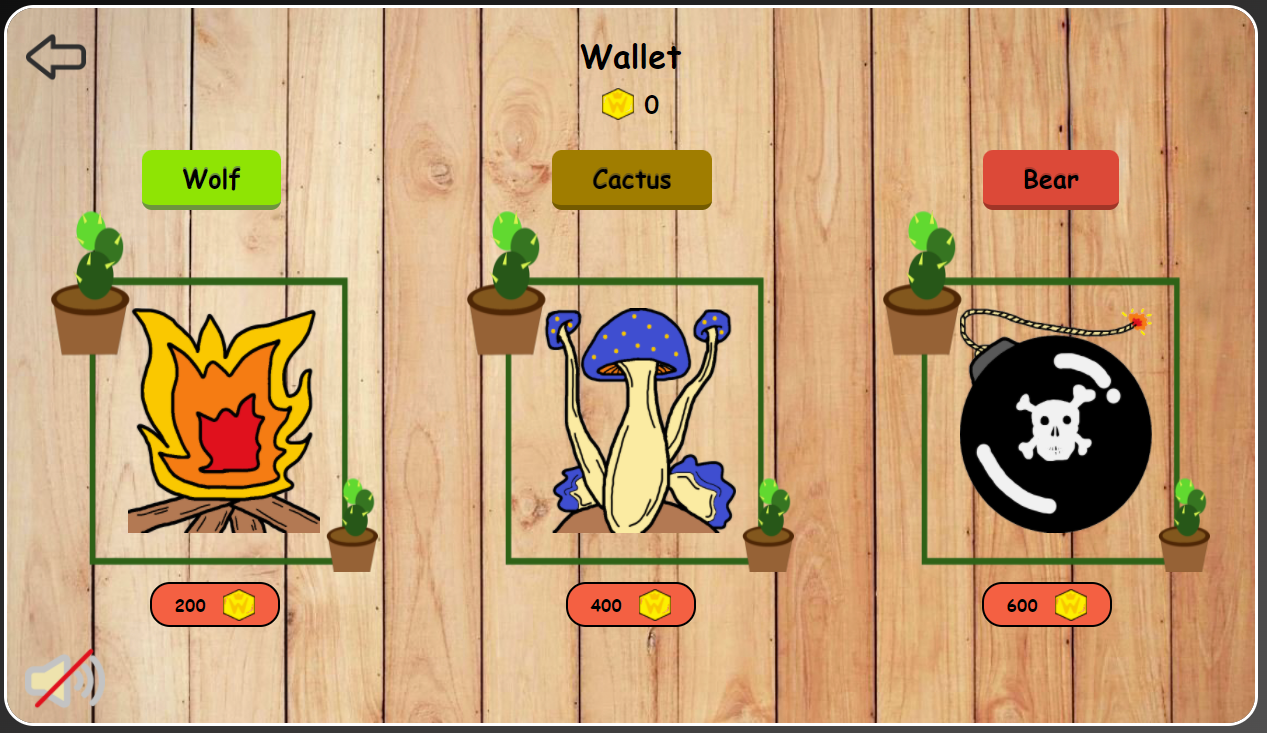
Tính năng play thì cơ bản như sau: Ngưởi chơi sẽ ấn nút W/ Space/ Arrowup để nhảy và né những chướng ngại vật, nếu va chạm thì sẽ có thông báo thua và phải chơi lại từ đầu hoặc thoát trò chơi nếu người chơi không muốn chơi nữa.





Game chúng em cũng có các tính năng ghi nhận như điểm số và tiền dùng để mua trang phục mới trong cửa hàng. Với mỗi một đồng vàng được 5 xu, đồng xanh được 10 xu.



Tiếp theo là hệ thống cửa hàng để người chơi có thể mua trang phục. Có ba loại trang phục mà người chơi có thể mua đó là Wolf ( nhân vật người chơi ) và Cactus, Bear ( chướng ngại vật ).  



### **Footer**

Cuối cùng là phần mô tả trò chơi bao gồm cách chơi và thông tin liên hệ với đội ngũ phát triển.



## **Mã nguồn**

### **Client**

Phần client tụi em chia làm 7 files:

