Technical Report by Six-Sided Dice

Project: 17T

Technology Enhanced Tabletop Board Game

Table of Contents

Introduction

The TEBS (see Figure 1) is an innovative, self-contained, technology-enhanced tabletop board game prototype that offers an unique social gaming experience. The device merges physical components such as Hall effect sensors, game pieces and push buttons with a digital interface, providing a hybrid experience that differs from purely digital games.

Figure 1: The TEBS - Technology-Enhanced Board Game System

The primary goal of the TEBS system is to modernise the traditional board game format with digital interactivity while maintaining the tactile elements. It also encourages social interaction, allowing players to enjoy games with friends or family in person, rather than playing alone or remotely online.

The system is designed to host multiple games, each incorporating distinctive mechanics such as minigames that increase the challenge and interaction of conventional board games. The first game available on this system is a reimagined version of Snakes and Ladders, featuring unique game rules (see Tools and Technologies (Backend) section for a description), different game board layouts and minigames that create a fresh and enjoyable gaming experience.

In addition to its high entertainment value, this system is an evolving product capable of supporting an expanding library of games, making it an innovative solution for modern tabletop gaming.

Competitive Landscape

Whilst brainstorming for our prototype, we researched existing interactive tabletop games in the commercial market and hobby community. This was useful not only to get ideas of how different types of technology were integrated into board games, but also to avoid developing a product that already existed. These are some of the different types of tabletop technologies we discovered during our research. We found that many of those products were expensive, impractical, or could only handle a single game.

RFID-Based Games

RFID tags and readers have been used in games to accurately track game pieces or identify different cards. However RFID technology is more useful for card games where the tags can be hidden within the cards and only one reader is required. (Robotgeek_official, 2024) The size of RFID tags and cost of multiple readers made them physically and economically impractical for board games with small grid squares.

An example of incorporating RFID technology into a board game is the Mindsports International Scrabble Board (see Figure 2) which debuted at the Prague Mind Sports Festival in 2012 (Fincher, 2012). The construction of this prototype cost US$31,000. Each of the 100 letter tiles contain a unique RFID tag and the 225 squares on the gameboard have individual RFID antennas (Wiegler, 2014). While its ability to graphically display the board state with player statistics in professional competition may justify its cost, this product is not commercially available and not economically suitable for a domestic market.

Figure 2: Scrabble by Mindsports International

Electronic and Sensor-Based Games

Other games used electronics and sensors to enhance gameplay. The Lexibook Harry Potter Interactive Electronic Chess Computer (see Figure 3), available for purchase through Amazon for $208 (Lexibook CG1300 Chessman Elite Interactive Electronic Chess Game, 2024), detects piece movement through pressure-sensitive squares. However, this technology results in the players adopting an unnatural movement when moving pieces across the board as they tap on and off a square. (Whitby, 2023)

Figure 3: Interactive Electronic Chess Computer

Figure 4: Beyond Tablet

Instead of being restricted to playing one game, products like the Beyond Tablet (see Figure 4) (Lkmuchik, 2019) and the YOYN Interactive Board Game (YOYN Interactive, 2010) use physical printed game board overlays on top of electronic devices to allow different games to be played. Although this adds flexibility, we did not like the need to swap between different physical game boards, and preferred a system where different boards were automatically displayed.

We also examined a custom gaming table (see Figure 5) that used addressable LEDs, a manually operated number pad and an Arduino to select player position and control lighting effects within the table's walls. A map was projected onto the table from an overhead projector. A player used controls to manually communicate user interactions via the table's built-in technology (Chuon, 2022). However, shadows from players' hand movements over the projected map obscured the board and disrupted gameplay, so overhead projection was rejected as a suitable method of displaying a game board.

Figure 5: Gaming Table with projected map

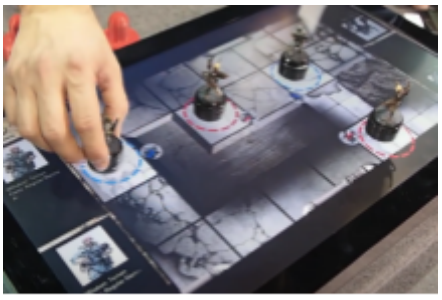

Figure 6: ePawn Arena

The ePawn Arena (see Figure 6) tracks the movement and rotation of figurines across a 26-inch screen using magnetic sensors. This device incorporates accurate physical piece recognition and game design flexibility. However this system also sends the game data to an external computer for processing (Tekunoloji, 2012). In our opinion, incorporating the game processing within the tabletop device would create an all-in-one system and improve user experience by making the device self-contained and more intuitive to use.

Online Gaming Platforms

Online gaming platforms like Tabletop Simulator offer a wide range of board games in fully digital or virtual 3D environments. However some games on those platforms require VR equipment, while others require intensive graphic rendering, which sometimes results in slow gameplay, bugs and other performance issues (Adam the Fanatic, 2021). These games attract a niche group of players that enjoy immersive and remote digital experiences, but the technical requirements and performance issues may make them less appealing to casual gamers, and are more suitable to connect remote players with each other.

Digital Gaming Tables

Similar to our ultimate prototype design, the 32 Infinite Game Table and Game Board by Arcade1Up allows players to gather around a large touch screen to play games, providing a similar experience to playing games on a tablet, but also substitutes tactile game elements such as dice rolling with haptic feedback through the screen. It has partnered with Hasbro to provide traditional games such as Monopoly as well as custom games. Both the Infinity Game Table and Game Board are currently only available for sale in the US and Canada for US$999.99 and US$499.00 respectively (Game Night Fun Elevated, 2024).

The New Zealand Best Design Awards winner in 2016 in the Student Design category (see Figure 7) uses a combination of tactile pieces and touch screen technology. Users place physical blocks on a touch screen to play four different games, for the purpose of assisting children with fine motor skills. Although not commercially available, this product received the following positive comments from an award Judge:

> *A compelling game mechanic and platform supported by a great combination of hardware and software. The table's minimalist appearance belies the thought put into its gameplay, which enables play from all sides, ties learning into its tactile approach, and improves fine motor skills for kids.* (Designers Institute of New Zealand, 2016)

Figure 7: Orbit Rescue – One of four games in the NZ Best Design award winning interactive game.

By analysing these existing commercial and custom board game designs and technologies, we identified several innovative approaches to interactive tabletop gaming. We also noticed some limitations due to expensive components, impracticality or single-game inflexibility. These factors helped guide the direction of our prototype to one which uses cost-effective technology within a self-contained device to play unique board games, whilst keeping the social and tactile aspects of traditional board games.

Prototype Implementation Choices

Designing a physical tabletop board game is a very open ended challenge. So for this, we narrowed it down to three major decisions: the form of the device, the types of games it would support, and the method of player interaction or input.

We observed that many of the existing technology board games focused on either a chess or checkers style game, played only one game, or required different board overlays for different games. This showed a gap in the market for a system that could play multiple games, answering our second question.

That answer significantly narrowed down the scope of our first question. If we wanted something to play many games, what could we do to digitally enhance most, if not all, of them? The simple option was to use addressable leds on each square, notifying a player with coloured lights. While this would work, it didn't allow us much flexibility for user feedback and dynamic game instructions. Several of us have previous experience designing and coding web pages, so we decided to use a web-app style design for the graphical user interface and a monitor screen as the physical playing surface.

