BOUNCE IT!

Prateek Choudavarpu	Shubham Pardeshi	Manikanta Kovvuri
Grad, Comp. Science	Grad, Comp. Science	Grad, Comp. Science
pchouda@clemson.edu	ssparde@clemson.edu	bkovvur@clemson.edu

- **Objective:**

  "BOUNCE IT!" is an engaging casual gaming experience available on both mobile and web platforms. The game presents players with the challenge of skillfully navigating a ball through a diverse array of obstacles and trials. Characterized by its straightforward yet exceptionally compelling gameplay, "BOUNCE IT!" caters to a broad audience, encompassing both novice and seasoned gamers. The core objective of the game entails skillfully guiding a bouncing ball through a series of intricate obstacles and challenging scenarios, ultimately aiming for a successful completion at a designated endpoint.

- **Rules to Play:**

  1. Upon launching the game, the initial interface is the Home page.
  2. The Home page features two distinct buttons: "Start Game" and "Instructions." Opting for "Start Game" will seamlessly transition you to the game interface, while "Instructions" provides indispensable insights into the game's regulations.
  3. The game comprises three progressively challenging levels that players are tasked with conquering.
  4. For maneuvering your character, the left and right arrow keys offer directional control, while the up arrow key enables character jumps.
  5. Exercise prudence when encountering Red tiles, as they possess the potential to prematurely terminate your game.
  6. Upon successfully reaching the endpoint, demarcated by a Yellow tile, you will advance to the subsequent level.
  7. Upon reaching the third level, the ultimate stage, successful completion will prompt a return to the Home page.
  8. Throughout gameplay, a prominently displayed timer and scoreboard facilitate the tracking of your performance.

- **Technology Stack:**

HTML/CSS are employed to establish the foundational structure and aesthetics of the game interface, while JavaScript is utilized to augment interactivity and feature-rich functionality.

- **Setup and Deployment:**

1. Clone the repository using the following command: `git clone https://github.com/Bounce-It/Bounce_It.git`.

2. Open the "index.html" file in Visual Studio.

3. Execute the file within the Visual Studio environment.

4. Launch a web browser and navigate to "http://127.0.0.1:5500/" to commence gameplay.

5. Additionally, you can access the game via the following link: (https://bounceit.netlify.app/#).

- **Credits for third-party assets and code:**

1. **jQuery**: jQuery is a high-performance, lightweight JavaScript library that offers a rich array of features. It simplifies tasks such as HTML document traversal, manipulation, event handling, animation, and Ajax operations, providing a user-friendly API that is compatible with a wide range of web browsers.

   Code Snippet: <script src="https://code.jquery.com/jquery-3.7.1.min.js"></script>

2. **Sweet Alert 2:** This API facilitates the creation of alert boxes tailored to specific requirements.

   Code Snippet 1: <script src="https://cdn.jsdelivr.net/npm/sweetalert2@11.4.8/dist/sweetalert2.all.min.js"> </script>

   Code Snippet 2: scripts:{

   next_level:

   "Swal.fire({icon: 'success', title: 'Successful', text: 'Redirecting to Level 2!', confirmButtonText: 'Next Level'}).then((result) => {if(result.isConfirmed){window.location.href = '/Level2.html'}})",

   death:

   "Swal.fire({icon: 'error', title: 'You Died!', text: 'Reloading the same Level!', showConfirmButton: false, timer: 2000}); this.load_map(map);"

}

3. **Icons 8:** Icons8 is a web platform that provides the capability to download logos.

