ARcade Brick Breaker: Use of Real-World Objects for Direct Manipulation and Interaction with a Virtual Object

Alex Tanasescu 30041538 (BSc Comp Sci
Lougheed Wan 30063200 (BSc Comp Sci)
John Leanard Valencia 30041150 (BSc Comp Sci)
Akashdeep Singh 30128444 (BSc Comp Sci)
Brandon Nguyen 30009135 (BSc Comp Sci)

Figure 1: The system allows real-time manipulation and interaction with a virtual object using any real-world physical objects in an Augmented Reality Brick Breaker game.

ABSTRACT

This paper introduces ARcade Brick Breaker, an augmented reality brick breaker game that introduces a new approach for tangible user interface and virtual interaction using real-world physical objects. Unlike traditional augmented reality games which require the use of separate movement and manipulation tools, we propose the use of readily available physical objects as the grounds for our controllers. This allows our game to be much more accessible and have a much larger ease of use for a broader spectrum of users. Our system will be able to track and locate real life objects to use within our virtual game, with real time physics generation and object tracking. Our proposed plan will use a regular phone camera to display this augmented reality to our users. We aim to have our system be easy to use and accessible for everyone.

CCS CONCEPTS

• **Human-centered computing** → **Mixed / augmented reality**;

KEYWORDS

augmented reality; mixed reality; physical controllers; tangible interface; tangible interaction

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Tangible User Interfaces (TUIs) have been gaining momentum in the realm of Augmented and Mixed Reality (AR/MR) technology. These interfaces allow users to interact with digital content in a more natural and intuitive way by using physical objects as a means

of control. TUIs use physical objects as controllers or in-game elements, providing a more natural and tangible experience compared to traditional input methods. TUIs offer many advantages, such as increased immersion and engagement, and are particularly useful in scenarios where traditional input methods, such as keyboards or mice, are not practical or desirable.

1.1 Whats the problem?

With recent advancements in augmented reality software and hardware, there have been huge improvements for the interactions between the virtual and the real world. With the use of third-party robots and advanced AR interfaces, we can now interact with the virtual world seamlessly with the use of physical real world objects. However, in existing AR tools there does not exist an AR interface that allows users to modify general accessible objects and use them to interact with the virtual world without the use of robotics or advanced tangible user interfaces. Our purpose for this paper is to make virtual and real world interactions accessible to the everyday user without the use of robotics or expensive controllers.

1.2 How did we solve the problem?

In this paper, we introduce Arcade Brick Breaker, an augmented reality brick breaker that uses real-world objects for real-time interaction and manipulation with a virtual object. The AR game is suitable in almost any physical environment and with a variety of possible objects. Unlike the traditional augmented reality games which requires use of specialized controllers or robots for tangible experience, our system augments and tracks regular physical objects to be used in the game. By using object recognition and tracking, real-world objects are augmented to be the controller and target of the game. When the user plays the game, user can choose any real-world object to be the paddle of the game and it allows the user to directly manipulate and interact with the virtual ball in order to hit the target. By moving an actual object, the system can provide real-time feedback and enhanced immersion while playing the game. By combining use of physical objects with AR technology, we aim to demonstrate the potential of this approach in creating immersive and engaging gaming experiences that blur the line between the physical and digital worlds.

1.3 How we did it?

We used Unity and C# to create the two main parts of our project initially. These are the:

- Physical object tracking
- Brick breaker game simulation

Brick breaker game : We built a 2D version of the classic game where the user controls a paddle using the arrow keys. The player then tries to break all the bricks in the game by bouncing the ball off said paddle. This was accomplished using the Unity physics engine, colliders and drawn sprites.
Object tracking : The user can move around any solid object and we would be able to track where the object is in a smooth and seamless like motion without any lag. With the use QR codes the objects are tracked in real time and any movement of the physical object with the QR code instantly results in a movement of the produced synthetic virtual object.
While it is true that objects without QR codes may also be tracked, the challenges lies in implementing a sophisticated object detection that not only detects the object but also the overall system runs in real time. Satisfying both is a heavy task to apply. Alternatively, object could also be tracked via color, however the issues lies in instance that the object color is not unique or has similarities with the space around it.
Thus, using QR codes we are able to track objects in simple and robust fashion, and still accomplish our project requirements and goals.

The two parts have been merged into one. After this checkpoint, we implemented the collision detection between the virtual and physical objects. The virtual ball needs to bounce every time it hits the slider/paddle or the target object. Initially we made the collision work on a white background, so that it was easier to debug and test the functionality in general. But later, we extended the rest of the functionality to work with any background, since the user may play the game anywhere.