* **main.ejs** chứa các thẻ html để hiển thị giao diện web

|  |  |  |
| --- | --- | --- |
| **STT** | **Giải thích** | **Code** |
| 1 | Phần header chứa thanh navigation, form login, form thông tin người dùng sau khi đăng nhập | <header>        <input class="noDisplay" type="checkbox" id="loginBtn">        <input class="noDisplay" type="checkbox" id="signUpBtn">        <div id="navigation">           <button type="button" class="btnLog-Sign" onclick="loginOption('', '')">Login</button>           <button type="button" class="btnLog-Sign" onclick="signUpOption('', '')">Sign up</button>        </div>        <div id="playerInfo">           <span class="welcome">Welcome</span>           <div class="verticalLine"></div>           <div>              <a href="/?action=logout" class="logout-changePass">Logout</a>           </div>        </div>        <form action="/" id="loginForm" method="post">           <label id="lblUser" for="userInput">Username<br><input class="input" type="text" id="userInput" name="username"                 placeholder="Your name" /></label></br>           <label id="lblPass" for="passInput">Password<br><input class="input" type="password" id="passInput" name="password"                 placeholder="Your password" /></label></br>           <input class="noDisplay" type="checkbox" id="chbxConfPass" />           <label id="lblConfPass" for="confPassInput">Confirm Password<br><input class="input" type="password" name="confirm"                 id="confPassInput" placeholder="Confirm here" /></label>           <span class="alert" id="alert"></span>           <br>           <button type="button" id="signInBtn" onclick="login()">Sign in</button>           <button type="button" id="cancelBtn" onclick="cancelLogin()">Cancel</button>        </form>     </header> |
| 2 | Phần main chứa giao diện trước khi đăng nhập, giao diện cửa hàng và màn chơi | <main id="main">        <hr id="headline" style="size:3px" />        <div id="Forms">           <input class="noDisplay" type="checkbox" id="storeBtn">           <div id="menu">              <h1 id="gameTitle">Wolf Marathon</h1>              <h3>You need to login to play!</h3>              <div style="display: none;">                 <button type="button" onclick="startgame()">PLAY</button>                 <button type="button" onclick="storeBtn()">STORE</button>              </div>           </div>           <div id="store">              <div id="walletSection">                 <p>Wallet</p>                 <div>                    <img src="images/yellowCoin.png" alt="your wallet">                    <span id="wallet">0</span>                 </div>              </div>              <div id="navigationStore">                 <a href="#" class="action-button green active" id="btnWolf" onclick="selectSkin(0)">Wolf</a>                 <a href="#" class="action-button yellow" id="btnCactus" onclick="selectSkin(1)">Cactus</a>                 <a href="#" class="action-button red" id="btnBear" onclick="selectSkin(2)">Bear</a>              </div>              <div id="skins-section">                 <div class="outer-frame">                    <img src="images/wolf\_jump1.png" class="inner-frame" alt="skin #1">                 </div>                 <div class="outer-frame">                    <img src="images/wolf\_jump2.png" class="inner-frame" alt="skin #2">                 </div>                 <div class="outer-frame">                    <img src="images/wolf\_jump3.png" class="inner-frame" alt="skin #3">                 </div>              </div>              <div id="buy-section">                 <button type="button" class="buyBtn" onclick="playerBuy(0)">                    <span name="textBuyBtn">200</span><img src="images/yellowCoin.png" alt="coin" class="buyIcon">                 </button>                 <button type="button" class="buyBtn" onclick="playerBuy(1)">                    <span name="textBuyBtn">400</span><img src="images/yellowCoin.png" alt="coin" class="buyIcon">                 </button>                 <button type="button" class="buyBtn" onclick="playerBuy(2)">                    <span name="textBuyBtn">600</span><img src="images/yellowCoin.png" alt="coin" class="buyIcon">                 </button>              </div>              <button type="button" class="buyBtn" id="defaultSkinBtn" onclick="defaultSkin()">Default skin</button>              <button type="button" id="backStore" onclick="backStore()"></button>           </div>        </div>        <div id="playZone">           <canvas id="playCanvas"></canvas>           <div id="pauseBtn" class="pauseBtn" onclick="pause\_resumeGame()"></div>           <div id="scoreAndWallet">              <p>High Scores: <span id="highscore">0</span></p>              <p>Scores: <span id="score">0</span></p>              <div style="display: flex; flex-direction: row; align-items: center; justify-content: space-between;">                 Wallet: <span id="playWallet" style="margin: 0 5px;">0</span><img src="images/yellowCoin.png" alt="coin" id="divForWallet">              </div>           </div>           <div id="pauseForm">              <b style="-webkit-text-stroke: 3px red; color: white">PAUSED</b>           </div>           <div id="restartForm">              <h1>Game Over</h1>              <p style="font-size: 18px">Scores: <span id="scoreGameOver">0</span> </p>              <button type="button" id="restartBtn" onclick="init()">Restart</button>              <button type="button" id="quitBtn" onclick="quitGame()">Quit</button>           </div>        </div>        <button type="button" id="soundBtn" onclick="toggleMusic()"></button>        <hr id="footline" style="size:3px" />     </main> |
| 3 | Footer chứa thông tin web, hướng dẫn chơi | <footer>        <h3><span style="color:cyan">HOW TO PLAY?</span></h3>        <h3>Use <span class="controlSpanText">AD</span> or <span class="controlSpanText">Left/Right</span> key to move along. <br> Use <span class="controlSpanText">W/Spacebar/Up</span> key to jump. <br> Avoid obstacles and collect coins for fabulous skins in store.</h3>        <h4>           <span style="color:tomato">Any bug found please contact</span>           <a style="color:greenyellow;" href = "mailto:hung9cong@gmail.com?subject = Feedback&body = Message">here</a>        </h4>     </footer> |
| 4 | Phần audio để thêm âm thanh vào web | <audio src="sounds/Guitar.mp3" id="music" loop></audio>     <audio src="sounds/jump.wav" id="soundJump"></audio>     <audio src="sounds/lose.wav" id="soundLose"></audio>     <audio src="sounds/pickupCoin.wav" id="soundCoin"></audio>     <audio src="sounds/select.wav" id="soundSelect"></audio> |
| 5 | Phần script thêm các file js để xử lí ui, ux | <script src="js/wolf.js"></script>  <script src="js/obstacles.js"></script>  <script src="js/coins.js"></script>  <script src="js/variables.js"></script>  <script src="js/display.js"></script>  <script src="js/game.js"></script> |
| 6 | Phần script xử lí kết quả response từ server | <script>     let data = '<%=data%>';     if(data == 'wrongAcc'){        loginOption("Your username or password is incorrect!", "red");     }     else if(data == 'succeed'){        loginOption("Register succeed!", "green");     }     else if(data == 'failed'){        signUpOption("User existed!", "red");     }     else if(data == 'logout'){        logout();     }     else if(data != ''){        let userdata = {              id: '<%=data.id%>',              username: '<%=data.username%>',              highscore: '<%=data.highscore%>',              money: '<%=data.money%>',              wolf: '<%=data.wolf%>',              cactus: '<%=data.cactus%>',              bear: '<%=data.bear%>',        };        loginSuccess(userdata);     }  </script> |

* **variables.js** dùng để định nghĩa các biến sẽ xài nhiều lần trong chương trình