We also needed a way for the players to directly interact with the game. Touch screens were ruled out quickly as they were very expensive for any size over 8-10 inches, including retrofitting a touch

panel to the front of any display. Therefore an LCD screen was preferred as it was a thin surface, and a second-hand screen could be economically obtained.

We recognised the importance of incorporating physical game pieces as a method of player interaction to preserve the tactile feel of traditional board games. However this meant that we required a sensor of sufficient size and range that could accurately detect the game piece through an LCD screen. We identified a few budget-friendly options, namely reed switches, near-field communication (NFC) sensors, and radio-frequency identification (RFID) technology.

RFID and NFCs were ruled out quickly as they would cause the size of each grid square to be too large for our display. Reed switches return a voltage range based on their distance to a magnet, and are often used in window sensors. Magnetic sensors would normally be a good option to detect player pieces as they would work through almost any material, except metal. However testing showed that the reed switches didn't have sufficient range to detect magnets on the other side of an LCD screen.

Hall effect sensors were finally chosen for this task due to their flexibility as an electrical component whilst having sufficiently strong magnetic detection. They have a good range and very fast response which has allowed us to hide the whole sensor array underneath the screen.

We also needed ways to advance player turns. For this, we chose to use a large button, and later two buttons, for players to push, which further distinguishes the experience our device provides from a computer.

Having the flexibility of an LCD panel plus a hidden array allowed us to keep costs cheaper than using a touch screen, while still having great interactivity. We are able to display many different games that can fit within a 10x10 grid, including board games that do not conform to the standard grid shape.

Initial Game Choice

Although our prototype's physical construction and software design provides the flexibility to play a wide range of games in mind, Snakes and Ladders was chosen as the first game for several strategic reasons. First, its 10 x 10 grid is larger than the standard 8 x 8 grids used in games like chess or checkers, demonstrating the device's ability to handle a game with many small grid squares. This setup also allows us to display fewer squares or enable multiple sensors to recognise a game piece within a larger area.

The game of Snakes and Ladders originated in ancient India and was introduced to Great Britain in the 19th century, meaning it is in the public domain (Mingren, 2018). We could therefore use and modify the game's core mechanics without breaching copyright, provided we did not copy specific designs created by other manufacturers. This allowed us to modify the game mechanics, making it an ideal initial game for our prototype.

Movement mechanics within Snakes and Ladders is more complex than in some grid-based games. Players move forward by wrapping around rows in alternating directions, and moving up ladders and down snakes, which is more complex than simple linear movement. This highlights our software's ability to handle complex game movement.

Similar to traditional board games, the addition of physical game pieces means we cannot physically prevent players from moving their pieces incorrectly on the board. However, the software provides feedback about an incorrect position and prompts players to follow the game rules, reinforcing fair play.

Paper Prototyping

After receiving stakeholder feedback that emphasised the importance of user experience, we stopped development on the physical device for a couple of weeks to concentrate our efforts on creating a better game. Multiple paper prototypes of the Snakes and Ladders game were developed to examine player interactions and validate proposed game rules. User testing sessions with the paper prototypes were conducted amongst our team and also with family members, which led to game rules, mechanics and instructions being adopted or rejected. Paper prototyping was also continually used throughout the project to communicate the finer details of player movement with team members.

Paper prototyping and user testing resulted in the expansion of the traditional Snakes and Ladders game rules to promote competitive play. It also compensated for some technical limitations (eg. a sensor under a grid square cannot recognise more than one game piece on one square). User testing also informed us that physical dice rolls were not required to progress the game. The introduction of card instructions to output the movement of players as well as other mechanics (described below) increased the game's enjoyability.

UI Design

Figma was used to create different portrait and landscape layouts for the device screen. These designs delineated specific areas for the game board, game information, game title and team branding to house a game board of up to 10 x 10 grid squares or game positions. The designs also took into account the physical constraints imposed by the proposed placement of sensors and other components beneath the screen.

Developing these initial designs in Figma guided our decision to use CSS Grid rather than CSS Flexbox when styling the larger interface sections. CSS Flexbox is beneficial for creating responsive layouts, but it can lead to child components affecting neighbouring elements more easily. Alternatively, CSS Grid can be used to partition the display into separate components, preventing child elements from escaping their parent containers. (Powell, 2022) This provides more precise control over the positioning and containment of different parts of the interface, keeping them separate whilst still allowing for interactive changes within each area. This modular design also supports future game expansion by ensuring different game grids can be easily positioned over the physical board sensors, and other components such as game titles and information can also be appropriately displayed.

Tools and Technologies

Hardware

We were making a tabletop board game, so the most important for any hardware implementations was to keep it feeling like one. To us, this meant that we needed a physical playing surface and interaction with player pieces.

Our first form of interaction took the form of a large button that was used to advance the game or start a new player turn. Buttons are a common interaction method for arcade games. A similar concept was developed for the 1960s game Trouble (Coopee, 2018) with their Pop-O-Matic™ bubble that, when pressed, would pop a metal plate, causing a dice to jump around inside a plastic dome. The button interaction was also popular with our playtesters with one criticism; if the button is on the opposite side of the board to the player it can require a long reach to press. This led to the addition of a second button to prevent players from reaching across the board and regularly bumping the game or pieces. It also allowed games to receive two distinguishable inputs, as used in the Snakes and Ladders setup phase.

Hall effect sensors were adopted as the other method of player interaction due to their flexibility as an electrical component whilst having sufficiently strong magnetic detection. Each sensor switches on when a magnet is within range, and can work through a non-metallic playing surface. Arranging 100 of these to match a playing grid surface provided us with an accurate piece detection solution.

The playing surface of our game needed to be interactive. We decided that an LCD panel would be a good choice as, including the backlight, the panel is thin (around 5mm thick) to allow for magnets placed on the surface to be detected by the sensor array underneath. A screen could also be sourced easily and cheaply via secondhand marketplaces, and then dismantled to get to the LCD panel inside.

A computing device was required to interpret the sensor array data for use with the game program. This is where a popular tinkering computer the Raspberry Pi (RPi) comes in. It is a cost effective Single Board Computer (SBC) that is popular in both hobby projects and industry for its small size, ease of programming, flexible IO ports (Klosowski, 2021), and it runs a Debian based distribution of Linux: Raspberry Pi OS (Raspberry Pi, n.d.). This means we could plug our sensor array straight into the computer and program directly with its input, thereby running any Linux-compatible software.

Backend

Initially, we discussed the programming paradigm that would best suit the backend software for our prototype. We compared the benefits of using functional programming (Turing, n.d.) as opposed to object-oriented programming (Microsoft, 2023).

The functional programming paradigm utilises pure functions to implement logic. The main reasons we considered this was its native parallelisation support, ease of use, simple debugging, and efficiency (Turing, n.d.). However we foresaw several issues that would arise during development. All code loops and iterations would need to be implemented recursively, which would likely overcomplicate the backend architecture and thus increase development time. This might also impact any efficiency gains from using pure functions, as recursion can often be more computationally expensive than iterative methods. These flaws led our team to gravitate towards using the more familiar object-oriented paradigm in Python.

The backend was designed using Object Oriented Programming (OOP) (Microsoft, 2023) principles. OOP is a well-established design philosophy that provides a flexible, yet robust set of guidelines to write intuitive, modular code that scales to meet the demands of the program. The four main principles of OOP, which are abstraction, encapsulation, inheritance, and polymorphism, have enabled a straightforward implementation to meet our prototype requirements.