This is the main structure and design of our project, but however further features can be added and could be potential extension of the project:

- adding a visual effect the moment the virtual object hits the target object. For instance, if the target object is a laptop with a screen, if the virtual ball hits the screen one possible visual effect is to add cracks to the screen virtually and make them appear almost real as they are indistinguishable with the real world. The game would terminate the moment the screen collapses (which again, is a virtual visual effect).
  Thus, for specific target objects a unique visual effect may be done. A visual effect is not limited to removing "bricks", so chipping away parts of the real objects and replacing with background, or adding cracks but anything appealing and cool effects such as painting colors, fading, morphing, folding etc...
  Obviously, such a feature requires sophisticated real time implementation and may involves machine learning to some extent to add effects or generate portions of synthetic images (such as background) that resemble as best as possible with the surroundings.
- adding different shapes or styles for the virtual ball or paddle. Visual effects may not only apply to the target object but also to virtual ball and paddle as well. For example, a paddle may act as a torch or ray effect, with shadows.
  Such effects can potentially make a simple game interesting, fun and more immersive to the user, but obviously may require costly and cpu-intensive tasks such as when it comes to ray-tracing.

1.4 Contributions

Our project contributes in:

- ARcade Brick Breaker, an augmented brick breaker game that works any suitable physical environment

- Real-time feedback to improve interaction between the user and the physical objects
- Design of physical world's exploitation to create an immersive experience between the augmented and real world
- Design of interface for application in evaluation and cognitive analysis of users

2 RELATED WORK

2.1 Uni-directional and Bi-directional interactions

Interactions between the physical and virtual world can occur in either directions. Reality-Sketch [11] discusses the interactions of the virtual world affected by the physical world, where using physical objects of any shape, such as a small cap, specific effects or transformation are applied on the virtual world. A few examples are, using a small cap to act as as a slider and angle maneuver to scale and rotate a synthetic virtually drawn dinosaur and generating graph by tracking a physically moving object.

Our project involves this type of interaction where the slider will be used as a physical object to affect the movement of virtual ball flying around.

While Reality-Sketch is thus a uni-directional type of interaction, Sketched-Reality [7] allows interactions in either direction. The Bi-directional interactions between the two worlds allows to achieve a more seamless and immersive experience to the user. For example, moving one robot would generate a synthetic virtual rope that "simulates" dragging of other robots. Thus, the robots, which are part of the physical world, are used to affect changes on the virtual world, which in return also affects the position and movement of the same robots. This type of Bi-directional interaction is a potential for a better experience of the user.

While a bi-directional interaction is more powerful, it also draws more challenges as not all physical objects can react to interactions form the virtual worlds. In the case of Sketched-Reality tiny robots were used for this practical aspect. They have have the capability to move themselves, which is not true for all objects that the user may find around his/her environment.

Thus, for the scope of our brick-breaker project game we focused to begin with uni-directional interaction, where the virtual world is affected by interactions in the physical world, as we can achieve interactions using any physical object that the user may find. This fits best with the requirements that we provided initially in the **introduction** section, but however we considered bi-directional interactions still an important topic as our project may be further possibly in the future and later stages.

2.2 Object Manipulation

A virtual object can be moved, rotated or scaled based on physical interactions. However, some sort of "controller" is needed for such an application. Finding an appropriate object for such complex actions may require a sophisticated or well designed physical object, as traditional controller may not be highly suitable and comfortable for users. For example, cube or spherical shapes may be well suited for such applications.

One of the interaction techniques investigated in the paper "Evaluating Object Manipulation Interaction Techniques in Mixed Reality"[3] is the Tangible Cube, which is a tangible user interface for manipulating virtual objects. The Tangible Cube consists of a physical cube with markers on each face, which are tracked in real-time by the mixed reality system. Users can rotate and manipulate the cube to control the orientation and position of a virtual object. The authors found that the Tangible Cube performed well in terms of accuracy and user preference, with participants reporting that it was intuitive and easy to use. However, they also noted that the technique may be less effective for more complex tasks, where more degrees of freedom are required. Overall, the paper highlights the potential of tangible user interfaces for object manipulation in mixed reality, and provides insights into the strengths and limitations of specific techniques like the Tangible Cube.

Likewise mentioned, spherical objects may also play a similar or, possibly, a better role. David et. al. explore the use of a tangible, handheld sphere as a physical proxy [5] for manipulating virtual objects in Augmented Reality (AR) environments. The paper presents a button-less interaction technique that is suited to the characteristics of the sphere and compares it with three controller-based methods in a user study. The spherical object has 6 degrees of freedom same as a handheld controller, but can provide a more comfortable experience and higher perception of control with the virtual object in place. The study focused on an alignment task in mid-air and on a surface and found that the handheld sphere has significant advantages in task completion time and user perception. The findings suggest that a tracked sphere provides a solid basis for 3D spatial manipulation in AR environments, particularly those that require fast comprehension of interaction with natural objects. However, more work needs to be done to explore the use of other shapes and buttons/interaction surfaces on the sphere.