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên hàm** | **Code** |
| 0 | Khai báo biến | var canvas = document.getElementById("playCanvas");  var context = canvas.getContext("2d");  var menu = document.getElementById("menu");  var info = menu.getBoundingClientRect();  var buyBtn = document.getElementsByClassName("buyBtn");  var textBuyBtn = document.getElementsByName("textBuyBtn");  var buyIcon = document.getElementsByClassName("buyIcon");  var storeWallet = document.getElementById("wallet");  var playWallet = document.getElementById("playWallet");  var myScores = document.getElementById("score");  var music = document.getElementById("music");  var soundJump = document.getElementById("soundJump");  var soundCoin = document.getElementById("soundCoin");  var soundLose = document.getElementById("soundLose");  var soundSelect = document.getElementById("soundSelect");  canvas.width = innerWidth \* 0.65;  canvas.height = innerHeight \* 0.73;  music.volume = 0.2;  soundCoin.volume = 0.1;  soundJump.volume = 0.1;  soundLose.volume = 0.5;  soundSelect.volume = 0.1;  var player = new wolf(canvas.width, canvas.height); //(info.height \* 0.570);  var obs = new obsticals(canvas.width, canvas.height); //(info.width, info.height \* 0.611);  var coin = new coins(canvas.width, player.preY - player.jumpDistance);  var score = 0;  var highscore = 0;  var wallet = 0;  var difficulty = 1;  var playerStatus = 0; //0-run 1-jump 2-die  var skinPrice = [200, 400, 600];  var wolfSkin = [0, 0, 0]; // 0-for sell 1-equip 2-equipped  var cactusSkin = [0, 0, 0];  var bearSkin = [0, 0, 0];  var skinTarget = "wolf";  var currentObs = null;  var currentCoin = null;  var playerMoveDirection = 0; //0 - not moving   2 - move right   -4 - move left  var obsSpawned = false;  var coinSpawned = false;  var isPause = false;  var isMusic = false;  var isLose = true;  var userName = "";  var userID = null; |