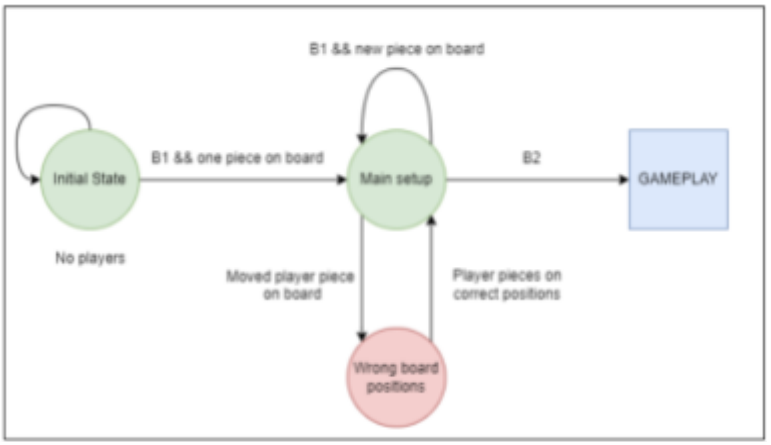

Figure 8: State Diagram of the game's setup phase. (B1 and B2 refer to button one (blue) and button two (red) respectively)

Abstraction allowed the model to be divided into six class states, further simplifying the code logic. These states are:

- **Setup**: Handles adding and removing players; starts the game after all players join (see Figure 8);
- **Gameplay**: Begins each player's turn and prompts player actions;
- **Draw Card**: Handles the card effects and resolutions;
- **Minigame**: Begins and plays activated minigames;

- **Game Over**: Signifies the game is over, declares the winner, and provides instructions for restarting the game; and
- **Incorrect Square**: Handles position error checking and notifies the incorrectly-positioned player.

Apart from card instructions directing players to move between one and six spaces forward, there are also six new mechanics that keep the game from being purely linear, and introduce elements of surprise, namely:

- **Take the Lead**: The card allows the current player to swap places with the leading player. This keeps the game competitive, ensuring no one can feel secure in their lead position.
- **Helping Hand**: This card helps players who are behind, closing the gap between them and the second last player. This gives them a chance to catch up and stay in the game.
- **Reverse Snakes and Ladders**: This card swaps the snake and ladder positions, adding or removing opportunities to land on a snake or ladder.
- **Descend Next Snake**: A player moves forward to the next snake, and then down its tail. Depending on how close they are to the next snake, and the length of its body, this can either be a shortcut or place a player far behind their previous position.
- **Overslept**: This causes a player to forfeit their current turn.
- **Jump Ahead**: This causes all players to jump ahead, potentially causing them to land on nearby ladders or snakes they had failed to reach on the previous turn.

If a player lands on a snake's head or ladder's bottom rung during a movement card (excluding the six new mechanics cards), then a minigame is activated. The player must win the minigame to climb up the ladder or avoid sliding down the snake's tail. There are three minigames that may randomly occur:

- **Wild Goose Chase** requires the player to physically slide their game piece across the screen to catch all the geese on the board before the timer ends.
- **Falling Fruits** requires the player to catch all the ripe fruits, but avoid catching the rotten ones (see Figure 9).
- **Pest Control** gives a player a single chance to swat a fly on any of the coloured squares before the timer ends.

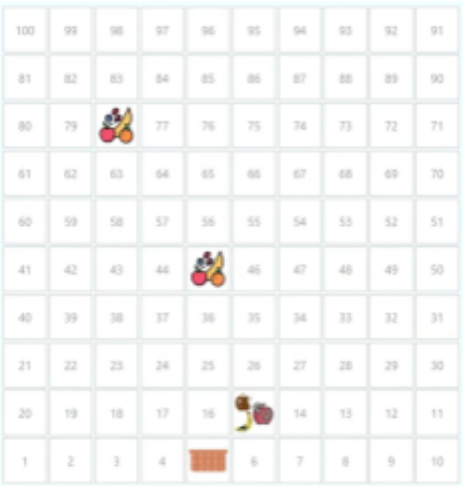

Figure 9: Falling Fruits minigame

The use of encapsulation has kept the scope of the code clear and well-defined. Handling the above states through only intended interactions has kept bugs to a minimum. Inheritance and polymorphism were used extensively in both the card and minigame mechanics.

To facilitate communication between the backend and frontend components we opted for an internally run server. We used the FastAPI (Tiangolo, 2024) web framework to quickly deploy a server that can both receive post requests and broadcast to all connected clients, as well as communicate asynchronously. For the communication itself, we used the WebSocket protocol.

The WebSocket protocol (Internet Engineering Task Force, 2011) was used to send real-time JSON data from the backend server to the frontend app. While REST is more commonly used for typical web apps, with REST the frontend can only retrieve data asynchronously by polling the server (Amazon Web Services, 2024). This is not suitable for our project as the display needs to show the most current game information and provide immediate feedback after moves. Alternatively, by using WebSockets, a TCP connection can remain open between the backend server and frontend client. This means that the frontend immediately receives any data sent from the backend and can update the display promptly, making it a more suitable option. There is also a wide range of libraries that

support WebSockets, allowing for faster development and iteration. For the backend server FastAPI (Tiangolo, 2024) was used, and for the frontend react-use-websocket (Taussig, 2024) was used.

Frontend

The React.js framework was used to create a frontend web app for the game display. A React web app was chosen for the display over a game engine such as Godot (Godot Foundation, 2024) or an integrated Python GUI framework such as Qt (The Qt Company Ltd, 2024), because it is a well known and versatile framework. Additionally, React's declarative, functional, and component based programming model allows for more modular (see Figure 10) and less bug-prone code over traditional imperative JavaScript (Meta Platforms Inc, 2024). It also enables developers to more intuitively translate components from design prototypes to code, supporting faster iteration (Meta Platforms Inc, 2024).

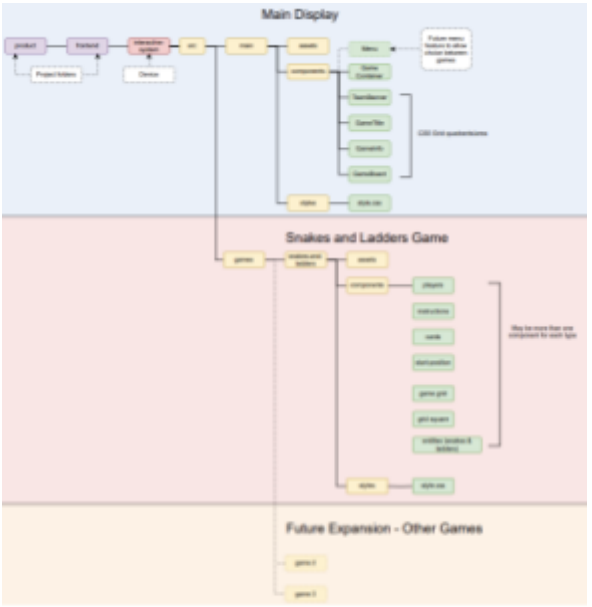

Figure 10: Diagram showing frontend code modular structure that can accommodate future expansion

Initially, the project was started with Create React App (CRA) (Facebook Inc, 2022) as the build tool. However, the build tool has been switched to Vite (VoidZero Inc, 2024) for speed and React best practices (also discussed in Security). Compared to CRA, Vite has faster page loading, hot module replacement update time and build time, as well as a smaller output build size (Omotayo, 2022). This makes Vite more optimal for running on a microcontroller where hardware resources are limited.

