

Physically Prototyping Physics for Play: Possibilities and Practicality

Jackson Froehlich
Department of Computer Science
Trinity University
San Antonio, USA
jfroehli@trinity.edu

Britton Horn
Department of Computer Science
Trinity University
San Antonio, USA
bhorn@trinity.edu

Abstract—Game designers commonly use paper prototyping to evaluate educational effectiveness, enjoyment, flow, and usability, while also reducing costs and exploring alternative implementations. However, creating a paper prototype that yields actionable feedback can be challenging due to the wide range of methods available, from low-fidelity sketches to high-fidelity mockups. Most paper prototypes are static and progress discretely, making it difficult to prototype physics-based games effectively, unless focusing on interfaces, narrative, or underlying systems. This paper details the creation of three prototypes of varying fidelity and metaphor for a physics-based educational game on concurrency and parallel programming. Each prototype undergoes playtesting to assess construction methods and their effectiveness in gathering player feedback.

Index Terms—Game Design, Software Prototyping, Computer Science Education

I. INTRODUCTION

While prototyping has long been recognized for its efficiency and effectiveness as an early software development process across fields such as web interfaces, controller design, and game development, debate over which system aspects should involve a prototype and the fidelity in which to create them has remained [1]–[4]. Designers create prototypes to evaluate the look-and-feel, implementation, and role of ideas [1]. Look-and-feel describes the concrete visual and tactile experience such as menus and interfaces, implementation covers the specifics of how a system operates including rules and logic, and role refers to how a system affects users, such as if an intended difficulty range or educational benchmark is met. Integration prototypes explore a combination of all three.

Prototypes can also be divided into at least two categories based on how authentic, realistic, or complete they are compared to the actual system: low- and high-fidelity [5]. Low-fidelity prototypes are cost-effective but require designers to assess what information is valuable to determine the necessary detail. These prototypes allow designers to assess major questions that can drastically change project direction, often in multiple contradictory ways. Conversely, high-fidelity prototypes, typically used in later stages of development, focus on refining detailed aspects such as visual style, assuming that major decisions have been made. Low-fidelity prototypes

often represent layouts, concepts, and design decisions, while high-fidelity prototypes are interactive and allow developers to analyze user interactions and system progression [6].

Due to their facilitation of rapid idea generation and evaluation, many game designers integrate prototyping early into their design process, utilizing digital or paper prototypes, each with their own advantages and disadvantages [7]. Though many advocate for paper prototypes, few have produced general best practices or even determined which systems can effectively be prototyped, leaving major decisions to the designer [8]. Without extensive experience or helpful guidance, designers are left asking themselves: *Is it worth making a paper prototype or should I jump straight into a digital prototype? How detailed should I make my prototype and what can I include?* [4]

Some game genres clearly lend themselves well to paper or physical prototypes because they already resemble a tabletop genre such as card games (e.g. *Magic: the Gathering* [9], *Hearthstone* [10]), board games (e.g. *Clash of Cultures* [11], *Civilization* [12]), or puzzle games (e.g. *Calico* [13], *Tetris* [14]) [15]. Others are more challenging to prototype in a tabletop manner such as physics-based games including *Cut the Rope* [16], *Human Fall Flat* [17], or any of the countless side-scrolling platformers. Paper prototypes for challenging genres often are limited to interfaces, underlying systems such as armor and weapons, or narrative rather than core game mechanics. Furthermore, countless possibilities for materials lead to construction challenges and wasted time that may not have an impact on the prototype’s effectiveness [4].

LEGO prototyping, popular in engineering sciences [18] [19], remains underutilized in game research aside from map layouts [20] [21] despite potential benefits. This method uses LEGO construction kit pieces as the main material allowing novices and experts alike to contribute during design in a dynamic, engaging manner [22]. Some engineers extend piece capabilities with 3D printing and electronics producing sophisticated prototypes to visualize ideas [23] [24].

This paper aims to tease out potentially beneficial and relatively approachable methods of physical prototyping for the core mechanics of a computer science educational game that doesn’t resemble a typical tabletop game genre. We focus on educational games because this work fits in to a larger

undergraduate thesis project developing a digital educational game, though we strongly believe the lessons learned are applicable to a much wider audience. Previous literature on educational and algorithmic thinking games demonstrate their potential benefit to the education space through their ability to engage users while providing an environment where players improve understanding. Unfortunately, creating an *effective* educational game remains a challenge for even experienced game developers [25]. First, finding an appropriate metaphor for the educational concepts is not a trivial task since the educational content rarely lends itself perfectly to a game context [26], and implementing that metaphor in an engaging manner raises further challenges [27]. While numerous definitions and applications of “metaphor” exist in game studies [27], we focus specifically on the visual representation or look-and-feel of a game and how elements are implemented [28].

Prototyping helps identify promising designs early, preventing wasted effort. While designers may attempt to evaluate an entire system in a single prototype, typically prototype complexity is reduced in one of three ways: representing all features but with reduced functionality, representing a limited set of features with full functionality, and reducing both features and functionality such that only a pre-defined path is simulated for the user [5]. Experienced game designers, such as Will Wright, note the value of experimenting with physical prototypes [29], yet physics-based game prototyping remains under-studied.