* **display.js** dùng để xử lí các thao tác hiển thị, mở đóng form

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên và vị trí hàm** | **Code** |
| 0 | toggleMusic  (4) | //bật tắt nhạc của game  function toggleMusic() {      if (!isMusic) {          music.play();          isMusic = true;          document.getElementById("soundBtn").style.backgroundImage = "url('../images/sound\_on.png')"      } else if (isMusic) {          music.pause();          isMusic = false;          document.getElementById("soundBtn").style.backgroundImage = "url('../images/sound\_off.png')"      }  } |
| 1 | loginOption  (17) | //Hiển thị form login  function loginOption(message, color) {      document.getElementById("passInput").value = "";      document.getElementById("confPassInput").value = "";      let login = document.getElementById("loginBtn");      if (login.checked) {          document.getElementById("main").style.opacity = 1;          login.checked = false;      } else {          document.getElementById("main").style.opacity = 0.1;          document.getElementById("signUpBtn").checked = false;          document.getElementById("chbxConfPass").checked = false;          document.getElementById("signInBtn").innerHTML = "Sign in";          document.getElementById("loginForm").action = '/?action=signin'          document.getElementById("alert").style.color = color;          document.getElementById("alert").innerHTML = message;          login.checked = true;      }  } |
| 2 | signUpOption  (37) | //Hiển thị form sign up  function signUpOption(message, color) {      document.getElementById("passInput").value = "";      let signup = document.getElementById("signUpBtn");      if (signup.checked) {          document.getElementById("main").style.opacity = 1;          signup.checked = false;      } else {          document.getElementById("main").style.opacity = 0.1;          document.getElementById("chbxConfPass").checked = true;          document.getElementById("loginBtn").checked = false;          document.getElementById("signInBtn").innerHTML = "Sign up";          document.getElementById("loginForm").action = '/?action=signup'          document.getElementById("alert").style.color = color;          document.getElementById("alert").innerHTML = message;          signup.checked = true;      }  } |
| 3 | validateInput  (56) | //Kiểm tra dữ liệu input của người dùng  function validateInput(login, username, password, confPass) {      let alertText = document.getElementById("alert");      if (username == "") {          alertText.innerHTML = "Please enter your username";          return false;      }      else if (password == "") {          alertText.innerHTML = "Please enter your password";          return false;      }      else if (!login.checked) {          if (confPass == "") {              alertText.innerHTML = "Please confirm your password";              return false;          }          else if (password != confPass) {              alertText.innerHTML = "Confirm password does not match";              return false;          }      }      return true;  } |
| 4 | login  (80) | //thực hiện gửi request signin tới server  function login() {      let login = document.getElementById("loginBtn");      let username = document.getElementById("userInput").value;      let password = document.getElementById("passInput").value;      let confPass = document.getElementById("confPassInput").value;      if (validateInput(login, username, password, confPass)) {          document.getElementById("loginForm").submit();      }  } |
| 5 | cancelLogin  (91) | //tắt form signin và signup  function cancelLogin() {      document.getElementById("loginBtn").checked = false;      document.getElementById("signUpBtn").checked = false;      document.getElementById("main").style.opacity = 1;  } |
| 6 | loadData  (98) | //gán dữ liệu người dùng vào các biến để xử lí khi chơi  function loadData(userdata) {      userID = Number(userdata.id);      userName = userdata.username;      highscore = Number(userdata.highscore);      wallet = Number(userdata.money);      for (let i = 0; i < 3; i++) {          wolfSkin[i] = Number(userdata.wolf[i]);          cactusSkin[i] = Number(userdata.cactus[i]);          bearSkin[i] = Number(userdata.bear[i]);      }  } |
| 7 | setCookies  (111) | //cấu hình cookies sau khi người dùng login thành công  function setCookies() {      document.cookie = `id=${userID}; expires=Thu, 18 Dec 2022 12:00:00 UTC`;      document.cookie = `highscore=${Math.round(highscore)}; expires=Thu, 18 Dec 2022 12:00:00 UTC`;      document.cookie = `username=${userName}; expires=Thu, 18 Dec 2022 12:00:00 UTC`;      document.cookie = `money=${wallet}; expires=Thu, 18 Dec 2022 12:00:00 UTC`;      document.cookie = `wolf=${wolfSkin.toString().replace(/,/g, "")}; expires=Thu, 18 Dec 2022 12:00:00 UTC`;      document.cookie = `bear=${bearSkin.toString().replace(/,/g, "")}; expires=Thu, 18 Dec 2022 12:00:00 UTC`;      document.cookie = `cactus=${cactusSkin.toString().replace(/,/g, "")}; expires=Thu, 18 Dec 2022 12:00:00 UTC`;  } |
| 8 | loginSuccess  (122) | //hiện giao diện sau khi người dùng login thành công  function loginSuccess(userdata) {      loadData(userdata);      setCookies();      document.querySelector("#menu div").style.display = "block";      document.querySelector("#menu h3").style.display = "none";      document.getElementById("playerInfo").style.display = "flex";      document.getElementById("navigation").style.display = "none";      document.querySelector(".welcome").innerHTML = "Welcome, " + userdata.username;  } |
| 9 | logout  (133) | //xóa cookies của người dùng khi logout  function logout() {      document.cookie = `id= ; expires=Thu, 18 Dec 1999 12:00:00 UTC`;      document.cookie = `highscore= ; expires=Thu, 18 Dec 1999 12:00:00 UTC`;      document.cookie = `username= ; expires=Thu, 18 Dec 1999 12:00:00 UTC`;      document.cookie = `money= ; expires=Thu, 18 Dec 1999 12:00:00 UTC`;      document.cookie = `wolf= ; expires=Thu, 18 Dec 1999 12:00:00 UTC`;      document.cookie = `bear= ; expires=Thu, 18 Dec 1999 12:00:00 UTC`;      document.cookie = `cactus= ; expires=Thu, 18 Dec 1999 12:00:00 UTC`;  } |