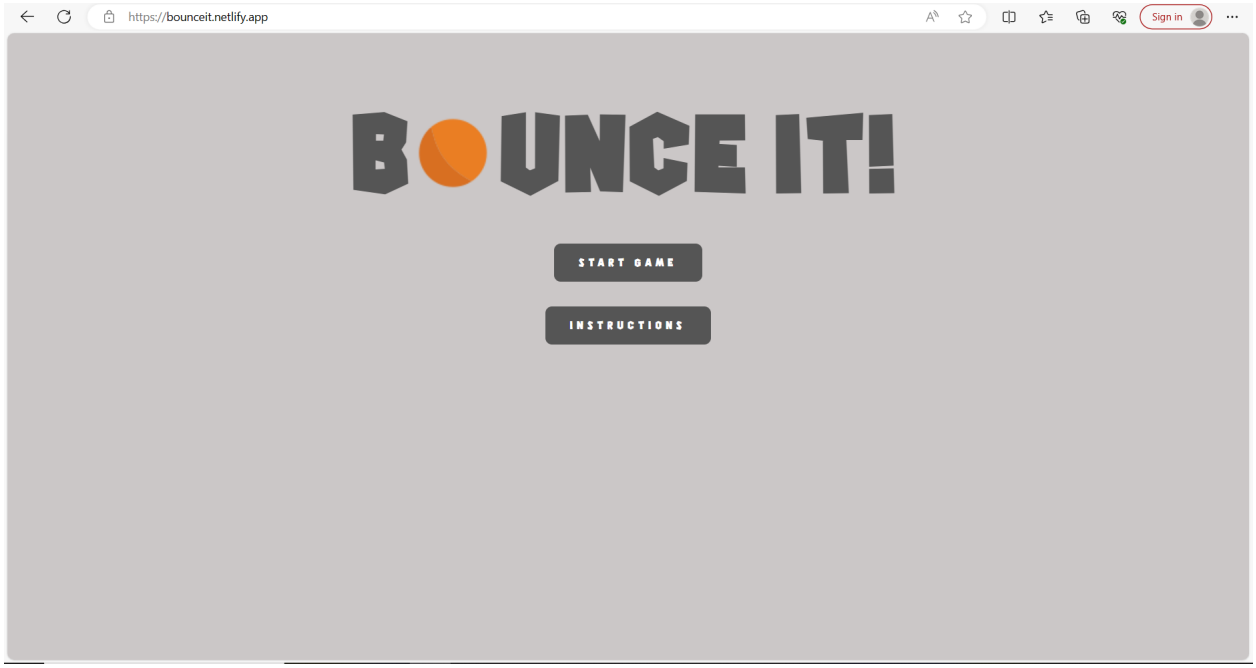
   <link rel="icon" type="image" href="/logo.png"/>

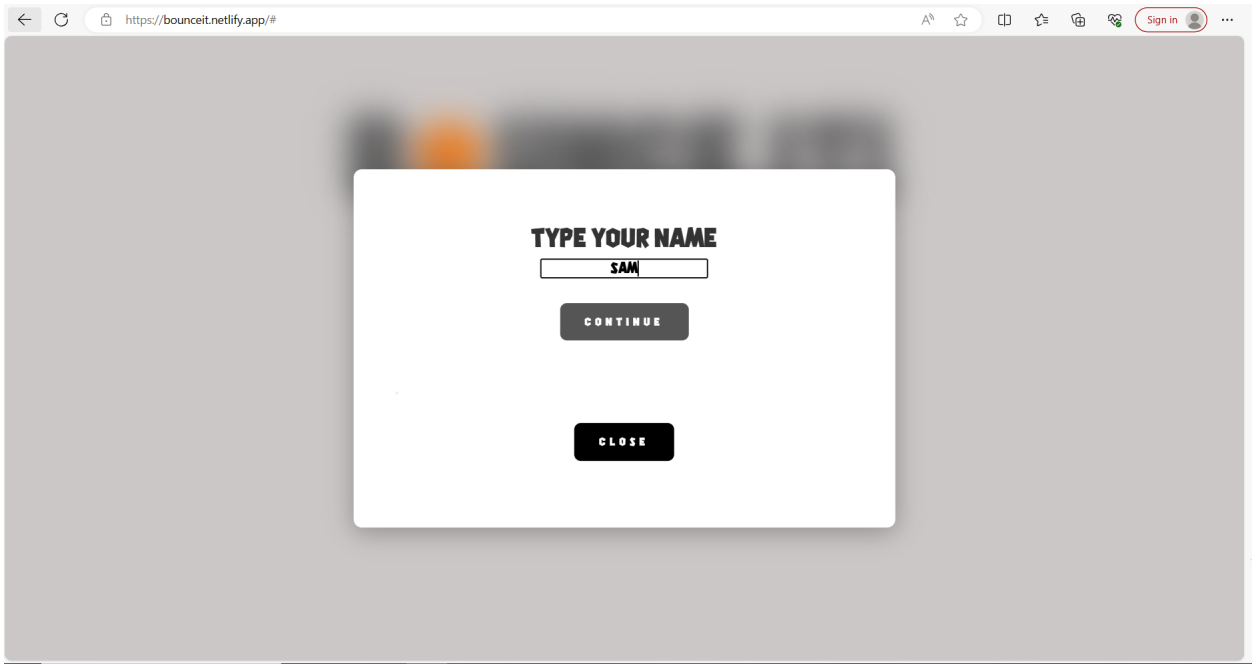
4. **FontsGet:** FontsGet is a web platform that affords us the ability to choose and utilize custom fonts for various applications.