In this paper, we describe the process of evaluating potential visual metaphors for an educational game to teach concurrency and parallel programming concepts, producing low- and high-fidelity prototypes to determine advantages and disadvantages in regards to construction speed, flexibility, and usability. While no silver bullet exists for all educational game design contexts, this paper attempts to discover effective, under-utilized game prototyping techniques through player feedback and designer insights prior to digital prototyping. Namely, we explore three stereotypical design questions during prototyping regarding visual representation, construction, and fidelity:

- How does look-and-feel impact player understanding?
- What construction techniques facilitate faster and easier redesign?
- What level of fidelity elicits adequate feedback for a physics-based game?

To answer these questions, we prototype an educational game delivering classic concurrency problems such as race conditions, locking, and shared resources as part of an undergraduate research experience, integrating player feedback to inform the prototyping process.

In subsequent sections, we review related work on physical game prototyping, prototype fidelity, and concurrency-related educational games. We then detail the prototyping process for three visual metaphors—abstract, marbles, and trains—discussing construction techniques, challenges faced, feedback, and lessons learned. We close with an examination of construction method trade-offs and conclusions.

II. RELATED WORK

This section details major areas of influence for our work, covering novel ways paper and physical prototypes have been applied to games, variations in fidelity and form factor, and recent examples of educational games on parallel programming. Notably absent from the literature is discourse on prototyping games with significant physics components.

A. Physically Prototyping Games

Schade and Stagge explored the ability of paper prototyping to aid designers creating a serious game for workplace safety, utilizing player input from think-aloud sessions and interviews to assess game characteristics such as mechanics, storylines, rules and navigation [15]. Ultimately, the paper prototype let players be central to the design process and showed that building, structuring, and organizing a paper prototype takes creativity and comes with limitations. Particularly, interface activity such as buttons and click options need to be prominent and consistent, especially at the start of a test run.

Recent research demonstrates the rising popularity of paper and physical prototypes in the educational games space, aiding communication between stakeholders and developers for requirements and expectations [30]. This work attempts to answer, among others, the following questions: *Which design patterns can be abstracted?*, *How can stakeholders communicate that items are associated?*, and *What meanings can be encoded in size, color, or other visual attributes?* Through the use of sketchpads, colored notecards, stickies, and other common office supplies, they demonstrate methods for stakeholders to describe detailed requirements for three video game genres—racing, 2D side-scrolling platformer, and first-person shooter—demonstrating the generality of this process across game types.

Other work approaches prototypes more traditionally through the lens of designers gathering feedback from users about design choices. Specifically, Ollila, Suomela and Holopainen describe the process of creating paper and rapid digital prototypes for location- and context-aware pervasive games to discover improvement ideas and identify problems in a game’s design, identifying areas where paper or software prototypes may be more appropriate [31]. They suggest creating software prototypes immediately if the entire design team is experienced with the requisite skills and much of the gameplay requires sensors or other software-oriented technology. They suggest paper prototyping only a portion of the game such as the core gameplay or interface.

Another example of physical prototyping is Urbanek and G ldenpfennig’s tangible prototyping method for designing audio games using LEGO construction kits [21]. In an audio game, there are no graphics so the focus is on reduced complexity and a strong narrative. However, designing and prototyping these games is challenging because everyone involved in the typical prototyping process requires software programming experience. By stacking LEGO bricks, they create game entities comprised of colors denoting attributes such as enemy, moveable, destructible, and soundsource. The

player moves the “player” stack of bricks around the map (also created with LEGO bricks) to complete objectives. While this method is focused on audio games, we believe it can be extended to additional game genres where the LEGO bricks are used for more than attribute descriptors and map layout.

B. Prototyping Feedback

While variations in prototype fidelity can produce differing user feedback [32]–[34], some research has also looked at the role of variations in user expertise during prototype sessions [35]. This work examines the roles of novices and experts during usability tests with various levels of prototype fidelity. Experts identified more issues with usability than novices, but issues found by novices were deemed more severe than those found by experts. Both computer-based and paper-based low-fidelity UI prototypes lead to almost the same quantity and quality of user critiques, yet users typically prefer computer prototypes [36]. Based on these results, there are three cases where paper prototypes should generally be used: when available prototyping tools do not support the initial design, when some team members lack the software development skills to participate, or when usability tests produce many figures or drawings to be discussed by the design team later. Additionally, paper prototyping should attempt to decrease a participant’s feeling of being observed and reduce facilitator effort to direct the playtesting session.

Specific to educational games for children, previous research investigates user feedback when altering fidelity, audience, and form-factor demonstrating low-fidelity prototypes may be less suitable for children due to negative perceptions of unfinished designs [37]. Altering fidelity in educational game prototypes for children affects user feedback from both children and adults during testing sessions [2]. This research produces two nearly identical digital prototypes differing only in graphics quality and find, perhaps unsurprisingly, no difference in usability between the versions. However, the authors leverage adult users to find major usability issues suggesting that the best playtesters may not be from the target audience. Ultimately, the authors find low-fidelity prototypes efficient in exploring the design space, predictive of user preferences for the final product, and enhance user participation during design.

