Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Group Assessment Submission Form

CS7GV6 Project 2

Group 4

1. Group Members and Contributions

Group Member Name (Alphabetical Order by Surname)	Contribution & Role in Project
Viraj Vilas Bhor	3D Assets & Environment – created custom assets and sourced required assets from online resources
Sarvesh Suresh Jaiswal	UI/UX Designer – responsible for the design and implementation of UI elements
Yokesh Muthu Kathiravan	Physics Programmer – contributed to physics interaction prototyping and assisted with testing
Priyansh Nayak	Technical Lead – overall architecture, system integration, and coordination
Akshit Shrivastava	Gameplay Designer – designed game rules and feedback systems

---

2. Video Submission

YouTube Link to Project Video (5-7 mins max): https://youtu.be/D6GAtmjE1pk

Project Repository Link: https://github.com/Norged-Out/Carnival-Games

---

3. Project Implementation Description (up to 1 page)

The project is a **Virtual Reality Carnival Games experience** developed using **Unreal Engine 5.7.1** with the **UE VR Template (OpenXR)**. The application was tested on **Meta Quest 2 and Meta Quest 3** headsets to ensure correct VR interaction, controller input handling, and performance stability. The experience consists of three classic carnival-style mini-games: **Cup Topple**, **Ring Toss**, and **Dart Throw**, each designed with a focus on **physics-based interaction**, **clear gameplay structure**, and **immersive VR feedback**.

Each mini-game operates within a **defined play zone**. When the player enters a zone, the corresponding game is initialized and the player is granted a **limited chance to play**. Once an attempt is completed, the game transitions to an end state and further scoring is disabled. If the player exits the play area and re-enters, the game automatically resets to its initial configuration, allowing the challenge to be replayed. This approach ensures controlled gameplay flow and prevents unintended repeated scoring.

The **Ring Toss** game allows the player to grab and throw rings toward a set of vertical target pegs. The rings are physics-enabled objects, requiring the player to judge distance, direction, and throwing force accurately. Successful attempts are detected using **overlap zones** combined with stability checks to ensure that the ring has properly landed on a peg rather than briefly intersecting it. Once an attempt concludes, the score is registered and the game ends. Audio feedback is triggered depending on the ring's evaluation of successful or unsuccessful tosses.

In the **Dart Throw** game, the player throws darts toward a target board and moving balloons using VR motion controller input. Dart motion is influenced by gravity and the initial impulse applied during the throw. Upon collision with the targets, the dart transitions from simulated motion to an attached state, creating the visual effect of the dart sticking to the surface. The target is divided into scoring regions, and points are calculated based on the region struck. Audio feedback is triggered on successful hits to enhance player feedback and immersion.

In the **Cup Topple** game, the player interacts using a **gun-based shooting mechanism** rather than direct hand throwing. The gun fires a physics-enabled ball toward a stacked arrangement of cups placed on a table. The ball follows a physics-driven trajectory, and cup interactions are handled using collision detection. The cups are physics-enabled and respond naturally to impacts, allowing them to topple or fall based on the force and direction of the hit. Scoring is determined by detecting successful impacts and displacement of cups from their stacked positions.

A **centralized Score Manager system** collects scores from all three mini-games. After the player has completed all available games, the system displays **individual game scores along with a final combined score**, providing a clear summary of the player's overall performance across the carnival experience.

To support immersion and visual consistency, the environment was created using a combination of **custom 3D assets modeled in Blender** and **assets sourced from the Unreal Engine Marketplace (Fab)**, ensuring both originality and efficient development.

---

4. Group Organization (up to 1 page)

The group adopted a **role-based development structure** to ensure efficient collaboration, clear responsibility boundaries, and smooth integration of system components. Each member was assigned responsibilities aligned with their strengths, which helped streamline development and reduce overlap.

The **Technical Lead** coordinated the overall project architecture, managed system-level integration, and ensured that all mini-games functioned cohesively within a single VR experience. This role also involved resolving integration issues, refactoring incompatible systems, and maintaining consistency across gameplay mechanics and scoring logic.

The **Gameplay Designer** focused on defining the intended rules, behaviour, and flow of each minigame. This included outlining scoring logic, gameplay feedback mechanisms, and contributing audio cues to reinforce player actions. Design decisions were iterated through testing and feedback sessions to refine the overall gameplay loop.

The **Physics Programmer** contributed to the prototyping of physics-based interactions and assisted with testing gameplay mechanics in VR. This role was intended to support the development of reusable physics systems; however, a portion of the implemented logic required further adaptation and integration to align with the project's centralized architecture.

The **3D Assets and Environment** role involved creating and sourcing assets for the project, as well as assembling the final game environment. This included ensuring that asset scale, placement, and complexity were suitable for VR performance and maintaining visual consistency across the scene.