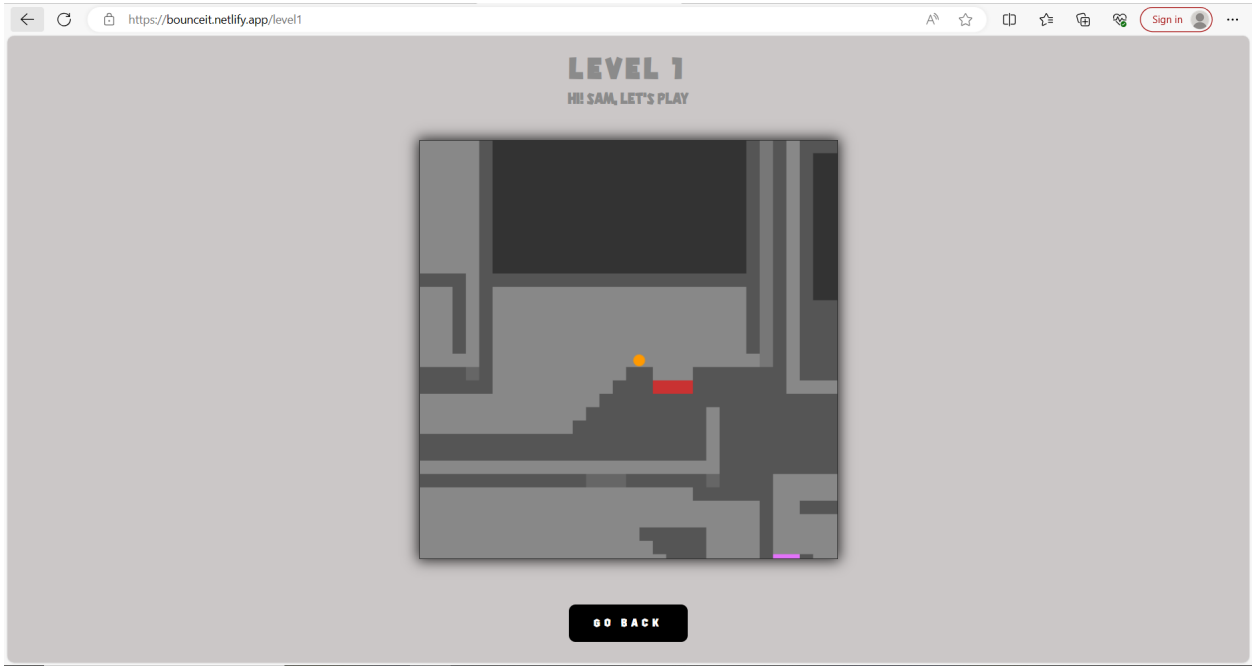
   @font-face {
    font-family: game;
    src: url(Squirk.otf); /*Change the Source URL to change font*/
   }