While low-fidelity prototyping, and paper prototyping in particular, have been well-established as a means for developing traditional two-dimensional interfaces, Maguire investigates its applicability to augmented and virtual reality applications [3]. Through two student workshops on prototyping AR/VR applications, this research gathers participant feedback to analyze how traditional paper prototyping is applied in various participant-generated AR/VR applications. They find AR/VR well-suited to cardboard frames representing a viewing apparatus where objects with labels, figurines, and other suitable content is placed in the real world. This research highlights that a particular context such as a traditional user interface, AR/VR application, or digital game has prototyping techniques that may or may not suit it.

C. Parallel Programming Games

Parallel programming has been a topic of academic discussion for quite some time, but recent work on parallel programming education through games shows a burgeoning interest in the educational and serious games fields. Concurrency-related games have moved from early examples of games with concurrency topics merely present or tangential to the main game objectives [38] to more intentional research on parallel programming education with games [39]–[41].

As supplemental material for a university operating systems course, Marmorstein modifies an existing game called *OpenTTD*—an open-sourced transportation simulation game based on *Transport Tycoon Deluxe*—to provide students with concurrency topics through game elements [38]. *OpenTTD* was not built to support concurrency topics so implementing some mechanics was particularly challenging, such as simulating a race condition, because the AI will prevent trains from navigating unsafe tracks. Overall, this work demonstrates deep player engagement, with some students participating for substantially longer than their assignments required.

Parapple [40] and work by Akimoto and Cheng [41] demonstrate games built specifically to teach parallel programming concepts. Both describe grid-based concurrency educational games where players control units to solve various concurrency problems such as resource sharing, load balancing, synchronization, and non-determinism. While there is some discussion of design principles including the overarching goals of simple gameplay, execution of programs, and commands, there is no discussion on the design process in general.

Conversely, research on *Parallel* explicitly describes the process for choosing [28] and implementing [39] a visual metaphor including the extent to which a player codes, digital prototype testing, and educational efficacy. The creators of *Parallel* define an axis along which programming can be abstracted including: pure programming, abstract programming, and environment as program. They generate an initial set of game ideas along this axis and have students implement prototypes (either paper or digital) to assess the design. Similar to Akimoto and Cheng’s work, the creators eventually settle on the environment as a program with no player coding and processes represented as tracks. The researchers then assess multiple visual metaphors where the tracks are implemented as roads, railways, or abstractly. After a small user test (2 users), they find the abstract visual metaphor to be the most promising, though it is unclear how much the specifics of each prototype’s implementation methods contribute to player feedback. Specifically, the work does not go into detail about the prototyping construction process, materials used for each prototype, or guidance for game designers in the field leaving others to wonder if different conclusions would have been reached with different prototyping methods.

III. DEVELOPMENT PROCESS AND GOALS

This section outlines the initial development of a CS undergraduate research project, undertaken by a student with some experience in digital game development and paper prototyping.

Following direction from Schell’s *The Art of Game Design* [8], we create prototypes to make crucial design decisions early and allow for course corrections. To investigate the impact of visual representation, construction techniques, and fidelity on playtesting feedback, we develop three prototypes of varying fidelity for an educational game demonstrating synchronization concepts. Key decisions focused on the representation of threads, collisions, locks, and waiting.

We decided on three visual metaphors following previous work on player understanding of educational concepts across differing contexts [42]. Regarding look-and-feel, we chose specific metaphors—abstract, marbles, and trains—following previous work on concurrency education games [28] and popular game genres easily translatable to a concurrency context [43], [44]. We assign fidelity to each metaphor with the abstract being the lowest fidelity, followed by marbles, and trains as the highest utilizing LEGO prototyping techniques.

While we strove to keep each prototype in its own development bubble, the abbreviated timeline of an undergraduate thesis meant some concurrent production had to occur. We don’t view this as an issue with understanding the impact of construction on player engagement or efficacy, because most of the cross-prototype lessons learned were about playtesting instructions, terminology, and player feedback format. In fact, some previous work indicates that creating multiple prototypes at the same time is ideal to increase designer creativity and prototype effectiveness [45]. We separate feedback about the metaphor from quality of construction as much as possible, but there may be situations where it is inevitably linked.

We gathered feedback from 9 undergraduate CS students (4 female, 4 male, 1 non-binary) ranging from Sophomore to Senior. 5 students tested every prototype, while the remainder tested one prototype. Out of 9 people, 3 described themselves as “comfortable” with parallel programming concepts, though all 9 had seen parallel programming in their first year introductory CS courses. After each playtesting session, we asked open-ended questions such as: *How much control do you feel you had? Were there any prototype construction issues? Was it clear why [trains/marbles/tokens] aren’t allowed to collide?* We then looked for common feedback between testers that could help us improve the prototype or gain insight into existing design decisions. Based on the feedback and the rapidness it could be incorporated, we then made adjustments to each prototype before presenting it to the next tester.