The **UI/UX Designer** was responsible for designing and implementing some user interface elements required for the games.

Communication within the team was primarily managed through a **dedicated WhatsApp group**, which allowed for continuous discussion, clarification of issues, and feedback during development. This channel was used to share progress updates, discuss implementation challenges, and coordinate task handovers. In addition to asynchronous communication, the team conducted **online meetings twice every week and whenever required**. These meetings were used to review progress, debug technical issues collaboratively, and align on design and implementation decisions.

GitHub was used as the primary version control system, alongside a kanban board to track overall progress. Team members committed their completed tasks to the shared repository, allowing updates to be immediately reflected across the team. This approach supported **parallel development** of the different mini-games and reduced dependency bottlenecks. Regular commits and updates helped minimize merge conflicts and ensured that integration issues could be identified and resolved early. Version control also played an important role in tracking development progress and maintaining a stable project state during the final integration and polishing phase.

Overall, this structured workflow enabled the team to manage the development of multiple VR mini-games within a single project efficiently. Clear role allocation, consistent communication, and disciplined use of version control contributed to maintaining coordination, code consistency, and timely completion of the project.

---

5. Conclusion and Lessons Learned (up to 1 page)

This project provided extensive hands-on experience in designing and implementing a physics-driven Virtual Reality application using Unreal Engine. Developing multiple interactive mini-games within a single VR environment highlighted the importance of balancing realism, usability, and performance, particularly when working with physics-based interactions and motion controller input. Unlike traditional non-VR games, even minor inconsistencies in object behaviour or input responsiveness were immediately noticeable, reinforcing the need for careful system design and validation

One of the most significant lessons learned was the impact of **physics tuning** and **collision handling** on overall gameplay quality. Small adjustments to object mass, collision responses, impulse strength, and damping values produced noticeable differences in how objects behaved in VR. Achieving stable and predictable interactions, such as accurate ring detection, reliable cup toppling, and consistent dart attachment, required iterative testing and frequent refinement. This process reinforced the importance of continuous playtesting, particularly in VR environments where unexpected physics behaviour can quickly break player immersion or cause discomfort.

The project also emphasized the importance of **structured game flow and state management**. Implementing clear play-zone detection, limited play chances, and automatic reset mechanisms ensured that each game behaved consistently and prevented unintended scoring. These systems not only improved the player experience by making game rules more transparent but also simplified debugging and maintenance by clearly defining when gameplay systems were active or inactive.

Another key learning outcome was the value of **modular and reusable system design**. By separating gameplay logic, physics interactions, UI components, and scoring systems into distinct modules, the team was able to integrate multiple minigames more effectively within a single project. The use of centralized controllers and a shared scoring system reduced redundancy and improved maintainability, demonstrating the importance of **scalable architecture** when managing multiple interactive systems in a complex application.

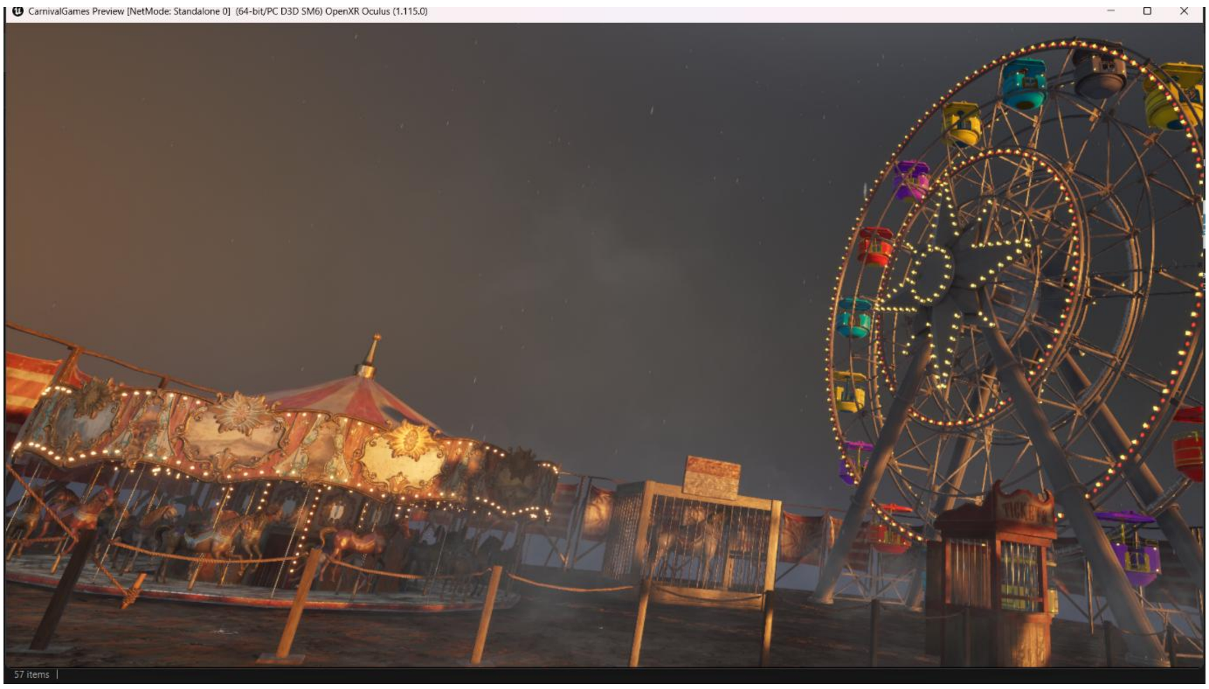
From a collaborative perspective, the project highlighted the importance of **clear role allocation, communication, and cross-role coordination**. Integrating work from different disciplines required consistent communication and iterative feedback, particularly when resolving system incompatibilities or performance issues. Structured testing phases and regular progress reviews helped align individual contributions with the overall project goals and ensured that technical decisions supported both gameplay quality and stability.