Thus, to allow a better user-experience it is important that the user has a good sense of control and perception of the virtual object. In our case, the slider will be the virtual object that the user can have control of and the only moves associates with it are of translation type, at a 2d level. Thus, since only translation moves are needed fewer degrees of freedom are necessary for user to achieve the appropriate level of control. This implies that any object can be well suited for such a purpose, without loosing the perception of control.

2.3 Everyday Objects as Proxies

In addition to commercial headsets and controllers, researchers explored different handheld devices in order to enhance user's immersion and virtual experience. The handheld controllers aim to provide sense of touch feedback and the solutions focused on varying components such as weight shift [14, 15], stiffness [9], texture [13], damped oscillation [12], grasping and throwing [8], and resistance and flow [6]. These controllers, however, utilize complex and specialized devices and are only applicable to certain scenarios or applications. These controllers are also hard to recreate by regular users as it requires expensive equipment such as 3D printers. Furthermore, user experience evaluation on the above prototypes indicated heaviness as a common aspect that disrupts immersion. To resolve these limitations, researchers explored different approaches on leveraging everyday objects, that are readily available in user's surroundings, as proxies in virtual reality.

The paper titled "Substitutional Reality: Using the Physical Environment to Design Virtual Reality Experiences" [10] investigates the use of everyday objects to interact with virtual objects. In the first experiment, the research explored the use of an actual mug in order to interact with virtual mugs that have different substitutions. For the second experiment, users play a lightsaber virtual game and used an umbrella, torch and lightsaber replica as the controller. For their first experiment, they determined that objects that had different shapes and temperature detracted users from their suspension of disbelief. In the lightsaber test, torch was preferred over the umbrella and replica and the main reason is because it is lightweight and therefore less tiring.

The research article titled "Usability Evaluation of a Tangible User Interface and Serious Game for Identification of Cognitive Deficiencies in Preschool Children" [2] investigates the usability of a tangible user interface (TUI) and a serious game for identifying cognitive deficiencies in preschool children. The study employed a usability evaluation method of using a stuffed toy as a controller in a serious game. The study showed that the TUI was effective in providing children with an engaging and intuitive way of interacting with the game, which could potentially aid in the early detection of cognitive deficiencies. The physical interface provided a level of engagement and interaction that was not present in traditional cognitive assessments, making it an innovative tool for assessing cognitive abilities in preschool children. Our project may show contributions and a possible practical application in such scenarios. Thus, Arcade Brick Breaker may be also used as a game to apply cognitive assessments on users.

In order to further explore the use of everyday objects in virtual environment, [4] introduces Ad hoc UI which is a prototype toolkit that empowers users to turn everyday objects into opportunistic interfaces on the fly. The paper defined opportunistic interface as a type of user interface that enables users to interact with everyday objects in their environment by associating virtual content with physical objects. With Ad hoc UI, users can transform everyday objects such as business cards, coffee mugs, and even a person's hand into tangible interfaces. For example, a user can pick up a business card and both sides will show different widgets that allow their own unique tangible interactions. Another example is anchoring UIs to textured objects while tracking their 6DoF poses, which allows users to interact with the UIs via touch gestures. Ad hoc UI is a prototyping toolkit which the authors hoped that other researchers can use in order to explore more of integrating everyday objects into virtual reality objects.

Our work differs from all the above by proposing a novel type of using everyday objects as components of virtual game with real-time feedback and without requiring the use of VR/AR headsets.

3 DESIGN

Our design for this project includes creating a 2D brick breaker game in a 3D AR environment. We plan to incorporate a player paddle that is controlled via a physical real world object and as well as have bricks that are overlayed on top of a physical real world object as well.

3.1 What is BrickBreaker?

BrickBreaker is a classic computer game from the mid 1970s[1] it was first developed on the Atari and was called "Breakout". This classic game required the user to keep a ball bouncing between a paddle controlled by the player and bricks that break on the other side of the screen when the ball touches them. The user wins when all the bricks are destroyed, and subsequently the user loses when they miss the ball coming back to them.

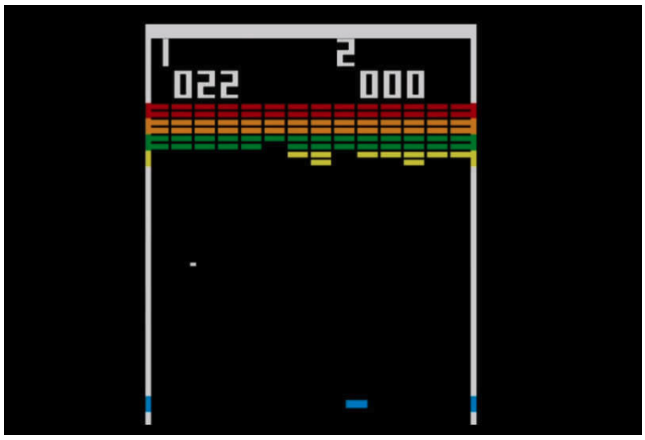


Figure 2: Breakout (1976)

3.2 AR Camera

Since this project is based around accessibility and usability, it will only involve the use of 1 singular smartphone camera. Before the implementation of the project we also had thoughts of using more complex mixed reality systems such as Microsoft Hololens but decided against it, as our goal is to make this project as accessible as possible.