Figma was used to create initial game designs and layouts, so that our game interface would be consistent with the proposed hardware layout and modular design of the device. Figma was also instrumental in exporting royalty free PNG icons to SVG images so that they could be modified for use as minigame images in the prototype.

Another important consideration for our device was that it should be visually accessible, given it was being built for a domestic market. The Web Content Accessibility Guidelines (WCAG) provide guidance for developers to create accessible and inclusive user experiences for digital products. "Specific requirements… ensure text and images contrast sufficiently against their background, allowing for easy readability and visibility." (Awad, 2024)

To implement these recommendations, we used Figma's A11y Color Contrast Checker plugin (Gorr, n.d.) to confirm that our various colour and font choices met the WCAG's Contrast (Enhanced) success criterion AAA rating (see Figure 11), meaning there is a very high contrast ratio between foreground and background colours. (Web Content Accessibility Guidelines (WCAG) 2.1, 2023)

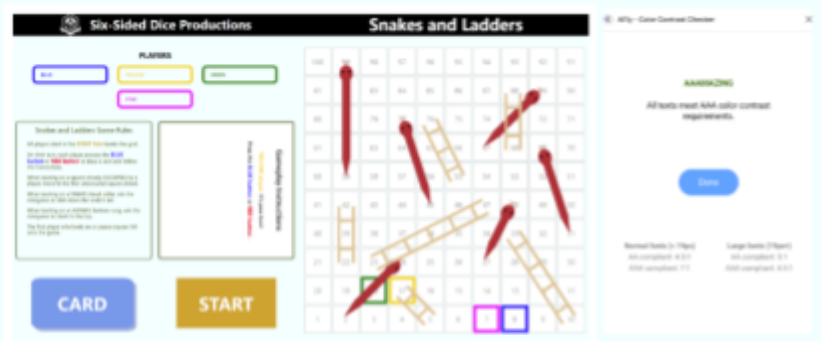

Figure 11: Results from A11y Color Contrast Checker

Figure 12: Color Blind plugin test

Using a Figma plugin called Color Blind (see Figure 12), we also confirmed that the colours chosen for the player colours and main grid components were distinguishable by persons experiencing one of the 7 colour vision deficiencies (Mason de Caires, n.d.).

Project Management and Collaboration

Throughout this project, our team was guided by the four values and twelve guiding principles of the Agile framework (Beck et al., 2001) to rapidly develop our prototype while maintaining the flexibility to handle unforeseen changes (Landau, 2022). We attempted to uphold the Agile values of "Individuals and interactions over processes and tools" and "Responding to change over following a plan" (Beck et al., 2001) to stay adaptable as new challenges arose. We also structured our practices in accordance with the Scrum framework's three pillars of transparency, inspection and adaptation (Schwaber, n.d.) which assisted us to implement continuous feedback loops and frequent adjustments to the project timeline, task list and our workflow. The Agile framework of project management is ideal for projects that have a fixed schedule (in our case, 11 weeks) and resources (six team members) while the scope (features we could build) may vary (Aljaber, n.d.).

Team members' responsibilities were divided into three subgroups: Hardware, Backend and Frontend:

- The **Hardware** team was responsible for building the physical device, ensuring the sensors accurately detect player piece movement on the game board, and sending that data to the software controller for processing.
- The **Backend** team was responsible for managing the game's logic and mechanics, such as movement rules and game state changes.
- The **Frontend** team handled designing the user interface and user experience, visually displaying the correct information in the user interface, and testing that the game functions correctly to ensure all actions between the players and the game are intuitive.

This division of responsibility allowed team members to focus their attention on different areas of expertise, whilst maintaining collaboration across all three teams. Additionally, the concept of pair programming was a major influence here. We didn't want to have one person solely responsible for any one part to mitigate the risk that a team member was sick or unable to complete their task. It also allowed knowledge to be better spread among team members and resulted in improved code quality (Agile Alliance, 2024).

Consistent with the Agile principle that "Working software is the primary measure of progress" (Beck et al., 2001), our teams worked iteratively to deliver and test working software and hardware within weekly sprints. However, the importance and complexity of delivering software for this project, and the reluctance to purchase more expensive components until the project had progressed further, delayed hardware development in the middle stages of our project. We also underestimated the complexity of soldering the large quantity of hardware components together and integrating the hardware system with the evolving software, despite continuous testing of the individual components. We now recognise that completing the hardware build earlier would have provided more time to identify and resolve the multiple hardware-related problems we encountered. Fortunately we were able to pivot quickly in the later stages of our project to address these problems. However, focusing on this earlier would have better aligned with the Agile principle of "Continuous attention to technical excellence and good design enhances agility" (Beck et al., 2001) to progress the whole device's completion in smaller, iterative stages (Landau, 2022).

Collaboration across the whole team is crucial to iterative design, production and testing. To enable this, we regularly used Discord, Trello, Google Drive, and GitHub for the following purposes:

- **Trello** allowed all team members to manage the list of tasks to be completed each sprint, or to draw attention to bugs or blocks that prevented team members from completing tasks.
- **Discord** allowed us to communicate daily via voice, video or text without requiring all team members to always be present at once. We also posted screenshots of our progress, discussed ideas or advised other team members of important information, building an invaluable record of our team communication and decisions throughout the project.
- **GitHub**'s version control and merge features allowed us to work collaboratively on the same files within the codebase without impacting other team members' workflow.
- A shared **Google Drive** allowed us to securely store videos and photos of our progress and share team documentation easily.

These four tools were invaluable in allowing us to "Satisfy the customer through early and continuous delivery of valuable software" (Beck et al., 2001) as a highest priority.

We structured our workflow using the Scrum framework, which helped us to stay organised and focused on our project goals. Stuart Moyes, as the product owner, was responsible for communicating the project goal and managing the product backlog's list of outstanding project tasks. During our sprint planning sessions at the start of each week, Stuart would prioritise tasks for inclusion in the sprint backlog, where team members took equal responsibility in delegating and deciding which tasks to take on for that sprint (Schwaber, n.d.).

Although team members took on some of the scrum master's responsibilities, such as communicating Scrum procedures and addressing blockers, the role itself was not formally assigned. None of us had prior experience working in an Agile team, and we wrongly assumed that implementing sprint planning sessions and delegating tasks would cover these responsibilities. We failed to recognise the full scope of the scrum master's role, which includes guiding the team in scrum practices, fostering self-organisation and ensuring the removal of impediments (Schwaber, n.d.).

Due to time constraints imposed by other course commitments, we adapted the daily Scrum to a weekly stand-up. In these meetings, team members shared their progress, discussed any blockers and identified their next steps (*The Daily Scrum Is NOT a Status Meeting*, 2017). While this approach helped with transparency and kept the team synchronised to some degree, it did not provide the continuous inspection and adaptation that daily stand-ups offer, which are the remaining two pillars of the Scrum framework (*Three Pillars of Scrum: Understanding Scrum's Core Principles*, n.d.). We relied heavily on Discord communication throughout the week to compensate for this, but regular in-person or virtual check-ins might have helped us identify and resolve issues more quickly.

Overall, using the Agile and Scrum frameworks enabled us to be collaborative, flexible and deliver iterative results, however the need for more regular inspection cycles was also highlighted.

Ethics, Security and Data Privacy Considerations

When developing our board game system, we considered several ethical, security and data privacy issues and made the following decisions to address them. We also considered the risk assessment created at the start of the project, and how we mitigated the risks that occurred.