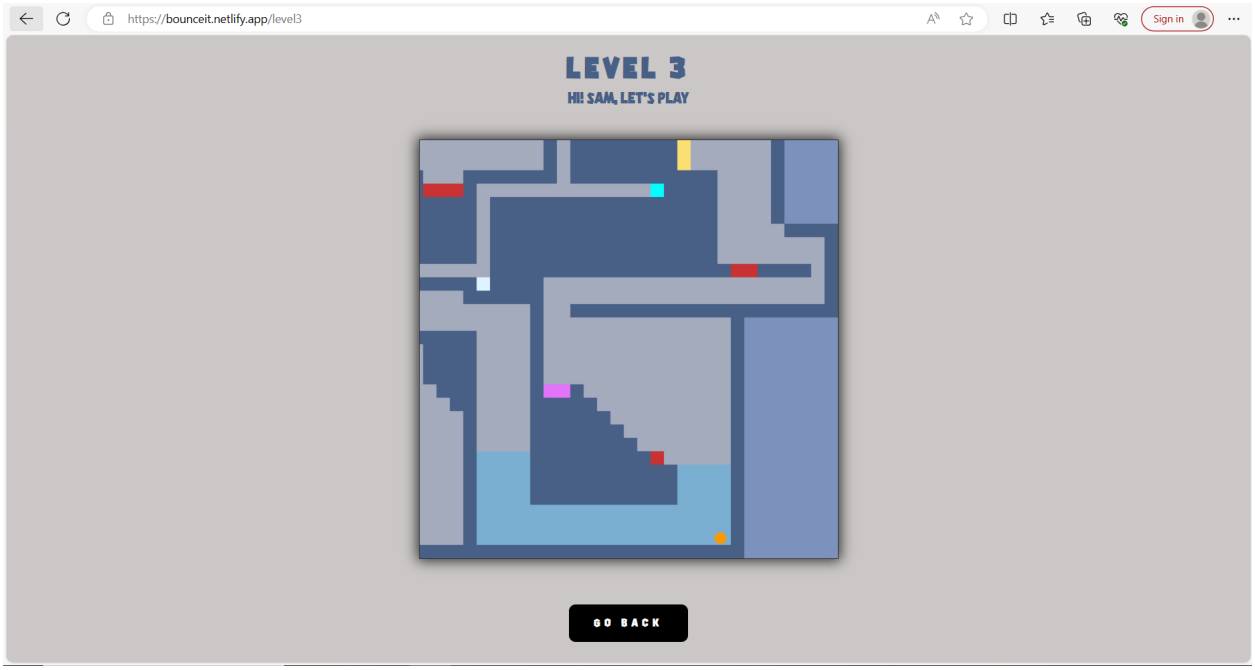
5. **ColorSpace:** Color Palettes Generator and Color Gradient Tool.

- **Screenshots:**




https://bounceit.netlify.app/#
Sign in
TYPE YOUR NAME
SAM
CONTINUE
CLOSE


https://bounceit.netlify.app/level1
Sign in
LEVEL 1
HI! SAM, LET'S PLAY
GO BACK



- **Game Link:**

  Bounce It

- **Project Repository:**

  GitHub - Bounce-It/Bounce_It

- **Reflection:**

  INTRODUCTION

  The Bounce It game project, a dynamic web application leveraging HTML, CSS, and JavaScript, alongside integrations with the canvas, jQuery, and sweetAlert2 APIs, was an ambitious venture from its conception. Over 1.5 months, our Scrum team worked diligently to bring this vision to life. This reflection aims to highlight our journey, the challenges faced, and the lessons learned throughout the Scrum cycles.

  SPRINTS OVERVIEW

  Over six weeks, we organized our development cycle into three two-week sprints. The initial sprint focused on setting up the game's basic architecture and integrating the canvas and jQuery APIs. The second sprint was centered around game dynamics, user interface, and integrating sweetAlert2 for enhanced user feedback. The final sprint was dedicated to testing, refining, and finalizing our product.