IV. ABSTRACT PROTOTYPE

Inspired by previous design decisions of a parallel programming educational game [39], we created a prototype using an abstract visual representation. This paper-based prototype resembles a traditional board game and represents the lowest fidelity level of our designs. In this prototype, tokens (i.e., dice) move through discrete steps along a predetermined series of board spaces, as shown in Figure 1.

Our primary consideration was how to represent threads or processes to players effectively. We decided on tokens to progress through time in discrete steps, facilitating clear



Fig. 1. Initial abstract prototype with dice as running programs and discrete board spaces. Players use notecards to denote where to place *lock* and *unlock* actions on the program sequence. Overlapping paths signify a shared resource.

turns and decision-making akin to popular tabletop games. We viewed the player controlling threads reflective of how a competent programmer would do the same—careful forethought of important interactions prior to implementing a solution and running the program. Playing the game would essentially consist of laying out a series of locks and then hitting *Run* to watch the threads move and interact.

Regarding thread collisions, we established a rule that only one token may occupy a space at any time interval, with players losing if this rule is violated. We believed a clear rule would guide the player to understand that a collision is something they should simply avoid at all costs. Thus, we needed to afford them preventative measures on the board (i.e. locks). Deciding on lock placement and resolution of race conditions posed challenges. We explored different approaches such as resolving race conditions before, during, or after a player’s move and playtested to determine clarity and effectiveness. For waiting threads, we considered whether lock freeing and acquisition occur simultaneously or with a delay, testing both to gauge player comprehension.

A. Construction Methods

As seen in Figure 1, our tabletop-inspired prototype utilized dice for tokens, paper game spaces with paperclips to visualize thread paths and attach lock/unlock actions, and a “key” symbol indicating lock availability. Facilitators adjusted the board to demonstrate various collision scenarios such as *Thread 1* always tries to obtain the lock before *Thread 2* or one where they will both necessarily try at the same time.

B. Player Feedback

Players found the abstract nature confusing, especially discerning threads and their states. Even with simple setups, the order of operations for obtaining locks, unlocking, and moving forward were opaque at best. As seen in Figure 1, each thread follows one line and could collide at the single intersection point. Both thread paths require an *unlock* action placed, but players consistently wanted to put the unlock on the intersection itself since they would only have to place it once. Confusion persisted among players with advanced computer science backgrounds stating that it simply didn’t

make sense to them. Feedback highlighted the need for simplifying lock/unlock actions, clarifying operations order, limiting facilitator involvement, and improving thread distinctiveness.

V. MARBLE PROTOTYPE

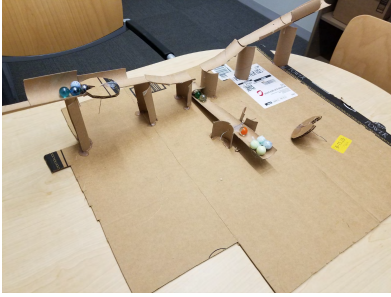


Fig. 2. Completed marble prototype for initial level. This figure shows the weight-biased bridge triggering from two marbles occupying the segment at the same time.



Fig. 3. Selection of marble intersection prototypes during early construction. We found these to be too time-sensitive and produced inconsistent results.

The marble prototype attempts to embody thread collisions in a natural and self-explanatory context, relying on physical elements rather than explicit instructions. We produced this prototype with a higher fidelity than the abstract prototype as it looks significantly more “complete,” but not as polished as the train prototype in the following section.

We designed custom track segments to simulate race conditions and collisions, where marbles falling off the track results in losing. This required iteration to land on a consistent and workable solution as seen in Figure 3, but it is very clear for players and takes less commentary from the facilitator. The player’s goal resembles that of the abstract prototype—allow the game pieces to move to the end of the level without colliding. Due to the similarities, the rules given by the facilitator mirrored those of the abstract prototype with minor updates.

A. Construction Methods

Unlike the abstract prototype, we wanted the marble to prototype to feel *real* so the player could see strategy consequences without much involvement from the facilitator. We primarily used cardboard tubes for the construction of tracks and superglue, paperclips, and tape as fasteners.

As the primary design goal was to make a collision as obvious and contextually relevant as possible, we needed to create track segments where calamity ensued if two marbles

attempted to occupy it at the same time, yet a lone marble would always traverse successfully. Since the speed of marbles was highly variable, we discovered that constructs requiring exact timing (e.g. two marbles bumping into each other at an intersection with enough force to knock each other off) were sufficiently inconsistent, and, after testing numerous ideas, we settled on creating a weight-biased slanted bridge that flips the wrong direction with two marbles (see Figure 2). The time delay from when a marble first entered this segment to when it completed a traversal down the slope granted ample time for a successive marble to trigger the weight-bias without as much concern for marble speed variability.