| 10 | storeBtn  (144) | //hiển thị giao diện cửa hàng  function storeBtn() {      document.getElementById("storeBtn").checked = true;      storeWallet.textContent = wallet.toString();      selectSkin(0);  } |
| 11 | backStore  (151) | //quay lại menu chính  function backStore() {      document.getElementById("storeBtn").checked = false;  } |
| 12 | playerBuy  (156) | //xử lí trang phục người dùng mua dựa theo tham số slot  function playerBuy(slot) {      if (skinTarget == "wolf") {          buy\_equipSkin(wolfSkin, slot);          update\_BuyBtn(wolfSkin);      }      else if (skinTarget == "cactus") {          buy\_equipSkin(cactusSkin, slot);          update\_BuyBtn(cactusSkin);      }      else {          buy\_equipSkin(bearSkin, slot);          update\_BuyBtn(bearSkin);      }      setCookies();  } |
| 13 | buy\_equipSkin  (173) | //xử lí người dùng mua hoặc trang bị trang phục  function buy\_equipSkin(skin, slot) {      if (skin[slot] == 0) { //buy          skin[slot] = 1;          bill(slot);      }      else { //Dùng để chuyển Equip -> Equipped          for (let i = 0; i < skin.length; i++) {              if (skin[i] == 2) {                  skin[i] = 1;              }          }          skin[slot] = 2;          updateSkin();      }  } |
| 14 | updateSkin  (190) | //load trang phục vào cho các nhân vật và vật cản tương ứng  function updateSkin() {      for (let i = 0; i < wolfSkin.length; i++) {          if (wolfSkin[i] == 2) {              player.loadImg(i + 1);              break;          }          player.loadImg(0);      }      for (let i = 0; i < bearSkin.length; i++) {          if (bearSkin[i] == 2) {              obs.loadImageBear(i + 1);              break;          }          obs.loadImageBear(0);      }      for (let i = 0; i < cactusSkin.length; i++) {          if (cactusSkin[i] == 2) {              obs.loadImageCactus(i + 1);              break;          }          obs.loadImageCactus(0);      }  } |
| 15 | bill  (217) | //trừ tiền khi người dùng mua thành công  function bill(slot) {      wallet -= skinPrice[slot];      storeWallet.textContent = wallet.toString();  } |
| 16 | selectSkin  (223) | //xử lí hiển thị trang phục theo tham số option khi người dùng lựa chọn loại trang phục  function selectSkin(option) {      let btnWolf = document.getElementById("btnWolf");      let btnCactus = document.getElementById("btnCactus");      let btnBear = document.getElementById("btnBear");      let slots = document.getElementsByClassName("inner-frame");      if (option == 0) { //wolf          btnWolf.classList.add("active");          btnCactus.classList.remove("active");          btnBear.classList.remove("active");          slots[0].src = "../images/wolf\_jump1.png";          slots[1].src = "../images/wolf\_jump2.png";          slots[2].src = "../images/wolf\_jump3.png";          update\_BuyBtn(wolfSkin);          skinTarget = "wolf";      }      else if (option == 1) { //cactus          btnWolf.classList.remove("active");          btnCactus.classList.add("active");          btnBear.classList.remove("active");          slots[0].src = "../images/cactus1.png";          slots[1].src = "../images/cactus2.png";          slots[2].src = "../images/cactus3.png";          update\_BuyBtn(cactusSkin);          skinTarget = "cactus";      }      else if (option == 2) { //bear          btnWolf.classList.remove("active");          btnCactus.classList.remove("active");          btnBear.classList.add("active");          slots[0].src = "../images/bear1.png";          slots[1].src = "../images/bear2.png";          slots[2].src = "../images/bear3.png";          update\_BuyBtn(bearSkin);          skinTarget = "bear";      }  } |
| 17 | update\_BuyBtn  (261) | //cập nhật trạng thái của các nút mua trang phục  function update\_BuyBtn(skin) {      for (let i = 0; i < skin.length; i++) {          if (skin[i] == 0) {              buyBtn[i].classList.remove("equippedBtn");              buyBtn[i].classList.remove("equipBtn");              buyIcon[i].style.display = "block";              textBuyBtn[i].innerText = skinPrice[i].toString();              if (wallet < skinPrice[i]) {                  buyBtn[i].classList.add("poorBtn");              }              else {                  buyBtn[i].classList.remove("poorBtn");              }          }          else {              buyIcon[i].style.display = "none";              if (skin[i] == 1) {                  buyBtn[i].classList.add("equipBtn");                  buyBtn[i].classList.remove("equippedBtn");                  textBuyBtn[i].innerText = "EQUIP";              }              else {                  buyBtn[i].classList.add("equippedBtn");                  buyBtn[i].classList.remove("equipBtn");                  textBuyBtn[i].innerText = "EQUIPPED";              }          }      }  } |
| 18 | defaultSkin  (292) | //cập nhật skin mặc định cho nhân vật, vật cản  function defaultSkin() {      if (skinTarget == 'wolf') {          for (let i = 0; i < wolfSkin.length; i++) {              if(wolfSkin[i] == 2){                  wolfSkin[i] = 1;                  break;              }          }          updateSkin();          update\_BuyBtn(wolfSkin);      }      else if (skinTarget == 'bear') {          for (let i = 0; i < bearSkin.length; i++) {              if(bearSkin[i] == 2){                  bearSkin[i] = 1;                  break;              }          }          updateSkin();          update\_BuyBtn(bearSkin);      }      else if (skinTarget == 'cactus') {          for (let i = 0; i < cactusSkin.length; i++) {              if(cactusSkin[i] == 2){                  cactusSkin[i] = 1;                  break;              }          }          updateSkin();          update\_BuyBtn(cactusSkin);      }      setCookies();  } |