3.3 Player Paddle

The mixed reality aspect of our game stems mainly from our player paddle. We designed the project to use a physical real world object as the controller for the player's paddle in the brick breaker game. This allows for the player to control how the paddle moves in the virtual world with a physical object in the real world. We believe this will allow for more accessibility and ease of use than a traditional game pad or arrow keys.

3.4 Bricks

The second mixed reality aspect of our game will come from how the bricks are placed, we plan to overlay the brick placement with a real world object, thus the brick size and density will be determined on the size and shape of the object in the real world. This will add another layer of uniqueness to our project.

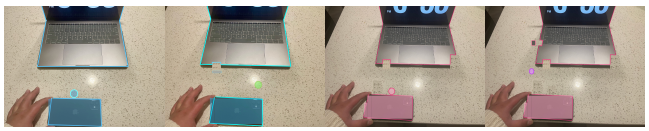


Figure 3: Our Lofi Design and Idea

4 IMPLEMENTATION

Our development timeline consisted of dividing up the project into multiple tasks, then merging them back together at then end. Each task aimed to tackle as specific unique aspect of the project and was designed to have no overlaps with other tasks. Since source control was a major issue in the development of the project, the tasks were designed to be able to complete individually and without the need for other code or dependencies with other sections.

4.1 Initial AR Testing

One of our first tasks were to design and implement basic object tracking to merge the physical world with the virtual one. The goal of this step was to be able to project some object or plane onto a physical object in the real world, and be able to move the virtual plane or object with the physical one. This gave us the basis on how we would implement our player paddle and interaction throughout our entire project. We decided to experiment with a QR code (either printed or put on a phone screen) and attempted to project a basic cube onto it (figure 3). We found that the tracking was quite precise and thus we decided on using this option going forward.

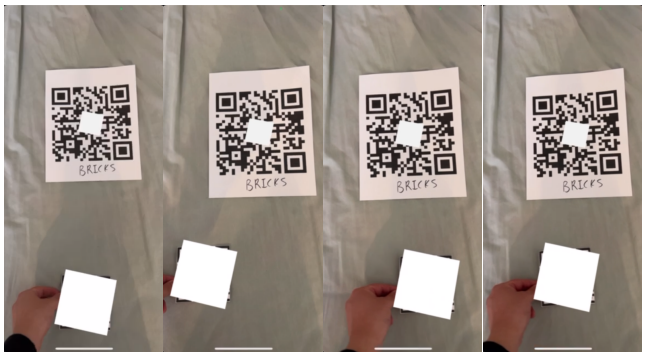
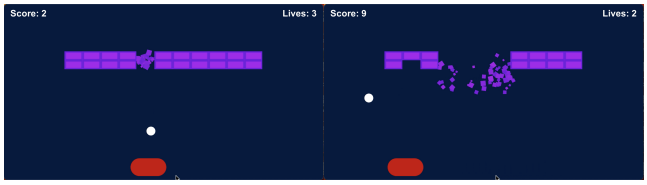


Figure 4: AR object tracking

4.2 Game Logic

Our next objective was to learn and implement how to create a 2D brick breaker game in our Unity game engine. The goal of this step was to create a working completely virtual brick breaker game that we can use as the basis for our game logic moving forward. Important aspects such as the physics on how the ball bounces from the sides and from the paddles were discussed and initially worked on in this step. Though this step did not directly contribute with the development of our project, it provided us with a important understanding of game engine physics and collusion physics. At the end of this step we implemented a full virtual (non VR/AR) brick breaker game, with complete game logic and physics for us to base our project off of.



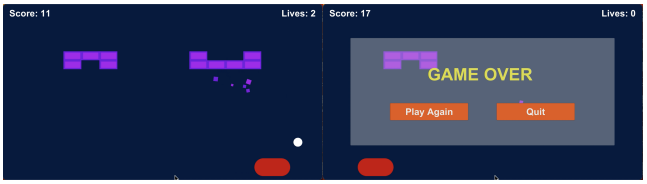


Figure 5: Brick Breaker Game Play

4.3 Closing the Loop (initially)

With the above steps complete, our next goal was to move the physics and game logic from a 2D space to our AR environment. We wanted to see how much of the game we can really transfer over into a mixed reality 3D environment. Our initial design idea was to use the borders of the camera as the borders of game, such that the ball will bounce around the framing of our camera. Upon implementing our game to AR, we realized this is not feasible and instead made virtual boarders for the ball to bounce around instead. The second part of this step was to implement the object tracking code discussed in 3.1, this was a relatively simple procedure as we directly reused the code implemented before and inserted it into our project. We continued to use a QR code as our physical object but modified the cube to be in line with the design of our player paddle.