Locks were implemented with a cardboard wheel attached to a paperclip inserted through the track at specific hole-punched locations. When the player places a lock on the track and two marbles attempt to pass at the same time, the facilitator chooses one at random to let through. The player then needs to indicate where the lock is freed and the facilitator will only let another marble through when the first reaches that point. If multiple marbles are waiting on the same lock, they simply queue up as expected. After some minor instruction, players are able to perform this action on their own and the facilitator can focus on answering player questions.

B. Player Feedback

Feedback from playtesting sessions revealed improved player understanding and engagement, particularly regarding the concept of locks. However, lock construction inconsistencies occasionally required manual intervention creating player confusion and frustration, as players understood locks in a theoretical sense, but not necessarily how they physically interacted with the prototype. Some confusion persisted regarding how locks communicated between themselves to decide which marble was given access to the shared track. The metaphor of a single “key” opening either lock remained easily understandable for players.

Once players grasped the mechanics, they appreciated the ability to experiment with different scenarios independently and enjoyed the fast-paced gameplay. The increased complexity of construction meant the track was prone to mechanical errors not seen in the abstract prototype such as the balanced bridge being biased incorrectly or marbles jumping off the track. A number of respondents found this aspect frustrating and stated that it detracted from their experience.

VI. TRAIN PROTOTYPE

We implement our final prototype with trains and the highest fidelity. Leveraging LEGO prototyping methods, we utilize an existing train construction kit. Similar to the marble prototype, shared resources are portions of track where multiple trains attempt to occupy simultaneously, and collisions between trains signify failure making the objective instantly comprehensible. Additionally, plastic passengers from the train set serve as additional shared resources for more complex levels.

Locks are implemented as one-armed bandit gates that only raise when a lock is successfully obtained. The gate allows

one train through at a time, and the player places a colored track segment that corresponds to a colored gate representing where an *unlock* action occurs.

A. Construction Methods

Distinct from previous prototypes, this version utilizes an existing train set supplemented by battery-powered trains and 3D-printed custom track segments. Custom pieces include the gates, intersections not included in the original set, and additional parts to make connecting the track easier.

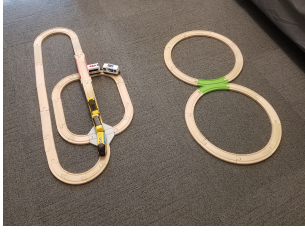


Fig. 4. Two versions of the same layout with one version using a custom 3D printed piece to simplify the construction. The simplified version requires neither elevated track segments nor manual train diversion as would be required if using stock segments.

We produced additional intersection types because the set comes with just two: an *X* intersection and a *Y* intersection. The *Y* portion did not allow us to choose which direction the train would use, though there are pre-existing train sets that do, so we created a similar type of intersection where we could force the train in a particular direction allowing planar levels instead of requiring bridges. An example of two functionally equivalent levels can be seen in Figure 4. It is worth noting that, even without printing any custom pieces, we are able to construct functionally equivalent levels to the abstract and marble prototypes’ initial layouts as long as we use paper to indicate *lock* and *unlock* locations.

Printing of custom pieces was done on a university 3D printer requiring between one and five hours of print time depending on the piece. We produced all custom pieces with existing 3D models obtained for free online¹ and only modified the initial model in *Blender* when absolutely necessary. Little technical expertise with either 3D printing or 3D modeling was required by the design team. Since our train construction kit contains manufacturer-standardized pieces, compatible 3D-printable pieces, battery-powered trains, and extension kits are easy to find. Low 3D printer resolution caused some pieces not to fit precisely, but minor sanding or a bit of elbow grease was enough to make problematic segments connect.

B. Player Feedback

Feedback from playtesting sessions indicates reduced mechanical issues compared to previous prototypes, enhancing the overall gameplay experience. Players praised the prototype’s completeness and interactivity, expressing a preference for its straightforwardness. The ability to manipulate tracks



Fig. 5. Completed train prototype for the *Dining Philosopher’s Problem*. Both common track segments and passengers represent the two shared resources for this problem. Custom parts include both 3D-printed track sections, gates, and cardboard add-ons.

and trains independently increased player engagement, with many respondents finding the prototype enjoyable. One person who played all three prototypes referred to the train prototype and said “this one is good.”

3D-printing allows pieces denoting locations to be printed in different colors creating a clean visual way to show the links with matching colors. While players appreciated the clarity and visual representation of lock and unlock locations, some expressed a desire for more automation in gameplay mechanics, suggesting improvements for future iterations. Interestingly, one respondent enjoyed the train prototype overall because it was “cool and straightforward,” yet thought the marble prototype was more fun because of the speed and quantity of marbles going down the track. They thought that faster or more trains would improve the experience.