* **game.js** dùng để xử lí cơ chế game, xử lí nút nhấn từ người dùng

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên và vị trí hàm** | **Code** |
| 0 | userInteraction  (4) | //xử lí dữ diệu từ nút nhấn bàn phím của người dùng  function userInteraction() {      window.addEventListener("resize", () => {          canvas.width = innerWidth \* 0.65;          canvas.height = innerHeight \* 0.73;      });      document.addEventListener("click", () => {          soundSelect.play();      })      document.addEventListener("keyup", (event) => {          if (!isPause && !isLose) {              if (event.key == "ArrowLeft" || event.key == "ArrowRight" || event.key == "d" || event.key == "a") {                  playerMoveDirection = 0;              }          }      });      document.addEventListener("keydown", (event) => {          if (!isPause && !isLose) {              event.preventDefault();              if (event.key == " " || event.key == "ArrowUp" || event.key == "w") {                  soundJump.play();                  player.isJump = true;                  if (player.y < player.preY && player.y > (player.preY - player.jumpDistance / 1.5) /\*&& player.isFall\*/) {                      player.jumpDelay = true;                  }              }              else if (event.key == "ArrowLeft" || event.key == "a") {                  playerMoveDirection = player.speedBackward;              }              else if (event.key == "ArrowRight" || event.key == "d") {                  playerMoveDirection = player.speedForward;              }          }      });  } |
| 1 | startgame  (40) | //hiển thị giao diện màn chơi  function startgame() {      document.getElementById("playCanvas").style.display = "block";      document.getElementById("pauseBtn").style.display = "block";      document.getElementById("Forms").style.display = "none";      document.getElementById("scoreAndWallet").style.display = "block";      init();  } |
| 2 | init  (49) | function init() {      player = new wolf(canvas.width, canvas.height);      obs = new obsticals(canvas.width, canvas.height);      coin = new coins(canvas.width, player.preY - player.jumpDistance);      score = 0;      difficulty = 1;      background1X = 0;      background2X = canvas.width;      playerMoveDirection = 0;      isPause = false;      player.isJump = false;      isLose = false;      obsSpawned = false;      coinSpawned = false;      document.getElementById("highscore").textContent = Math.round(highscore).toString();      document.getElementById("restartForm").style.display = "none";      updateSkin();      startBgAnimation();      loop();  } |
| 3 | loop  (74) | //vòng lặp chính của trò chơi  function loop() {      if (!isLose) {          if (!isPause) {              context.clearRect(0, 0, canvas.width, canvas.height);              playerJump();              playerMove();              spawnObstacle();              spawnCoin();              checkHitCoin();              checkLose();              drawCoin();              drawObstacle();              drawPlayer();              gainScores();              increaseDifficulty();          }          setTimeout(() => loop(), 10 - difficulty);      }      else {          gameOver();      }  } |
| 4 | inscreaseDifficulty  (99) | //tăng độ khó dựa theo điểm của người dùng  function increaseDifficulty() {      if (score % 700 == 0 && difficulty < 9) {          difficulty++;      }  } |
| 5 | gameOver  (106) | //hiển thị giao diện thua cuộc  function gameOver() {      document.getElementById("restartForm").style.display = "block";      document.getElementById("scoreGameOver").textContent = Math.round(score).toString();      if(score > highscore){          highscore = score;      }      stopBgAnimation();      setCookies();      soundLose.play();  } |
| 6 | checkHitCoin  (118) | //kiểm tra và xử lí người dùng chạm tiền thưởng chưa  function checkHitCoin() {      if (coin.x <= player.x + player.width && coin.x + coin.resolution >= player.x) { //conflict x-axis          if (player.y <= coin.y + coin.resolution) { //conflict y-axis              coinSpawned = false;              wallet += coin.value;              soundCoin.play();          }      }      playWallet.textContent = wallet.toString();  } |
| 7 | checkLose  (130) | //kiểm tra người dùng chạm vật cản chưa  function checkLose() {      if (obs.x <= player.x + player.width && obs.x + obs.width >= player.x) { //conflict x          if (player.y + player.height >= obs.y) {              isLose = true;              playerStatus = 2;          }      }  } |
| 8 | gainScore  (140) | //tăng điểm cho người dùng theo mỗi vòng lặp  function gainScores() {      score += 0.1;      document.getElementById("score").textContent = Math.round(score).toString();  } |
| 9 | spawnCoin  (146) | //sinh ngẫu nhiên tiền thưởng trong màn chơi  function spawnCoin() {      if (!coinSpawned) {          let type = Math.round(Math.random() \* (1000));          coin.x = canvas.width;          if (type <= 10) {              coinSpawned = true;              if (type <= 8) {                  currentCoin = coin.imgYellowCoin;                  coin.y = coin.yYellowCoin;                  coin.resolution = coin.rYellowCoin;                  coin.value = coin.vYellowCoin;              } else {                  currentCoin = coin.imgBlueCoin;                  coin.y = coin.yBlueCoin;                  coin.resolution = coin.rBlueCoin;                  coin.value = coin.vBlueCoin;              }          }      }  } |
| 10 | spawnObstacle  (168) | //sinh ngẫu nhiên vật cản trong màn chơi  function spawnObstacle() {      if (!obsSpawned) {          let type = Math.round(Math.random() \* (500 + difficulty));          if (type <= 6 + difficulty) {              obs.x = canvas.width;              obsSpawned = true;              if (type > 5) {                  currentObs = obs.imgBear;                  obs.y = obs.yBear;              }              else {                  currentObs = obs.imgCactus;                  obs.y = obs.yCactus;              }          }      }  } |
| 11 | playerMove  (187) | //xử lí khi người dùng qua trái qua phải  function playerMove() {      if(player.x <= 0 && playerMoveDirection == player.speedBackward)          playerMoveDirection = 0;      if (player.x + player.width >= canvas.width && playerMoveDirection == player.speedForward)          playerMoveDirection = 0;      player.move(playerMoveDirection);  } |
| 12 | playerJump  (196) | //xử lí khi người dùng nhảy  function playerJump() {      if (player.isJump) {          if (player.jump()) {              if (!player.jumpDelay)                  player.isJump = false;              else                  player.jumpDelay = false;          }          playerStatus = 1;      }      else {          playerStatus = 0;      }  } |
| 13 | drawCoin  (212) | //vẽ hình tiền thưởng lên màn chơi  function drawCoin() {      if (coinSpawned) {          if (coin.move()) { //coin go out of zone              coinSpawned = false;          }          context.drawImage(currentCoin, coin.x, coin.y, coin.resolution, coin.resolution);      }  } |
| 14 | drawObstacle  (222) | //vẽ hình vật cản lên màn chơi  function drawObstacle() {      if (obsSpawned) {          if (obs.move()) { //obs go out of zone              obsSpawned = false;          }          context.drawImage(currentObs, obs.x, obs.y, obs.width, obs.height);      }  } |
| 15 | drawPlayer  (232) | //vẽ nhân vật lên màn chơi  function drawPlayer() {      switch (playerStatus) {          case 0: context.drawImage(player.img\_run, player.x, player.y, player.width, player.height); break;          case 1: context.drawImage(player.img\_jump, player.x, player.y, player.width, player.height); break;          case 2: context.drawImage(player.img\_die, player.x, player.y, player.width, player.height); break;      }  } |
| 16 | pause\_resumeGame  (241) | //hiện hoặc tắt giao diện khi tạm ngừng trò chơi  function pause\_resumeGame() {      if (!isLose) {          if (!isPause) {              document.getElementById("pauseBtn").style.backgroundImage = "url('../images/resume\_btn.png')"              isPause = true;              document.getElementById("pauseForm").style.display = "block";              stopBgAnimation();          }          else {              document.getElementById("pauseBtn").style.backgroundImage = "url('../images/pause\_btn.png')"              isPause = false;              document.getElementById("pauseForm").style.display = "none";              startBgAnimation();          }      }  } |
| 17 | quitGame  (259) | //tắt giao diện màn chơi và hiển thị  function quitGame() {      document.getElementById("playCanvas").style.display = "none";      document.getElementById("pauseBtn").style.display = "none";      document.getElementById("Forms").style.display = "block";      document.getElementById("scoreAndWallet").style.display = "none";      document.getElementById("restartForm").style.display = "none";  } |
| 18 | stopBgAnimation  (268) | //dừng chuyển động hình nền  function stopBgAnimation() {      canvas.style.animation = '';  } |
| 19 | startBgAnimation  (274) | //bật chuyển động hình nền  function startBgAnimation() {      canvas.style.animation = 'backgroundAnimation 3.1s infinite linear';  } |

* **wolf.js** tạo lớp tượng trưng cho nhân vật là người chơi

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên và vị trí hàm** | **Code** |
| 0 | constructor  (3) | //khởi tạo cho đối tượng của class  constructor(x, y) {          this.x = x \* 0.3;          this.y = y \* 0.57;          this.width = 100;          this.height = 120;          this.baseG = -6;          this.speedG = 0.11;          this.g = this.baseG;          this.preY = this.y;          this.speedBackward = -4;          this.speedForward = 2;          this.jumpDistance = 170;          this.isJump = false;          this.isMove = false;          this.jumpDelay = false;          this.isFall = false;          this.img\_run = null;          this.img\_jump = null;          this.img\_die = null;      } |
| 1 | loadImg  (27) | //lựa chọn hình ảnh để load vào nhân vật dựa theo tham số slot      loadImg(slot) {          this.img\_run = new Image;          this.img\_jump = new Image;          this.img\_die = new Image;          if (slot == 0) {              this.img\_run.src = "../images/wolf\_run.png"              this.img\_jump.src = "../images/wolf\_jump.png"              this.img\_die.src = "../images/wolf\_die.png"          }          else if (slot == 1) {              this.img\_run.src = "../images/wolf\_run1.png"              this.img\_jump.src = "../images/wolf\_jump1.png"              this.img\_die.src = "../images/wolf\_die1.png"          }          else if (slot == 2) {              this.img\_run.src = "../images/wolf\_run2.png"              this.img\_jump.src = "../images/wolf\_jump2.png"              this.img\_die.src = "../images/wolf\_die2.png"          }          else {              this.img\_run.src = "images/wolf\_run3.png"              this.img\_jump.src = "images/wolf\_jump3.png"              this.img\_die.src = "images/wolf\_die3.png"          }      } |
| 2 | jump  (53) | //xử lí khi nhân vật nhảy      jump() {          this.y += this.g;          this.g += this.speedG;          if(this.y > this.preY){              this.y = this.preY;              this.g = this.baseG;              return true;          }          return false;      } |
| 3 | move  (64) | //xử lí khi nhân vật qua trái qua phải      move(direction) {          this.x += direction;      } |