4.4 Restricting Plane of Motion

Upon inserting our object tracking code into the project, we discovered that the tracking library inherently tracks the projected object in all 3 dimensions, meaning that even though our game state is projected on a single z axis (since we are making a 2d game) the object being tracked is moving in all 3 dimensions. Thus if the paddle is not positioned in the correct distance from the camera, the player paddle in game will also not be in the right plane and thus not register a ball hit as it is bellow or above the plane where the ball is. To solve this issue we increased the size of the player paddle vertically, giving the user more wiggle room on where to place the paddle.

4.5 Adding Bricks

Our next step in this project is to implement how the bricks will function in the 3D environment, we designed some virtual cubes and placed them around the game area. when the ball hits one of these bricks, it explodes and removes itself from the environment.

4.6 OpenCV

To implement the second mixed reality portion of the project, we used openCV to layer mask a physical object and detect it in the virtual environment, we then take the tracked physical object and layer ontop the bricks that we have created in 4.5. Thus finishing the bricks portion of the project. However, we found that openCV seemed to be depreciated with our current android systems, thus this part of the project was removed due to time constraints. Instead we used complete virtual bricks.

4.7 Closing the Loop (final product)

Finally with all the previous steps combined, we collaborate and input every step together to create our final product. During this

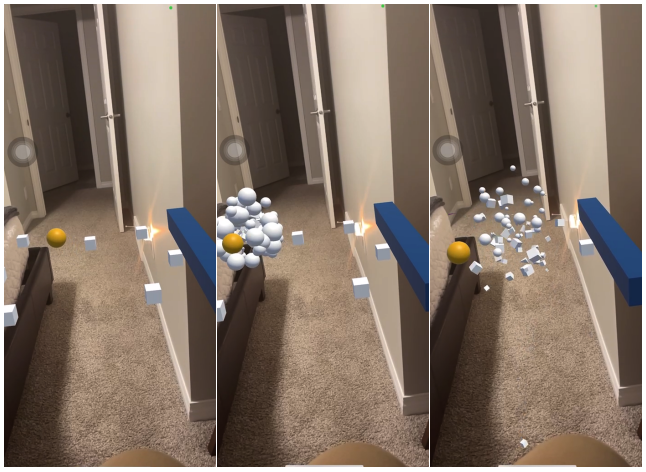


Figure 6: Bricks implementation

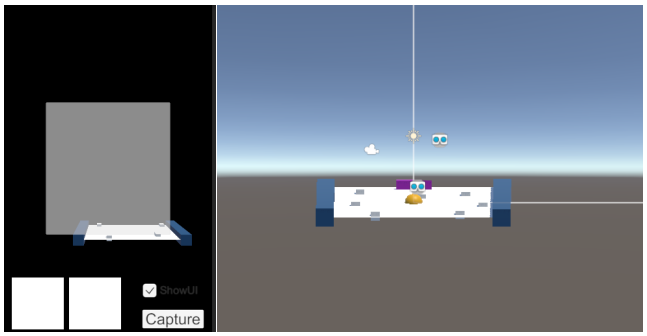


Figure 7: OpenCV Attempt

step we fixed any scaling issues we had and ensured our camera is where we want it to be. We did final touch ups including code optimizations and aesthetic changes to round off the implementation.

5 FINDINGS

As we had very limited time to complete this project as well as restrictions on finding user groups to study, it was difficult to find a good sample size to test and try out our project. However between the 5 of us and a few other individuals we were able to derive preliminary findings on our project.

5.1 Usability

Since our project is designed around giving entertainment and recreation to others, we wanted the usability of our app to be easy and simple. Our primary goal was to keep the controls of the game intuitive and the game design easy and straightforward. We found that we did achieve what we wanted, the game is intutive and easy to understand, and players tend to not have any issues or questions understanding how the game works or how the controls work. However, there were definitely some issues which decreased usability as well. A good example would be some slight jittering and movement of the entire game area during the course of the player playing the game, the game would move ever so slightly to the left or right when the camera is moved around (this was more prevalent in larger movements and sudden movements). This made controlling the player paddle a little harder than it should have been.

Another issue we realized was that the game was genuinely quite hard, moving the physical object that controls the player paddle seemed harder than it looked. Although we do not have a direct answer on why this is, we still felt it was important to take note of it.