VII. DISCUSSION

Time and Resource Investment. Obviously, differing approaches to physical prototyping will incur advantages and disadvantages in time, flexibility, quality and resource investment. The best practices we describe in this section attempt to separate the visual metaphor used from the construction techniques involved in each prototype. We posit that reversing the fidelity of each prototype (i.e. abstract as the highest game-like fidelity and trains as the lowest) would provide similar engagement from players indicating it is the *completeness* of the prototype to which they are positively responding rather than the visual metaphor itself. In fact, work by [28] explicitly shows that implementing a high quality abstract visual metaphor is effective in this context.

While a low fidelity prototype leads to fast prototype creation and quick prototype iteration, it also poses challenges for players. Much of the thread progression and decision points in the abstract prototype were made by the facilitator during play sessions giving a “man behind the curtain” feeling causing some player confusion. This should be expected since a quick prototype naturally requires more intervention from the facilitator [8]. In future iterations, we would attempt to

¹<https://www.thingiverse.com/>

write complete rules and have the players themselves move thread tokens and settle collisions with dice rolls, coin flips, or other random outcome generator.

The marble prototype, despite its cost-effectiveness and clarity, required significant construction time (about 40 hours vs. 4 for the abstract prototype and 8 for trains including initial 3D prints) due to its reliance on gravity-driven interactions. It was also the least adaptable to player feedback since any modifications required significant customization. In contrast, the train prototype, utilizing materials from an existing train set, mitigates some of these limitations, allows for more varied track designs and quicker adjustments, and retains visual clarity for players. We were able to use this extra bandwidth to create a train version of the *Dining Philosophers Problem* (Figure 5), a canonical example in concurrency education demonstrating a potential deadlock scenario. However, crafting detailed prototypes upfront risks wasted time if initial designs prove faulty. For example, if we found trains to be a poor visual metaphor, swapping representations would involve purchasing an entirely new construction kit.

Scalability and Adaptability. Our process showed that scalability of physical prototypes depends heavily on the materials used. Intricate, customized construction methods are less adaptable to player feedback and struggle with additional iterations. For example, the marble prototype’s cardboard construction hindered scalability, whereas the train prototype’s pre-existing components facilitated easier adaptation. The LEGO method’s compatibility and 3D printing also allow for quick adjustments, reducing overhead. With the least specialized materials, the abstract prototype was the most adaptable.

Designers should seek cheap materials that are durable enough to stand up to multiple testing sessions, particularly materials that can be moved, customized, extended, and augmented easily. We found a pre-existing train set and 3D printing ideal for this because the set was built from high quality wood that could take heavy use from children, and it could be adapted easily with 3D printed parts since connections fit a common standard used by many train set manufacturers. Without a specific visual representation in mind, we suggest designers start with a standard LEGO kit.

Conversely, we found the custom-made marble track elements challenging to adapt and displayed significant wear-and-tear. Following the LEGO prototyping method, utilizing a marble track and only crafting required custom pieces (i.e. locks, balanced bridges, and other shared tracks) could reduce customization and increase durability.

Insights for Ease-of-Use. After multiple prototype versions and playtesting sessions, feedback showed that we created a much clearer representation of concurrency constructs in the marble and train prototypes—not due to metaphor, but rather the clarity of construction. While players responded to the abstract prototype more negatively in general due to its simplicity and lack of engagement, we gathered valuable feedback about lock implementation and educational material. Particularly, players did not have a sufficient answer to “*Why do I care that things collide?*” This is in direct contrast

to the marble and train prototypes where the game context conveys an answer to the question in a concrete manner. We strongly believe each visual metaphor could be successful with a sufficiently designed high-fidelity prototype using the LEGO methodology. While players had a number of confusions about the abstract metaphor, perhaps the discrete nature of gameplay and rudimentary materials contributed the most to that lack of understanding. Using the LEGO methodology and continuous gameplay similar to that in *Parallel*, our abstract prototype could be significantly improved. We found hands-on, self-directed play enhanced player enjoyment and increased their desire to give quality feedback. Additionally, we posit that player engagement during testing directly correlates to the quality of feedback obtained. Player engagement has been a major topic of games research, but little has been done in the space of preliminary prototypes. As such, we highly encourage more study on the connections between fidelity, player engagement, and quality of feedback.

VIII. CONCLUSION

In this paper, we have outlined our process of designing and implementing physical prototypes for a parallel computing-focused educational game, utilizing three different visual metaphors: abstract, marbles, and trains. Each prototype varied in fidelity and construction method, providing valuable insights into their effectiveness in engaging players and conveying complex educational concepts.

The abstract prototype, being low-fidelity and simple to construct, allowed us to quickly iterate on design ideas and understand the level of detail required in player instructions. However, it lacked engagement and clarity compared to the higher-fidelity prototypes. The marble prototype, while still relatively low-cost, offered a more tangible representation of thread interactions through custom-built track elements. This prototype aimed to demonstrate physical interactions rather than relying solely on players’ imagination, leading to deeper player engagement and increased feedback specificity. The high-fidelity train prototype, utilizing the LEGO prototyping method and pre-existing train set components, provided a comprehensive and customizable experience for players. This prototype facilitated deeper engagement and allowed for more complex puzzle designs to be tested.