* **obstacles.js** tạo lớp tượng trưng cho chướng ngại vật

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên và vị trí hàm** | **Code** |
| 0 | constructor  (3) | //khởi tạo đối tượng của class      constructor(x, y) {          this.yBear = y \* 0.611 - 30;          this.yCactus = y \* 0.611;          this.x = x;          this.y = this.yCactus;          this.width = 80;          this.height = 90;          this.imgCactus = null;          this.imgBear = null;      } |
| 1 | loadImageCactus  (16)  loadImageBear  (25) | //load hình ảnh cho chướng ngại vật      loadImageCactus(slot) {          this.imgCactus = new Image;          switch (slot) {              case 0: this.imgCactus.src = "../images/cactus.png"; break;              case 1: this.imgCactus.src = "../images/cactus1.png"; break;              case 2: this.imgCactus.src = "../images/cactus2.png"; break;              case 3: this.imgCactus.src = "../images/cactus3.png"; break;          }      }      loadImageBear(slot) {          this.imgBear = new Image;          switch (slot) {              case 0: this.imgBear.src = "../images/bear.png"; break;              case 1: this.imgBear.src = "../images/bear1.png"; break;              case 2: this.imgBear.src = "../images/bear2.png"; break;              case 3: this.imgBear.src = "../images/bear3.png"; break;          }      } |
| 2 | move  (35) | //xử lí khi chướng ngại vật di chuyển      move() {          this.x -= 4; //obs move          if (this.x < -this.width) {              return true;          }          return false;      } |

* **coins.js** tạo lớp tượng trưng cho tiền thưởng trong game

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên và vị trí hàm** | **Code** |
| 0 | constructor  (3) | //khởi tạo đối tượng của class      constructor(x, y) {          this.yYellowCoin = y + 50;          this.yBlueCoin = y;          this.rYellowCoin = 40;          this.rBlueCoin = 30;          this.vYellowCoin = 5;          this.vBlueCoin = 10;          this.x = x;          this.y = this.yYellowCoin;          this.resolution = this.rYellowCoin;          this.value = this.vYellowCoin;          this.imgYellowCoin = null;          this.imgBlueCoin = null;          this.loadImageYellowCoin();          this.loadImageBlueCoin();      } |
| 1 | loadImage  YellowCoin  (22)  loadImage  BlueCoin  (26) | //load hình ảnh cho tiền thưởng      loadImageYellowCoin() {          this.imgYellowCoin = new Image;          this.imgYellowCoin.src = "../images/yellowCoin.png";      }      loadImageBlueCoin() {          this.imgBlueCoin = new Image;          this.imgBlueCoin.src = "../images/blueCoin.png";      } |
| 2 | move  (31) | //xử lí khi tiền thưởng di chuyển      move() {          this.x -= 3;          if (this.x < -this.resolution) {              return true;          }          return false;      } |