5.2 Fun Factor

Even though the concept of "fun" was not in the forefront of our development ideology, it seems only appropriate to touch on this slightly as we are developing a video game. The classic Brick Breaker concept has been around for decades, its tried and true game mechanics still to this day stand up to the test of time. As we are implementing a modified version of this game, our core game mechanics stay the same as it would have been for the original. Though the idea of "fun" is subjective, we believe our core concept of the game can still be considered "fun". However, as stated in the previous segment, our implementation of the game seems to be more difficult than a purely virtual game of Brick Breaker. Although we thought by using physical objects will make the game more accessible, it in turn also made it much more difficult and harder to control, thus decreasing its "fun" factor.

5.3 Accessibility

Our primary concern throughout this endeavour was to keep the accessibility of our app high and friendly. We avoided using niche AR/VR technologies as we wanted to garner our project to the general public. We found that using simple devices such as smartphone cameras and everyday physical objects allowed easy accessibility for all ages and genders. However, we found that sometimes when the game starts up from your device, there sometimes is a slight issue where the user is unable to find the game space as their camera is not pointed in the right direction. This did in fact cause some confusion for the user before they were able to find where the game space is. As well, we found that our scaling for the game elements could have been a little larger, as sometimes the game space seems a little small.

6 DISCUSSION

During the development of this project, there were many roadblocks and interesting issues that arose. However, these issues were not complete blockers and thus could not be classified as a limitation of the project as a whole. This section will cover the main challenges we faced while working through this project in the last 3 months.

6.1 Learning Curve

Since each group member for this project is an undergraduate student in their senior years, none of us had any prior knowledge in developing AR or VR applications nor even how it all works on a theoretical level. This proved to be a large challenge for us as we all had to spend a large amount of time understanding and studying how Unity and AR development works. Furthermore, due to a lack of experience when issues arose during coding sessions, it was difficult for us to understand and diagnose the issue. Very minor issues can in turn take hours upon hours of debugging and learning to fix.

6.2 Challenges with plane of motion

When implementing our player paddle with the image tracking in Unity we approached an issue where the player paddle is tracked in all 3 dimensions, it means the paddle will be projected right ontop of where the physical object in the real world. This unfortunately is not our desired output, as we are designing a 2D game in the AR environment. Thus we need the vertical (Z) axis to be locked in place (on the same plane as our game space). We brainstormed potential solutions for this including trying to write scripts to overwrite the movement tracking in the z axis and as well as using different image tracking libraries. However none of these solutions seemed to work in any case and after some consideration we decided on just increasing the length of the player paddle to give the user more wiggle room on how far they need to hold the physical object away from the camera.

6.3 Tracking Issues

Another issue that stumped us during our development the sensitivity of the image tracking library. The tracking of our player paddle was not as consistent as we would have liked it to be. Often times if the lighting was off or if there were slight wrinkles in the paper (where the QR code was printed) it would cause the tracking to lag or disappear completely. As well, when there are quick rotations of the physical object, it would throw off the tracking and cause the player paddle to be rotated as well (twisted the flat side is not fully facing the game space). We concluded that this is most likely just a limitation of the tracking technology available to us at this day and age and since our game is not very fast paced and does not need super precise movements of the player paddle, we left this issue alone and procceded as planned.

6.4 2D to 3D

When developing anything in VR or AR, it is developed in a 3 dimensional environment, and thus naturally any application must take into account x, y and z axis. However, since ARcade is a 2 dimensional game, we had to create work arounds in the 3D environment to make our game work. As talked about in 6.2, a major issue we faced was restricting the plane of motion of our player paddle. However, this transition between dimensions also caused other issues such as creating our game space. Since we needed a ball to be able to bounce back and forth between a player and the bricks, we had to implement a virtual game space with boundaries in order to make it work. Our initial idea was to use the camera borders as our boundaries for the ball, but after hours of testing and debugging, we were unable to merge this 2D aspect into a AR environment where the camera also tracks the distance it is from other objects. This project made us realize that augmented reality is truly designed for 3D software over 2D, and if we were to design this app from the ground up again we would keep that in mind for our design.

6.5 Ball Bounce Issue

Tagging onto our development issues with the player paddle, another problem we came across was that the object tracked player paddle would not have any physical physics attached to it. Thus when the ball would hit the paddle, instead of bouncing off it into the opposite vector, it would attempt to continue along its path and roll off the paddle. Initially we thought the issue lied with the script we created not being attached properly to the new paddle, but after some investigation we ruled out that possibility. As it turns out, the issue stemmed from the player paddle not being instantiated as a "game object" within Unity, which caused Unity to not read any scripts attached to the paddle. After another bout of debugging and trial and error, we were able to reference a game object with the player paddle and the physics worked as intended.