Ethical Considerations

Safety warning for young children: Because our system includes small player pieces, magnets and external cords for buttons, we are required to include a choking hazard warning for young children, particularly those under 3 years of age. This complies with the *Consumer Goods (Toys for Children up to and including 36 Months of Age) Safety Standard 2023* (Australia: Adoption of the New Safety Standard for Toys for Children 3 Years Old and Under, 2023).

Affordability and suitability for purpose: We aimed to create a product that is cost-effective and accessible to families. This decision guided our material selection and design process, ensuring the device is affordable, easy to use and contains clear instructions. Additionally, the game was designed for offline play, which eliminates any exposure to third-party ads, promotions or microtransactions. This not only makes the game more family-friendly, but also ensures it does not exploit users through in-game financial transactions.

Accessibility and inclusivity: We prioritised accessibility and inclusivity in our system's design to ensure players with diverse needs could participate equally. Two features reflect this. The orientation of text on the device was specifically designed so that players seated on all sides of the board could view it without difficulty. This approach supports inclusive participation (Roussey, 2024), ensuring that all players can easily read instructions and game details regardless of their position around the board. Also, to help players with colour vision deficiencies to distinguish between the different game piece colours and board positions, appropriate colours were chosen and colour blind tests conducted on the design (see Tools and Technologies section). In future iterations of our prototype, we plan to enhance accessibility by incorporating voice instructions and prompts to assist young players or persons with reading difficulties, and multiple difficulty settings to adjust the game's challenges to players skill and comfort levels (Roussey, 2024).

Proper referencing of third-party software: In line with ethical software practices, our team took care to properly attribute all third-party libraries and icons used during the development process. We are currently using royalty-free, placeholder icons from Flaticon in our prototype's game, however we intend to replace these with custom artwork before the final production phase to ensure our game is original.

Security

General Hardware Security: The current prototype is contained within a removable acrylic and cardboard structure, to allow ease of access for debugging and testing. To improve security in future production models and to prevent unauthorised access or tampering, the hardware components can be sealed within a more durable plastic or metal structure that, if opened, will void manufacturing warranties.

Raspberry Pi Software: For the prototype device, we modified the default username and password on the Raspberry Pi, only known to a few people. This increases the security of the Pi because it makes it less susceptible to possible brute force attacks. (*Increasing Security*, n.d.). The prototype is currently controlled by a USB keyboard and runs a web server on localhost. It only connects to the internet as required to update code and keep dependencies up to date. This further reduces the likelihood of external access through the device itself.

At production, configuration settings on the Raspberry Pi would also be changed to disable password authentication; instead, opting for public and private keys to allow access from authorised devices only (Elder, 2019). Additional configuration settings would include disabling unused USB slots, bluetooth and ethernet ports, as well as downloading a firewall for further access control to the device over a network. These changes would be implemented and thoroughly tested before production.

Encryption: We considered encrypting the Python files to secure the source code. While encryption would enhance security, it includes a performance overhead because it requires the code to be decrypted before it can be used (*Encrypting Python (.Py) Files: A Comprehensive Guide*, 2024). Because the software's execution requires multiple fast calculations and interface changes to provide real-time feedback to players, encryption was not implemented at this time.

Dynamic Board Generation: The software is designed to dynamically generate a new board state for every game. This means that in-game user input does not need to be sanitised because the system will only read input from the hardware. This also minimises the chances of users accessing sensitive information through improper input.

Vulnerabilities in Dependencies: Early in the project, GitHub dependabot reported moderate and high security vulnerabilities in the nth-check, micromatch and PostCSS dependencies used by Create React App (CRA) (Figure 13) (Facebook Inc, 2022). CRA is not regularly maintained (Gaearon, 2021), contributing to these issues. Updating Node and Node Package Manager (npm) did not remove these vulnerabilities, so we switched to Vite, an alternative frontend build tool (VoidZero Inc, 2024), which eliminated the security alerts.

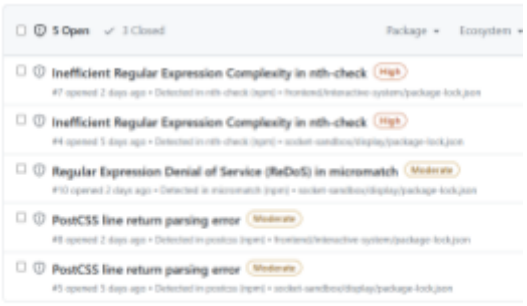

Figure 13: third party dependency vulnerabilities resulting from create-react-app

XSS Protection in React: Our program uses JSON-stored text for in-game card instructions. React automatically escapes raw HTML in strings to prevent malicious cross-site scripting (XSS) attacks that inject malicious scripts, allowing hackers to manipulate data returned by the browser. (Williams, 2024) While the React property dangerouslySetInnerHTML allows raw HTML injection, this solution was not acceptable to us because it bypasses React's built-in protections and reintroduces the XXS security risk. (Bilgili, 2023) As a safer alternative, we used regex commands to split text into different parts for individual JSX processing (Markhvaidze & 0stone0, 2022). This allowed normal HTML tags and CSS classes to be safely applied without using dangerouslySetInnerHTML. This solution was not commonly recommended by external sources (Bilgili, 2023) (alakkadshaw, 2024), but is a safer way to insert HTML in strings within React without using alternative, third-party applications.

Data Privacy

Confidentiality of user information: For the prototype device, we chose not to request or store any user data, ensuring users' privacy. Future versions may introduce user accounts for owners to purchase additional content or updating the device's firmware, where necessary to identify and securely update the device. Any personal data collected will comply with the Australian Privacy Principles (2022), including principle 3.2 which states the collection of personal information must be reasonably necessary for the device's function (Read the Australian Privacy Principles, 2022).

Other privacy policies that relate to device owners were considered, including:

- **Consent**: Owners may consent to having their name and game data stored and used to improve the game experience and suggest similar games to purchase. This would be an opt-in service.
- **Access**: Owners will have access to all information that they wish to provide, and can update the information as desired. Minimal information will be required and sensitive information will be encrypted (such as payment information and device id) using a secure encryption algorithm.
- **Transparency**: Owners will be clearly informed about how their data is being used and if their data is being shared with any third parties (*What Is Data Privacy?*, 2023).

Anonymity of Players: No player accounts are needed to play the games on the device, ensuring the personal information of players is not requested or stored. Instead, players are temporarily distinguished by in-game identifiers. For example, in the Snakes and Ladders game, players select a coloured game piece at the start, allowing them to switch colours between games. This design prevents the identification of players and maintains their anonymity whilst using the device.

Risk Assessment

A Risk Assessment Matrix prepared for the Statement of Work identified several ethical, security and privacy issues that were assumed at that early stage. Here is a summary of that matrix (Figure 14).

Risk	Impact	Likelihood	Mitigation
Problem integrating physical device components with display or game logic.	High	Medium	Research other integration options. Further prototyping, design or research alternative options.
Unable to complete sprint tasks due to sickness/injury.	Medium	High	All teams have dual team members which can provide assistance
Hardware component cannot be sourced in time	High	Medium	Obtain components as soon as prototyping identifies our requirements.
Disagreements resulting in non-engagement by team member	High	Low	Team Leader communicates with team members involved. Advice sought from tutor if no resolution.
Copyright issues with included games/art/design	Low	Low	Credit any art, design, or games included when designing, programming or building our prototype.
User data might be accessed without their consent by third party	High	Low	Offline prototype storing data locally. User data requires permission before being stored.