  SUCCESSES

  1. ***Clear Vision from the Outset***: Having a precise vision for our game from the beginning was instrumental. Team Discussions presented clear requirements, ensuring the team had a mutual understanding of the game's objectives and desired features.
  2. ***Adaptability***: Despite a few unforeseen technical challenges, especially around the canvas and sweetAlert2 integration, our team showcased remarkable flexibility. We adjusted our strategies in real time, ensuring that the development process remained unhindered.
  3. ***Effective Team Collaboration***: The synergy within our team was commendable. Open communication, paired with daily stand-ups, fostered an environment where everyone felt valued and heard. This collaborative spirit was instrumental in navigating challenges and ensuring timely deliveries.

  CHALLENGES

  1. ***Integration Hurdles***: Integrating multiple APIs, especially Canvas with jQuery, posed initial challenges. There were instances of overlapping functionalities and unexpected behaviors, which required additional time to debug and resolve.
  2. ***Scope Management***: As the project progressed, new ideas and enhancements surfaced. Balancing these with the existing backlog, while ensuring we didn't deviate from our primary objectives, was occasionally challenging.
  3. ***Testing on Different Browsers***: Ensuring compatibility across various browsers required additional effort during our last sprint. Some unexpected discrepancies emerged, especially in older browser versions, which demanded prompt solutions.

LESSONS LEARNED

1. ***Early Prototyping***: A prototype in the early stages could have helped identify potential integration issues sooner. Future projects could benefit from allocating time for this during the initial sprint.
2. ***Regular Backlog Refinement***: Continuous attention to the backlog, ensuring it remains prioritized and up-to-date, is crucial. We realized the significance of this particularly during the latter half of the project when the scope began to expand.
3. ***Enhanced Testing***: While our testing phase was robust, integrating continuous testing throughout the sprints could have flagged browser compatibility issues sooner.

CONCLUSION

The *Bounce It* game project was a valuable learning experience for our Scrum team. From understanding the intricacies of API integrations to navigating the complexities of scope management, each challenge presented an opportunity for growth. As we move forward to new projects, the lessons from *Bounce It* will undoubtedly serve as guiding posts, ensuring even more streamlined and efficient Scrum cycles in the future.

Setup and Installation:

The inclusion of clear instructions for setting up and running the game locally is a significant asset. Cloning the repository and using a popular code editor like Visual Studio Code makes it accessible to developers interested in understanding the game's codebase. The choice of the tech stack for this type of game is:

1. **HTML/CSS:** These technologies are used for structuring the game interface and styling it. The design, while not overly complex, is clean and functional.

2. **JavaScript:** JavaScript plays a pivotal role in enhancing interactivity and features. It handles the game's physics, collision detection, and the dynamic rendering of the game world on the canvas.

3. **Third-Party Assets:** The use of jQuery simplifies various tasks, including event handling and Ajax calls. It streamlines the code and helps maintain compatibility across different browsers. The integration of Sweet Alert 2 for custom alert boxes adds a polished touch to the user experience.

4. **Icons and Fonts:** The inclusion of custom icons and fonts from Icons8 and FontsGet respectively is a subtle yet impactful aspect of design. It adds uniqueness to the game's visual identity.

5. **Color Palette:** The use of ColorSpace for generating color palettes and gradients is a thoughtful touch. The game's vibrant and visually appealing color scheme adds to its overall charm.

Scripts and Customization:

The game's scripts section, which includes scripts for advancing to the next level and handling player deaths, demonstrates the developer's attention to detail. These scripts are both functional and visually appealing, enhancing the user experience.

- **Takeaways:**

To summarize, “BOUNCE IT!” is a shining example of how careful consideration of design and the smart use of development tools and libraries can result in a fun and engaging game. Its simplicity, coupled with the right balance of challenge and reward, makes it appealing to a wide audience. Moreover, the detailed setup instructions and the use of third-party assets showcase the developer's commitment to delivering a quality gaming experience. This game serves as a valuable case study for aspiring game developers and showcases how thoughtful design can turn a simple idea into a fine gaming experience.