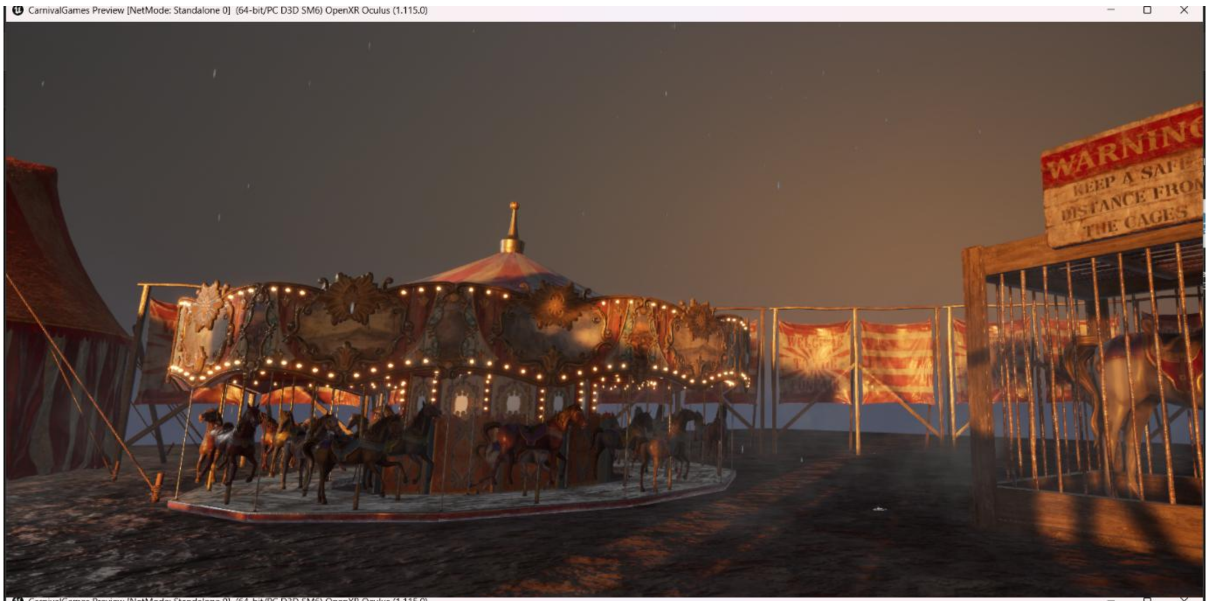
Overall, this project strengthened the team's understanding of **VR interaction design, physics-based gameplay development, and collaborative software engineering practices**. The technical challenges encountered, and solutions developed throughout the project provided valuable insight into building interactive, real-time systems. The lessons learned from this experience form a strong foundation for future work in Virtual Reality, game development, and computer graphics applications, particularly those involving complex physics-driven interactions and multi-system integration.