Figure 14: Summary of Risk Assessment Matrix contents

Integration of physical device components: We experienced short circuits and incorrect data readings when soldering the electrical components and wiring to a blank breadboard. To address this, we ordered custom printed circuit boards that incorporated our wiring designs into the breadboards, leaving sufficient room to properly solder the other components. We were fortunate that this only caused a week's delay in the build. We also had a lot of problems with the game mechanics when integrating the software with the hardware, as these could not be tested until game movement was governed by player pieces rather than a keyboard. And we had problems with artifacts appearing on the screen due to it being removed from its original protective case. It is difficult to prevent this type of risk when building an innovative product, however allowing extra time in the project timeline for unexpected delays can help minimise their impact on the project's progress.

Unable to complete sprint tasks due to sickness/illness: While sprint tasks experienced minimal delays on occasions due to the minor illness of a team member, having two individuals per specialty allowed us to cover responsibilities effectively during these times. However we found that other commitments, such as study obligations and family responsibilities, posed a greater challenge and often affected multiple team members simultaneously. This situation was difficult to mitigate, as not everyone could dedicate one hundred percent of their study time to the Agile project. For future projects, it is crucial to incorporate flexibility and redundancy when forming Agile teams to ensure that tasks can still be completed in the event of a team member's temporary unavailability.

Hardware components cannot be sourced in time: We were fortunate to always be able to source hardware components when required in a short period of time. However delays in postage or supply can occur, so it is important in future projects to consider ordering or purchasing hardware components as soon as their need is established.

Disagreements resulting in non-engagement by team member: We did not have a situation where a team member disengaged due to team disagreements. Our team discussed differences in design or implementation openly during meetings and in our Discord chat. This transparent dialogue allowed us to examine various perspectives and foster a collaborative environment. On some occasions team members did disengage or otherwise failed to communicate effectively, causing progress to stall for

other team members. The procedure was followed in that the product owner was notified, and if that failed to resolve the issue, our tutor was informed, resulting in improved communication over the following week. Maintaining regular communication is important for steady progress during sprint iterations. Escalating issues to management or supervisors early can help resolve conflicts and keep the project on track. In future projects, it would also be beneficial to establish a scrum master role (as discussed in the Project Management and Collaboration section) to facilitate our Agile processes more effectively.

Copyright issues with included games/art/design: We have sought to address any future copyright issues in our prototype by properly attributing our AI generated logo, and the use of royalty-free icons and third-party libraries in our software and this report. It is intended that the proper written attributions and notifications are made in the product's manual, and that the temporary icons will be replaced with custom images prior to the device's production.

User data might be accessed without their consent by a third party: The prototype does not use any user data, but we are committed to the security of any potential owner data. In future iterations, we will implement robust data protection measures, including obtaining explicit user consent before any data collection, employing secure storage solutions, and ensuring transparency regarding how data is used. Regular audits will be conducted to assess compliance with data privacy regulations, further reinforcing our commitment to user privacy and security.

Stakeholder Evaluation

In our project key stakeholders including Lorraine, our tutor, who acted as the business representative, providing weekly feedback; other course tutors, who offered valuable insights during our oral presentation; and external users, who tested our game prototypes. Feedback from all of these stakeholders was useful in refining the user experience and ensuring that our prototype met their expectations. Guided by the Agile principles that "Our highest priority is to satisfy the customer through early and continuous delivery of valuable software" and "At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly" (Beck et al., 2001), we continuously adapted our design and code based on user input throughout the project.

Initially we focused on replicating traditional board games for our device, but Lorraine's feedback in week three highlighted that we were implementing a solution without understanding the user experience (Six-Sided Dice, 9 August 2024). Although we had made significant progress in identifying the type of device and technologies, we had not considered how to make the interactive board game engaging. This aligned with the Agile value that "Customer collaboration is valued over contract negotiation" (Beck et al., 2001), meaning project requirements should remain flexible.

To address this, we created a paper prototype to test game mechanics and user interactions before coding. Team testing revealed that our traditional Snakes and Ladders game was not novel or exciting. Introducing card-based mechanics, for example allowing players to unexpectedly swap positions on the board, added variety and unpredictability to the gameplay. After discussing our test results with Lorraine, she acknowledged that we had put a lot of thought into the game mechanics, but encouraged us to explore more innovative gameplay options. For example, instead of a dice roll, we could investigate unique methods of obtaining a number output or otherwise augment the experience of playing the game (Six-Sided Dice, 16 August 2024).

We conducted additional user testing to discover new ways to incorporate new game mechanics that could be read by the sensors but not damage the screen. This led us to implement minigames that temporarily altered the game board, requiring players to complete challenges to avoid a penalty or gain an advantage. We originally intended to implement the minigames between player rounds, but after further testing revealed this slowed the pace of the game, we caused them to trigger when a

player lands on a snake's head or a ladder's bottom rung. Feedback from tutors during our oral presentation confirmed these features would align with stakeholder interests.

External user testing revealed that players might prefer dice rolls over card instructions when given a choice, leading to a boring game where they followed each other until a player won. This confirmed that the play was more exciting when unique mechanics, such as changing snake and ladder locations or player positions were implemented. Therefore we decided to eliminate dice rolls and only use card instructions to control player movement.

Our Hardware, Backend and Frontend teams focused on building a small, usable codebase weekly, using version control. We updated Lorraine on our progress through snapshots of our Trello tasks, and conducted playtesting sessions to ensure the new mechanics improved the user experience. This iterative process allowed us to evolve the traditional digital board game into an innovative and engaging experience. Our tutor provided us with positive feedback that we should continue doing what we were doing (Six-Sided Dice, 6 September 2024 and 13 September 2024).

Throughout the project, changes were made and new features were developed iteratively in line with the Agile principles that specify "Working software is the primary measure of progress" and "Deliver working software frequently" (Beck et al., 2001). Early and frequent feedback allowed us to clarify feature requests, address misunderstandings early, and ensure our final implementation met stakeholders' needs more effectively than our initial design.

Conclusion

Our prototype is the first step in addressing a gap in the market for a cost effective and interactive tabletop gaming system that integrates physical and digital elements to play games. By combining tactile gameplay with digital interfaces, we are modernising the traditional board game experience while preserving the social and engaging aspects that players value.

This prototype prioritises a tactile user experience by incorporating Hall effect sensors for piece detection and large buttons for turn advancement, while using an LCD panel for an interactive playing surface, all controlled by a Raspberry Pi. The backend was built using Python, with an internally run FastAPI server and WebSocket protocol facilitating real-time communication. The frontend software uses React.js to ensure a responsive and engaging game experience.

We prioritised ethical considerations and security by implementing safety warnings, ensuring accessibility, and securing hardware and software, whilst adhering to Australian privacy standards. We also conducted thorough risk assessment to mitigate potential challenges.

We used Agile and Scrum principles to enhance collaboration and adaptability across specialised teams, facilitating iterative development. We also successfully integrated feedback from key stakeholders, including business representatives (tutors) and external users, to continuously refine the user experience and adapt our design through iterative testing.

This prototype demonstrates a viable concept that can be refined. An immediate improvement would be to expand the current game library, providing a greater variety of games for users to enjoy. Further research into cost-effective touch screens could enhance player interaction, allowing for more nuanced input to complement the tactile button presses and game piece movement. Additionally, investment in robust hardware manufacturing is also required to improve the durability, security and portability of the device.

We believe that our product has the potential to redefine how people connect and play together in this digital age.