7 LIMITATIONS AND NEXT STEPS

Since all the work that was discussed in this paper was done in a span of 3 months, as well as each group member also having many other commitments (other classes) to deal with at the same time. Naturally there are many limitations and issues we were not able to figure out in the time frame alotted. As well as this project being an unfunded one as well, we were unable to access some features which we thought were key to developing in Unity. This section will go over those such limitations as well as some next steps if we were to continue to work on this project in the future.

7.1 Time Crunch

Unlike longer more relaxed projects, this project had a time frame of about 3 months. This definitely proved to be challenge for us, as developing an idea and completely following through with it in such a short time frame was extremely difficult. As well, since each member in this project is also an undergraduate student with other classes and commitments, it diminished the time we each had to work on this project even more. Subsequently, since none of us had any prior experience in Unity or mixed reality development, many issues and bugs that arose during our development process took much longer to debug and figure out than anticipated. With all that in mind, if we were to redo this project again and were given a 3 month time limit, we would have considered a simpler project or idea, and as well as looked to seek for help and aid much earlier in our development timeline to avoid congestions and long debug hours.

7.2 Source Control

Early on into our development process, we found a critical issue in regards to source control. While each group member had plenty of experience in group work and using GitHub and Git as the primary source control for every project, we realized that git and Unity are not compatible. Thus from the very start of our development process we were stuck with the issue of sharing and dividing the work up so that each of us are able to work on up to date code simultaneously. We initially did research into how companies who use Unity in the professional world collaborate and found that there is indeed a source management service for Unity called "Plastic SCM", however this service was locked behind a paywall and thus for did not work for us. With no other viable option, we were forced to zip up our raw project data and send it to each other via file sharing services, ultimately managing source control manually. Since this is the case, our development efficiency suffered as well.

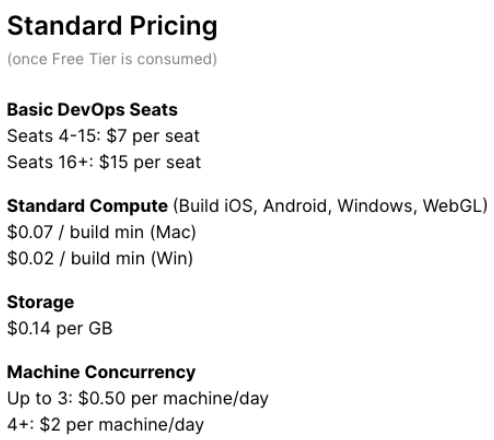


Figure 8: Plastic SCM paywall

7.3 User Study

After completing the development process, we wanted to do a user study on the design parameters defined in section 5. However due to time constraints and ethical constraints by the department of computer science we were unable to perform a study strong enough to have much statistical value. Though this was not a major aspect of this project, if time and ethics permitted we would have liked to develop and perform a full case study to compare our game to a traditional BrickBreaker game like "Breakout".

7.4 Graphics and Sound Design

The graphics produced in this project are simple and quite bare-bones, as asthetics and graphical looks were not the key idea behind this project we did not spend much time on developing and shaping our game to be unique in looks. As well, there are no sound effects implemented as well. If this project were to be further developed, designing our game to have a unique look and feel would be a concept we would take a look at and spend more time on.

7.5 More Game Logic

In our development process defined in section 4, we discussed a development of a fully virtual BrickBreaker game to aid us in the development of the game logic in AR. Although we developed a full working game virtually, not many of those features made it into our final product. Game logic such as lives and score keeping that were implemented in the virtual game ultimately were too complex to include in the final version of our game (due to time constraints). Though we have basic game logic down on our AR product for it to function, our next steps would have been to continue to implement other computer game concepts such as title screens, lives, scoring and levels.

7.6 Increase Usability

In section 6.3 we discussed some issues with our tracking library and scripts, this in turn decreased the usability of our product. As well, not having aspects such as title screens, tutorials and game settings further decreased the usability of the project. Again, the leading limiting factor for this would have been the tight turn around time required for this project and the amount of time each group member had to work on this over the course of the 3 months. Since usability is the core to our design for this project, further implementations of ARcade would primarily focus on increasing the usability of the product. Some improvements we can see happening is more research into the tracking algorithms, implementing more game logic and designing the game to be more asthetically pleasing.

8 CONCLUSION

ARcade, a mixed reality version of the classic game design "Brick Breaker" ultimately did prove to provide a fresh look and twist to the much loved game from the 70s. By using tangible interfaces as the player paddle we found that it made it easier for users to understand and play the game, and the mixed reality of the game ultimately made the game more fun and unique. However due to limitations in time, experience and technology, we also found the game to be harder to play, and can sometimes be less intuitive to play due to design choices and bugs. Overall ARcade is a good starting point into more in depth research on usability of video games in the mixed reality world and we hope to be able to continue studies down this path in the future.