### **Server**

|  |  |  |
| --- | --- | --- |
| STT | Tên file | Mã nguồn |
| 0 | server.js  (1) | //thêm thư viện cần thiết  const express = require('express');  const bcrypt = require('bcryptjs');  const db = require('./utils/db'); |
| 1 | server.js  (7) | //cấu hình app nodejs  const app = express();  app.use(express.urlencoded({      extended: true  }));  app.use(express.static('public'));  app.set('views', './views');  app.set('view engine', 'ejs'); |
| 2 | server.js  (18) | // Xử lý get request từ client tới server  app.get('/', function (req, res) {      let cookies = req.headers.cookie;      if (cookies !== undefined) {          if (checkCookiesValid(cookies)) {              if ('logout' === req.query.action) {                  console.log('logout');                  let user = convertCookiesToUser(cookies);                  db.updateUser(user, function(result){                      if(result){                          res.render('main', { data: 'logout' });                      }                      else{                          console.log('logout failed');                      }                  })              }              else {                  console.log('cookies valid');                  let user = convertCookiesToUser(cookies);                  res.render('main', { data: user });              }          }          else {              console.log('cookies not valid');              res.render('main', { data: 'logout' });          }      }      else {          console.log('in default');          res.render('main', { data: '' });      }  }); |
| 3 | server.js  (54) | // Xử lý post request từ client tới server  app.post('/', function (req, res) {      if ('signin' === req.query.action) {          console.log('in signin');          let username = req.body.username;          let password = req.body.password;          db.checkLogin(username, password,              function (result) {                  if (result != null) {                      res.render('main', { data: result });                  }                  else {                      res.render('main', { data: 'wrongAcc' });                  }              });      }      else if ('signup' === req.query.action) {          console.log('in signup');          let username = req.body.username;          let password = req.body.password;          let password\_hash = bcrypt.hashSync(password, 8);          db.addUser(username, password\_hash,              function (isAdded) {                  if (isAdded) {                      res.render('main', { data: 'succeed' });                  }                  else {                      res.render('main', { data: 'failed' });                  }              });      }  }); |
| 4 | server.js  (87) | // Kiểm tra các giá trị của Cookie  function checkCookiesValid(cookies) {      let count = 0;      if (getCookie('id', cookies) != '') {          count++;      }      if (getCookie('username', cookies) != '') {          count++;      }      if (getCookie('highscore', cookies) != '') {          count++;      }      if (getCookie('money', cookies) != '') {          count++;      }      if (getCookie('wolf', cookies) != '') {          count++;      }      if (getCookie('bear', cookies) != '') {          count++;      }      if (getCookie('cactus', cookies) != '') {          count++;      }      return count == 7;  } |
| 5 | server.js  (115) | // Chuyển định dạng cookie sang object  function convertCookiesToUser(cookies) {      let userdata = {          id: getCookie('id', cookies),          username: getCookie('username', cookies),          highscore: getCookie('highscore', cookies),          money: getCookie('money', cookies),          wolf: getCookie('wolf', cookies),          bear: getCookie('bear', cookies),          cactus: getCookie('cactus', cookies)      }      return userdata;  } |
| 6 | server.js  (129) | // Lấy giá trị Cookie trong chuỗi Cookies  function getCookie(cname, cookies) {      let name = cname + "=";      let ca = cookies.split(';');      for (let i = 0; i < ca.length; i++) {          let c = ca[i];          while (c.charAt(0) == ' ') {              c = c.substring(1);          }          if (c.indexOf(name) == 0) {              return c.substring(name.length, c.length);          }      }      return '';  } |
| 7 | server.js  (145) | // Đặt port để chạy server  const PORT = 3000;  app.listen(PORT, () => console.info(`listening on http://localhost:${PORT}/`)); |
| 8 | db.js  (1) | //thêm thư viện cần thiết  const mysql = require('mysql');  const bcrypt = require('bcryptjs');  const config = require('../config/default.json'); |
| 9 | db.js  (7) | //tạo pool  const pool = mysql.createPool(config.mysql); |
| 10 | db.js  (10) | //kiểm tra user có tồn tại chưa  function checkUser(para1, fn) {      let sql = 'select \* from user\_info where username = ?';      pool.query(sql, [para1], function (error, results) {          if (error) {              console.log(error.sqlMessage);              throw error;          }          else if (results.length > 0) {              return fn(false);          }          else {              return fn(true);          }      })  } |
| 11 | db.js  (26) | module.exports = {      //kiểm tra login thành công hay không      checkLogin: function (username, password, fn) {          let sql = 'select \* from user\_info, skins where user\_info.id = skins.user\_id and username = ?';          pool.query(sql, [username, password], function (error, results) {              if (error) {                  console.log(error.sqlMessage);                  throw error;              }              else if (results.length > 0) {                  if (bcrypt.compareSync(password, results[0].password)) {                      let userdata = {                          id: results[0].id,                          username: results[0].username,                          highscore: results[0].highscore,                          money: results[0].money,                          wolf: results[0].wolf,                          cactus: results[0].cactus,                          bear: results[0].bear,                      }                      return fn(userdata);                  }              }              return fn(null);          });      },      //thêm user mới vào database      addUser: function (username, password, fn) {          let sql = 'insert into user\_info set username = ?, password = ?, highscore = 0, money = 0';          checkUser(username, function (results) {              if (results) {                  pool.query(sql, [username, password], function (error) {                      if (error) {                          console.log(error.sqlMessage);                          throw error;                      }                      else                          return fn(true);                  })              }              else {                  return fn(false);              }          });      },      //cập nhật dữ liệu của user      updateUser(user, fn) {          let sql = 'update user\_info, skins set highscore = ?, money = ?, wolf=?, cactus=?,bear=? where id=user\_id and id=?'          pool.query(sql,[user.highscore, user.money, user.wolf, user.cactus, user.bear, user.id], function (error, result){              if(error){                  console.log(error.sqlMessage);                  throw error;              }              else if(result.affectedRows){                  fn(true);              }              else{                  fn(false);              }          })      }  }; |
| 12 | default.json  (cấu hình kết nối mysql) | {      "mysql": {          "host": "localhost",          "port": 3306,          "user": "root",          "password": "root123",          "database": "doan\_cntt",          "connectionLimit": 50      }  } |

### Database

Diagram

Description automatically generated

## **Khó khăn**

Trong quá trình làm việc, nhóm chúng em có một vài những khó khăn nhất định do học online nên không thể gặp và trao đổi trực tiếp, nhưng sau cùng nhóm cũng đã thống nhất được ý kiến chung khiến cho sự phối hợp dần trở nên thành công để hoàn thành đồ án môn học này. Bên cạnh đó cũng có một vài khó khăn nhỏ như do tự tìm tòi học hỏi từ nhiều nguồn nên mã nguồn không được gọn gàng và có một vài chỗ chưa được logic cho lắm.

# **PHẦN 3. KẾT LUẬN**

## **Kết quả đạt được**

Qua môn học này, nhóm chúng em đã ôn lại được những kiến thức đã học trong các năm qua, dần làm tiền đề để chúng em phát triển cho những năm sau này, mỗi thành viên đều cải thiện được khả năng làm việc nhóm sao cho hiệu quả.

Đồng thời chúng em cũng đã được học hỏi thêm được những kiến thức mới ngoài chương trình đào tạo.

## **Hướng phát triển**

Mục tiêu đặt ra sau khi hoàn thành đồ án môn học này, nhóm chúng em nhận thấy có thể phát triển thêm được nhiều tính năng như xếp hạng các người chơi, tích hợp được thêm nhiều tựa game trên một trang web để người dùng có thể lựa chọn chơi những game mình thích.

# **TÀI LIỆU THAM KHẢO**

Tài liệu HTML cơ bản: <https://www.w3schools.com/html/default.asp>

Tài liệu CSS cơ bản: <https://www.w3schools.com/css/default.asp>

Tài liệu Javascript cơ bản: <https://www.w3schools.com/js/default.asp>

Tài liệu về NodeJS: <https://www.w3schools.com/nodejs/default.asp>

Tài liệu về Cookie: https://www.w3schools.com/js/js\_cookies.asp