Bibliography

Adam the Fanatic. (2021, January 17). *The Fanatic Compares: Tabletopia vs Tabletop Playground vs Tabletop Simulator*. YouTube. https://www.youtube.com/watch?v=WfmkqZ18RLk

Agile Alliance. (2024, October 3). *Pair Programming: Does It Really Work?* Agile Alliance. https://www.agilealliance.org/glossary/pair-programming/

Aljaber, T. (n.d.). *Iron Triangle Project Management*. Atlassian. Retrieved October 15, 2024, from https://www.atlassian.com/agile/agile-at-scale/agile-iron-triangle

Andersen, G. (2024, February 2). *Ethical Considerations in Game Development: Balancing Fun and Responsibility*. MoldStud. Retrieved October 10, 2024, from https://moldstud.com/articles/p-ethical-considerations-in-game-development-balancing-fun-and-responsibility

Australia: Adoption of the new safety standard for toys for children 3 years old and under. (2023, September 4). TUV Sud. Retrieved October 10, 2024, from https://www.tuvsud.com/en/e-ssentials-newsletter/consumer-products-and-retail-essentials/e-ssentials-7-2023/australia-adoption-of-the-new-safety-standard-for-toys-for-children-3-years-old-and-under

Awad, N. (2024, March 26). *Accessible Color Contrast*. Accessibility.com. Retrieved October 1, 2024, from https://www.accessibility.com/blog/accessible-color-contrast

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowled, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). *Manifesto for Agile Software Development*. Agile Manifesto. https://agilemanifesto.org/

BeyondTablet Technology. (n.d.). Beyond Tablet. Retrieved October 9, 2024, from http://beyondtablet.com/website/technology.html

Bilgili, D. (2024, September 13). *Using dangerouslySetInnerHTML in a React application*. LogRocket Blog. Retrieved October 9, 2024, from https://blog.logrocket.com/using-dangerouslysetinnerhtml-react-application/

Chuon, T. (2022, March 3). *Hiding a Gaming Table in Plain Sight (with build plans!)*. YouTube. Retrieved October 9, 2024, from https://www.youtube.com/watch?v=Lj4hjtHxTu8

Configuration. (n.d.). Raspberry Pi. Retrieved October 10, 2024, from https://www.raspberrypi.com/documentation/computers/configuration.html

Coopee, T. (2018, October 15). *Trouble from Kohner (1965)*. Toy Tales. Retrieved October 9, 2024, from https://toytales.ca/trouble-kohner-1965/

The Daily Scrum is NOT a Status Meeting. (2017, June 5). YouTube. Retrieved October 12, 2024, from https://www.youtube.com/watch?v=i7_RPceEIYE

Designers Institute of New Zealand. (2016). *Massey University College of Creative Arts. Interactive Tabletop Game*. Best Design Awards.

https://bestawards.co.nz/digital/student-digital/massey-university-college-of-creative-arts/interactive-tabletop-game/

Elder, R. (2019, April 22). *A Beginners Guide to Securing A Raspberry Pi*. Blog. Retrieved October 10, 2024, from https://blog.robertelder.org/securing-a-raspberry-pi/

Encrypting Python (.py) Files: A Comprehensive Guide. (2024, August 10). 99RDP. Retrieved October 10, 2024, from https://99rdp.com/encrypting-python-py-files-a-comprehensive-guide/

Fincher, J. (2012, November 20). *The world's most high-tech (and expensive) Scrabble board*. New Atlas. https://newatlas.com/worlds-most-high-tech-and-expensive-scrabble-board/25097/

5 JavaScript Security Best Practices for 2024. (2024, July 31). The New Stack. https://thenewstack.io/5-javascript-security-best-practices-for-2024/?ref=dailydev

Gaearon, D. (2021, July 6). *We need regular CRA maintainer*. Github (Facebook, create-react-app). https://github.com/facebook/create-react-app/discussions/11768

Game Night Fun Elevated. (n.d.). Infinity Game Table. Retrieved October 9, 2024, from https://infinitygametable.com/

Getting Started. (n.d.). Vite. Retrieved October 9, 2024, from https://vitejs.dev/guide/

Gorr, K. (2020, February 3). *A11y - Color Contrast Checker*. Figma. https://www.figma.com/community/plugin/733159460536249875

How to safely use dangerouslySetInnerHTML in React. (2024, March 1). DeadSimpleChat. Retrieved October 9, 2024, from https://dev.to/alakkadshaw/how-to-safely-use-dangerouslysetinnerhtml-in-react-57l1

The Impact of Technology on Board Games: Apps, AI, AR & Beyond. (2024, January 12). Joyful Games. Retrieved October 9, 2024, from https://joyful-games.com/blogs/card-and-board-games-101/impact-of-technology-on-board-games-apps-ai-ar-and-beyond

Increasing security. (n.d.). The Raspberry Pi Guide. Retrieved October 10, 2024, from https://raspberrypi-guide.github.io/other/Improve-raspberry-pi-security

Internet Engineering Task Force. (2011). *RFC 6455 - The WebSocket Protocol*. IETF Datatracker. https://datatracker.ietf.org/doc/html/rfc6455

Jones, J. (n.d.). *11 Best Practices for Secure Web Applications*. LRS Web Solutions. Retrieved October 10, 2024, from https://www.lrswebsolutions.com/Blog/Posts/32/Website-Security/11-Best-Practices-for-Developing-Secure-Web-Applications/blog-post/

Klosowski, T. (2021, November 5). *Why We Love the Raspberry Pi | Reviews by Wirecutter*. The New York Times. Retrieved October 9, 2024, from https://www.nytimes.com/wirecutter/reviews/raspberry-pi/

Landau, P. (2022, March 7). *The 12 Agile Principles: Definitions & How to Use Them*. ProjectManager. Retrieved October 12, 2024, from https://www.projectmanager.com/blog/agile-principles

Lexibook CG1300 Chessman Elite Interactive Electronic Chess Game. (2024). Amazon. https://www.amazon.com.au/dp/B0B93GN9FH?ref_=mr_referred_us_au_au&th=1

Lkmuchic. (2019, January 13). *Beyond Tablet Interactive game Board at CES 2019*. Youtube. https://www.youtube.com/watch?v=LxWqoM3sIW8

Markhvaidze, L., & 0stone0. (2022, April 7). *How to break line in JSX from javascript string*. Stack Overflow. Retrieved October 9, 2024, from https://stackoverflow.com/questions/71779450/how-to-break-line-in-jsx-from-javascript-string

Mason, S. (2019, August 12). *Color Blind*. Figma. https://www.figma.com/community/plugin/733343906244951586/color-blind

Microsoft. (2023, July 11). *Object-Oriented Programming - C#*. Microsoft Learn. Retrieved October 9, 2024, from https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/tutorials/oop

Mingren, W. (2018, May 7). *The Origin of Snakes and Ladders: A Moral Guide of Vice and Virtue*. Ancient Origins. Retrieved October 9, 2024, from https://www.ancient-origins.net/history-ancient-traditions/origin-snakes-and-ladders-moral-guide-vice-and-virtue-0010012

Navigating the Risks of dangerouslySetInnerHTML in React. (2024, May 22). DEV Community. Retrieved October 9, 2024, from https://dev.to/shehzadhussain/navigating-the-risks-of-dangerouslysetinnerhtml-in-react-5c19

Powell, K. (2022, January 19). *Flexbox or Grid*. Youtube. https://www.youtube.com/watch?v=3elGSZSWTbM&t=820s