REFERENCES

[1] 2020. A Brief History of Brick Breaker Video Games. *Hero Concept* (2020).
[2] Sánchez-Morales. A, Durand-Rivera J.A, and Martínez-González C.L. 2020. Usability Evaluation of a Tangible User Interface and Serious Game for Identification of Cognitive Deficiencies in Preschool Children. *International Journal of Advanced Computer Science and Applications* 11, 6 (2020). https://doi.org/10.14569/IJACSA.2020.0110661
[3] Evren Bozgeyikli and Lal Lila Bozgeyikli. 2021. Evaluating Object Manipulation Interaction Techniques in Mixed Reality: Tangible User Interfaces and Gesture. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. 778–787. https://doi.org/10.1109/VR50410.2021.00105
[4] Ruofei Du, Alex Olwal, Mathieu Le Goc, Shengzhi Wu, Danhang Tang, Yinda Zhang, Jun Zhang, David Joseph Tan, Federico Tombari, and David Kim. 2022. Opportunistic Interfaces for Augmented Reality: Transforming Everyday Objects into Tangible 6DoF Interfaces Using Ad Hoc UI. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems (CHI EA '22)*. Association for Computing Machinery, New York, NY, USA, Article 183, 4 pages. https://doi.org/10.1145/3491101.3519911
[5] David Englmeier, Julia Dörner, Andreas Butz, and Tobias Höllerer. 2020. A Tangible Spherical Proxy for Object Manipulation in Augmented Reality. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 221–229. https://doi.org/10.1109/VR46266.2020.00041
[6] Seongkook Heo, Christina Chung, Geehyuk Lee, and Daniel Wigdor. 2018. Thor's Hammer: An Ungrounded Force Feedback Device Utilizing Propeller-Induced Propulsive Force. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems (CHI EA '18)*. Association for Computing Machinery, New York, NY, USA, 1–4. https://doi.org/10.1145/3170427.3186544
[7] Hiroki Kaimoto, Kyzyl Monteiro, Mehrad Faridan, Jiatong Li, Samin Farajian, Yasuaki Kakehi, Ken Nakagaki, and Ryo Suzuki. 2022. Sketched Reality: Sketching Bi-Directional Interactions Between Virtual and Physical Worlds with AR and Actuated Tangible UI. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology (UIST '22)*. Association for Computing Machinery, New York, NY, USA, Article 1, 12 pages. https://doi.org/10.1145/3526113.3545626
[8] Robert Kovacs, Eyal Ofek, Mar Gonzalez Franco, Alexa Fay Siu, Sebastian Marwecki, Christian Holz, and Mike Sinclair. 2020. Haptic PIVOT: On-Demand Handhelds in VR. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology (UIST '20)*. Association for Computing Machinery, New York, NY, USA, 1046–1059. https://doi.org/10.1145/3379337.3415854
[9] Neung Ryu, Woojin Lee, Myung Jin Kim, and Andrea Bianchi. 2020. ElaStick: A Handheld Variable Stiffness Display for Rendering Dynamic Haptic Response of Flexible Object. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology (UIST '20)*. Association for Computing Machinery, New York, NY, USA, 1035–1045. https://doi.org/10.1145/3379337.3415862

[10] Adalberto L. Simeone, Eduardo Velloso, and Hans Gellersen. 2015. Substitutional Reality: Using the Physical Environment to Design Virtual Reality Experiences. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. Association for Computing Machinery, New York, NY, USA, 3307–3316. https://doi.org/10.1145/2702123.2702389

[11] Ryo Suzuki, Rubaiat Habib Kazi, Li-yi Wei, Stephen DiVerdi, Wilmot Li, and Daniel Leithinger. 2020. RealitySketch: Embedding Responsive Graphics and Visualizations in AR through Dynamic Sketching. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology (UIST '20)*. Association for Computing Machinery, New York, NY, USA, 166–181. https://doi.org/10.1145/3379337.3415892

[12] Hsin-Ruey Tsai, Ching-Wen Hung, Tzu-Chun Wu, and Bing-Yu Chen. 2020. ElastOscillation: 3D Multilevel Force Feedback for Damped Oscillation on VR Controllers. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3313831.3376408

[13] Eric Whitmire, Hrvoje Benko, Christian Holz, Eyal Ofek, and Mike Sinclair. 2018. Haptic Revolver: Touch, Shear, Texture, and Shape Rendering on a Reconfigurable Virtual Reality Controller. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3173574.3173660

[14] André Zenner and Antonio Krüger. 2019. Drag:On: A Virtual Reality Controller Providing Haptic Feedback Based on Drag and Weight Shift. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300441

[15] André Zenner and Antonio Krüger. 2017. Shifty: A Weight-Shifting Dynamic Passive Haptic Proxy to Enhance Object Perception in Virtual Reality. *IEEE Transactions on Visualization and Computer Graphics* 23, 4 (2017), 1285–1294. https://doi.org/10.1109/TVCG.2017.2656978