Through our prototyping process, we found trade-offs between adaptability, engagement, and effort. Low-fidelity prototypes were quick to build but lacked engagement, while high-fidelity prototypes facilitated deeper player engagement but required more time and resources to construct. We recommend that game designers incorporate physical prototyping into their development process to assess visual representations, game mechanics, and player engagement. Utilizing pre-existing materials and quick, custom production techniques, such as 3D printing, can streamline the prototyping process. Additionally, we plan to implement digital versions of our prototypes to further analyze player engagement and educational quality. Overall, physical prototyping serves as a valuable tool, allowing designers to test ideas, gather feedback, and iterate on

designs in a tangible and engaging manner. LEGO prototyping enhances this process by adding game-like fidelity while retaining rapid redesign and customization.

REFERENCES

- [1] S. Houde and C. Hill, "What do prototypes prototype?," in *Handbook of human-computer interaction*, pp. 367–381, Elsevier, 1997.
- [2] B. Köhler, J. Haladjian, B. Simeonova, and D. Ismailović, "Feedback in low vs. high fidelity visuals for game prototypes," in *2012 Second International Workshop on Games and Software Engineering: Realizing User Engagement with Game Engineering Techniques (GAS)*, pp. 42–47, IEEE, 2012.
- [3] M. Maguire, "An exploration of low-fidelity prototyping methods for augmented and virtual reality," in *Design, User Experience, and Usability. Design for Contemporary Interactive Environments: 9th International Conference, DUXU 2020, Held as Part of the 22nd HCI International Conference, HCII 2020, Copenhagen, Denmark, July 19–24, 2020, Proceedings, Part II* 22, pp. 470–481, Springer, 2020.
- [4] M. Walker, L. Takayama, and J. A. Landay, "High-fidelity or low-fidelity, paper or computer? choosing attributes when testing web prototypes," in *Proceedings of the human factors and ergonomics society annual meeting*, vol. 46, pp. 661–665, Sage Publications Sage CA: Los Angeles, CA, 2002.
- [5] R. A. Virzi, "What can you learn from a low-fidelity prototype?," in *Proceedings of the Human Factors Society Annual Meeting*, vol. 33, pp. 224–228, SAGE Publications Sage CA: Los Angeles, CA, 1989.
- [6] J. Rudd, K. Stern, and S. Isensee, "Low vs. high-fidelity prototyping debate," *interactions*, vol. 3, no. 1, pp. 76–85, 1996.
- [7] H. Korhonen, J. Paavilainen, and H. Saarenpää, "Expert review method in game evaluations: comparison of two playability heuristic sets," in *Proceedings of the 13th international MindTrek conference: Everyday life in the ubiquitous era*, pp. 74–81, 2009.
- [8] J. Schell, *The art of game design : a book of lenses*. Amsterdam ; Boston : Elsevier/Morgan Kaufmann, 2008.
- [9] *Magic The Gathering*. Wizards of the Coast, 1993.
- [10] *Hearthstone*. Blizzard Entertainment, 2014.
- [11] *Clash of Cultures*. WizKids, 2021.
- [12] S. Meier, *Civilization*. MicroProse, 1991.
- [13] *Calico*. Flatout Games, 2020.
- [14] A. Pajitnov, *Tetris*. 1985.
- [15] C. Schade and A. Stagge, "Paper prototyping as a method for the evaluation of serious game concepts," in *European Conference on Games Based Learning*, pp. 515–XIX, Academic Conferences International Limited, 2020.
- [16] *Cut the Rope*. ZeptoLab, 2010.
- [17] *Human Fall Flat*. No Brakes Games, 2016.
- [18] S. S. Isa and A. Liem, "Exploring the role of physical prototypes during co-creation activities at lego company using case study validation," *CoDesign*, vol. 17, no. 3, pp. 330–354, 2021.
- [19] R. Gower, A. Heydtmann, and H. Petersen, "Lego: Automated model construction," 1998.
- [20] J. Legardeur, S. Minel, and E. Savoie, "A pedagogical game based on lego bricks for collaborative design practices analysis," in *Complex Systems Concurrent Engineering: Collaboration, Technology Innovation and Sustainability*, pp. 487–494, Springer, 2007.
- [21] M. Urbanek and F. Güldenpfennig, "Tangible audio game development kit: prototyping audio games with a tangible editor," in *Proceedings of the Eleventh International Conference on Tangible, Embedded, and Embodied Interaction*, pp. 473–479, 2017.
- [22] D. Boa, D. Mathias, B. Hicks, *et al.*, "Evolving lego: Prototyping requirements for a customizable construction kit," in *DS 87-4 Proceedings of the 21st International Conference on Engineering Design (ICED 17) Vol 4: Design Methods and Tools, Vancouver, Canada, 21-25.08. 2017*, pp. 397–306, 2017.
- [23] S. Follmer, D. Carr, E. Lovell, and H. Ishii, "Copycad: remixing physical objects with copy and paste from the real world," in *Adjunct proceedings of the 23rd annual ACM symposium on User interface software and technology*, pp. 381–382, 2010.