Raspberry Pi. (n.d.). *Raspberry Pi Documentation*. Raspberry Pi. Retrieved October 9, 2024, from https://www.raspberrypi.com/documentation/computers/os.html

Read the Australian Privacy Principles. (2022, July 25). OAIC. Retrieved October 11, 2024, from https://www.oaic.gov.au/privacy/australian-privacy-principles/read-the-australian-privacy-principles

Robotgeek_official. (2016, June 5). *Arduino RFID Flash Cards (Matching Game) : 4 Steps*. Instructables. Retrieved October 9, 2024, from https://www.instructables.com/Arduino-RFID-Flash-Cards-Matching-Game/

Roussey, B. (2024, May 23). *Inclusive Gaming: How Game Developers Can Make Video Games More Inclusive And Accessible To Gamers With Disabilities*. Accessibility.com. Retrieved October 12, 2024, from https://www.accessibility.com/blog/inclusive-gaming-how-game-developers-can-make-video-games-more-inclusive-and-accessible-to-gamers-with-disabilities

Schwaber, K. (n.d.). *What is Scrum?* Scrum.org. Retrieved October 12, 2024, from https://www.scrum.org/learning-series/what-is-scrum/

Secure Coding: 7 Defensive Programming Techniques to Fortify Your Code - By Sec1. (2024, March 1). Sec1. Retrieved October 10, 2024, from

https://sec1.io/blog/secure-coding-7-defensive-programming-techniques-to-fortify-your-code-by-sec1/

Six-Sided Dice. (2024, August 9). *Sprint Review*. Sprint Notes - 2024/08/09 Week 3. [link redacted]

Six-Sided Dice. (2024, August 16). *Sprint Review*. Sprint Notes - 2024/08/16 Week 4. [link redacted]

Six-Sided Dice. (2024, September 6). *Sprint Review*. Sprint Notes - 2024/09/06 Week 7. [link redacted]

Six-Sided Dice. (2024, September 13). *Sprint Review*. Sprint Notes - 2024/09/13 Week 8. [link redacted]

Taussig, R. (2024). *react-use-websocket: React Hook for WebSocket communication*. GitHub. Retrieved October 9, 2024, from https://github.com/robtaussig/react-use-websocket

Tekunoloji. (2012, January 13). *ePawn Arena - Digital Board Game Tabletop*. YouTube. https://www.youtube.com/watch?v=r8L5rLZuexQ

Three Pillars of Scrum: Understanding Scrum's Core Principles. (n.d.). Atlassian. Retrieved October 12, 2024, from https://www.atlassian.com/agile/project-management/3-pillars-scrum

Tiangolo. (2024). FastAPI. Retrieved October 9, 2024, from https://fastapi.tiangolo.com/

Turing. (n.d.). *What is Functional Programming and Why it is Important to Learn?* Turing. Retrieved October 9, 2024, from https://www.turing.com/kb/introduction-to-functional-programming

Web Content Accessibility Guidelines (WCAG) 2.1. (n.d.). W3C. Retrieved October 9, 2024, from https://www.w3.org/TR/WCAG21/#contrast-minimum:~:text=Success%20Criterion%201.4.3%20Contrast

What Is Data Privacy? (2023, December 19). IBM. Retrieved October 11, 2024, from https://www.ibm.com/topics/data-privacy

What is RESTful API? - RESTful API Explained. (2024). AWS. Retrieved October 9, 2024, from https://aws.amazon.com/what-is/restful-api/

Whitby, B. (2023, December 3). *Modified Pieces and Board Lexibook Harry Potter Interactive Electronic Chess Computer*. YouTube. https://www.youtube.com/watch?v=dfOqzlAxx40

Wiegler, L. (2014, July 11). *RFID Is The New Game In Town*. Atlas RFID. Retrieved October 9, 2024, from https://www.atlasrfidstore.com/rfid-insider/rfid-board-games-casino/

Williams, A. T. (2024, July 31). *5 JavaScript Security Best Practices for 2024*. The New Stack. Retrieved October 9, 2024, from https://thenewstack.io/5-javascript-security-best-practices-for-2024/

YOYN Interactive. (2010, June 10). *YOYN Interactive Board Game Platform*. YouTube. https://www.atlasrfidstore.com/rfid-insider/rfid-board-games-casino/

Attributes

AI Generated Image

Six-Sided Dice logo adapted from an image created with Dali 3 via Copilot Designer using the prompt "A logo of a design company showing a 6 sided dice with the caption "Six Sided Dice Productions"":
https://copilot.microsoft.com/images/create/a-logo-of-a-design-company-showing-a-6-sided-dice-/1-66a2e8ea4f64491c9184c987cd5d10bb

Royalty Free Icons

Clockwise arrow images in setup game grid display adapted from upward arrow icon created by Creative Stall Premium at Flaticon.com:
https://www.flaticon.com/free-icon/upward-arrow_2268143?term=curved+arrow&page=1&position=12&origin=search&related_id=2268143

Goose image in Wild Goose Chase minigame uses goose icon created by Freepik at Flaticon.com:
https://www.flaticon.com/free-icon/goose_2707763?term=goose&page=1&position=21&origin=search&related_id=2707763

Fruit image in Falling Fruit minigame uses fruit icon created by Freepik at Flaticon.com:
https://www.flaticon.com/free-icon/fruits_1625048?term=fruit&page=2&position=32&origin=style&related_id=1625048

Rotten fruit image in Falling Fruit minigame adapted from these icons created by Freepik at Flaticon.com:

- https://www.flaticon.com/free-icon/apple_2847460?term=rotten+fruit&page=1&position=60&origin=search&related_id=2847460
- https://www.flaticon.com/free-icon/rotten_4114810?page=2&position=2&term=rotten+fruit&origin=style-search&related_id=4114810
- https://www.flaticon.com/free-icon/banana_2494112?page=1&position=4&term=banana&origin=style-search&related_id=2494112

Basket icon in Falling Fruit minigame uses icon created by Freepik at Flaticon.com:
https://www.flaticon.com/free-icon/basket_3649105?term=food+basket&page=1&position=31&origin=style&related_id=3649105

Fly icon in Pest Control minigame uses icon created by Pixel Perfect at Flaticon.com:
https://www.flaticon.com/free-icon/fly_1198077?term=fly+insect&page=1&position=4&origin=search&related_id=1198077

Plugins

Colour-blind checks of UI colours conducted using Color Blind Figma plugin at
https://www.figma.com/community/plugin/733343906244951586

Colour contrast checks of UI display conducted using A11y Color Contrast Checker Figma plugin at
https://www.figma.com/community/plugin/733159460536249875

Libraries

FastAPI for backend server and WebSocket server: https://fastapi.tiangolo.com/

Raspberry Pi GPIO Library for enabling GPIO pin control within Python:
https://sourceforge.net/p/raspberry-gpio-python/wiki/Home/

React.js for frontend web app: https://react.dev/

Vite for frontend build tool: https://vite.dev/

react-use-websocket for frontend WebSocket client library:
https://github.com/robtaussig/react-use-websocket

Other

Printed Circuit Boards (PCBs) manufactured by PCBway (https://www.pcbway.com/) from the following designs created by Stuart Moyes and Erik Kelemen:

Sensor Array Blueprint:
https://drive.google.com/file/d/1jAWPEiqSlDl36QcoTvwNKDWIfdV89NLH/view?usp=sharing

Sensor Array Schematic:
https://drive.google.com/file/d/1a9u3XEIrN56e7WY288Ep1AojWU09qrls/view?usp=sharing