- [24] S. Mueller, T. Mohr, K. Guenther, J. Frohnhofen, and P. Baudisch, "fabrickation: fast 3d printing of functional objects by integrating construction kit building blocks," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 3827–3834, 2014.
- [25] A. Chaffin and T. Barnes, "Lessons from a course on serious games research and prototyping," in *Proceedings of the Fifth International Conference on the Foundations of Digital Games*, pp. 32–39, 2010.
- [26] J. Melero, P. Santos, D. Hernández-Leo, and J. Blat, "Puzzle-based games as a metaphor for designing in situ learning activities," in *7th European Conference on Games Based Learning, ECGBL 2013*, vol. 2, pp. 674–682, 2013.
- [27] S. M. Möring, *Games and Metaphor-A critical analysis of the metaphor discourse in game studies*. IT-Universitetet i København, 2014.
- [28] S. Ontañón, J. Zhu, B. K. Smith, B. Char, E. Freed, A. Furqan, M. Howard, A. Nguyen, J. Patterson, and J. Valls-Vargas, "Designing visual metaphors for an educational game for parallel programming," in *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pp. 2818–2824, 2017.
- [29] W. Wright, "Triangulation: A schizophrenic approach to game design," 3 2004. Speech given at the Game Developer's Conference 2004, (San Jose, CA).
- [30] D. Callele, "Physualization: Going beyond paper prototyping," in *2010 Fifth International Workshop on Requirements Engineering Visualization*, pp. 35–39, IEEE, 2010.
- [31] E. M. Ollila, R. Suomela, and J. Holopainen, "Using prototypes in early pervasive game development," *Computers in Entertainment (CIE)*, vol. 6, no. 2, pp. 1–17, 2008.
- [32] M. Deininger, S. R. Daly, J. C. Lee, C. M. Seifert, and K. H. Sienko, "Prototyping for context: exploring stakeholder feedback based on prototype type, stakeholder group and question type," *Research in engineering design*, vol. 30, pp. 453–471, 2019.
- [33] B. Dhillon, P. Banach, R. Kocielnik, J. P. Empananza, I. Politis, A. Raczeska, and P. Markopoulos, "Visual fidelity of video prototypes and user feedback: a case study," in *Proceedings of HCI 2011 The 25th BCS Conference on Human Computer Interaction* 25, pp. 139–144, 2011.
- [34] L. S. Jensen, L. Nissen, N. Bilde, A. G. Özkil, *et al.*, "Prototyping in mechatronic product development: How prototype fidelity levels affect user design input," in *DS 92: Proceedings of the DESIGN 2018 15th International Design Conference*, pp. 1173–1184, 2018.
- [35] J. Sauer, K. Seibel, and B. Rüttinger, "The influence of user expertise and prototype fidelity in usability tests," *Applied ergonomics*, vol. 41, no. 1, pp. 130–140, 2010.
- [36] R. Sefelin, M. Tscheligi, and V. Giller, "Paper prototyping-what is it good for? a comparison of paper-and computer-based low-fidelity prototyping," in *CHI'03 extended abstracts on Human factors in computing systems*, pp. 778–779, 2003.
- [37] G. Sim and B. Cassidy, "Investigating the fidelity effect when evaluating game prototypes with children," in *27th International BCS Human Computer Interaction Conference (HCI 2013)* 27, pp. 1–6, 2013.
- [38] R. Marmorstein, "Teaching semaphores using... semaphores," *Journal of Computing Sciences in Colleges*, vol. 30, no. 3, pp. 117–125, 2015. Publisher: Consortium for Computing Sciences in Colleges.
- [39] J. Zhu, K. Alderfer, A. Furqan, J. Nebolsky, B. Char, B. Smith, J. Villareale, and S. Ontañón, "Programming in game space: how to represent parallel programming concepts in an educational game," in *Proceedings of the 14th International Conference on the Foundations of Digital Games*, pp. 1–10, 2019.
- [40] E. Buzek and M. Kruliš, "An entertaining approach to parallel programming education," in *2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pp. 340–346, IEEE, 2018.
- [41] N. Akimoto and J. de Cheng, "An educational game for teaching and learning concurrency," in *Proceedings of the 1st International Conference on Knowledge Economy and Development of Science and Technology (KEST'03), Honjo, Japan*, pp. 34–39, 2003.
- [42] B. Horn, C. Clark, O. Strom, H. Chao, A. J. Stahl, C. Harteveld, and G. Smith, "Design insights into the creation and evaluation of a computer science educational game," in *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, pp. 576–581, ACM, 2016.
- [43] *Marble World*. Vector Interactive, 2021.
- [44] W. Group, *Tracks*. Excalibur Games, 2019.
- [45] S. P. Dow, A. Glassco, J. Kass, M. Schwarz, D. L. Schwartz, and S. R. Klemmer, "Parallel prototyping leads to better design results, more divergence, and increased self-efficacy," *ACM Transactions on Computer-Human Interaction*, vol. 17, Dec. 2010.