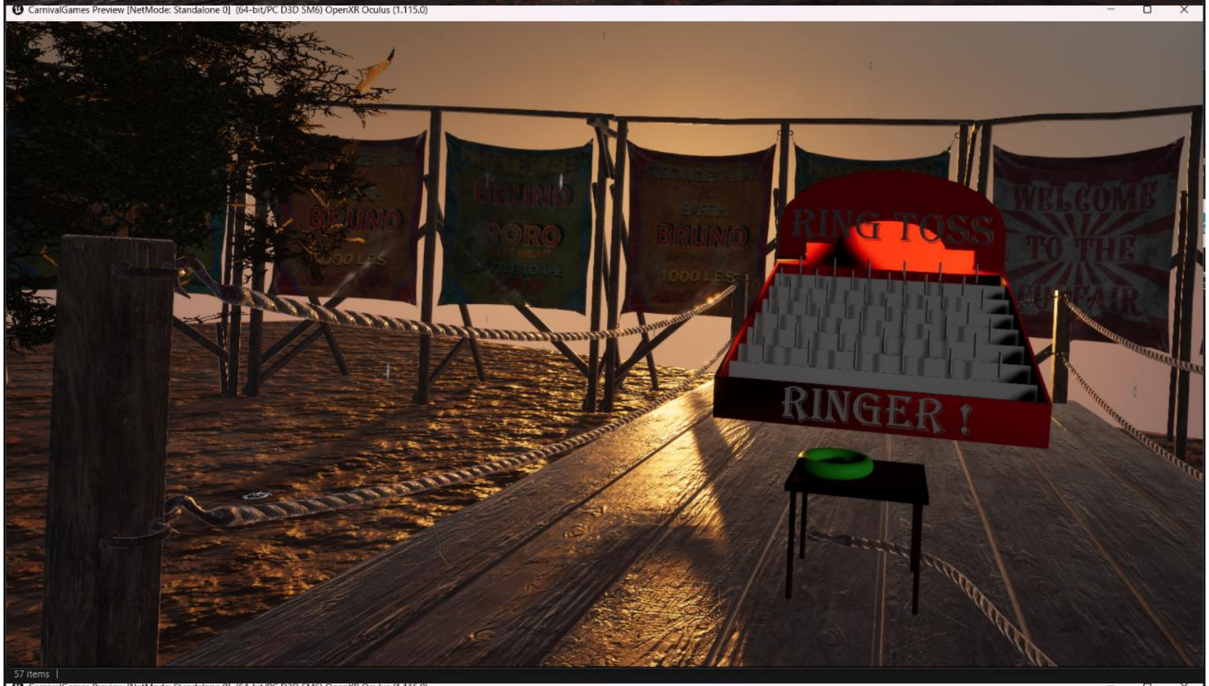
---

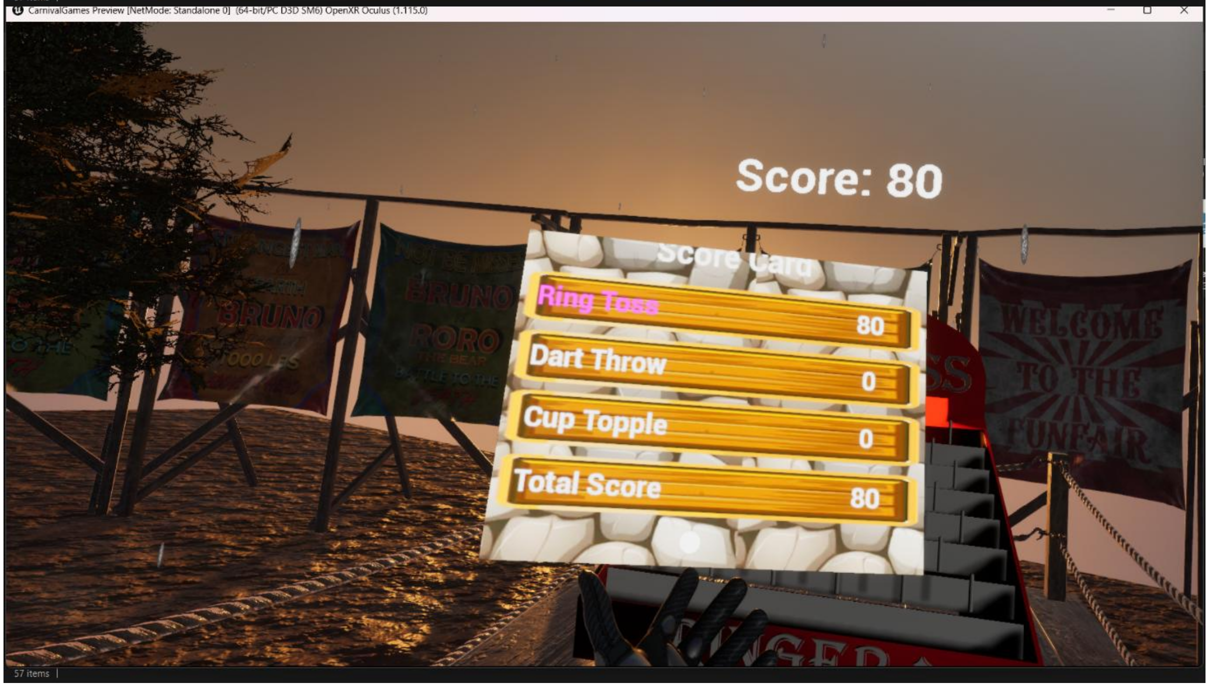
6. Screenshots

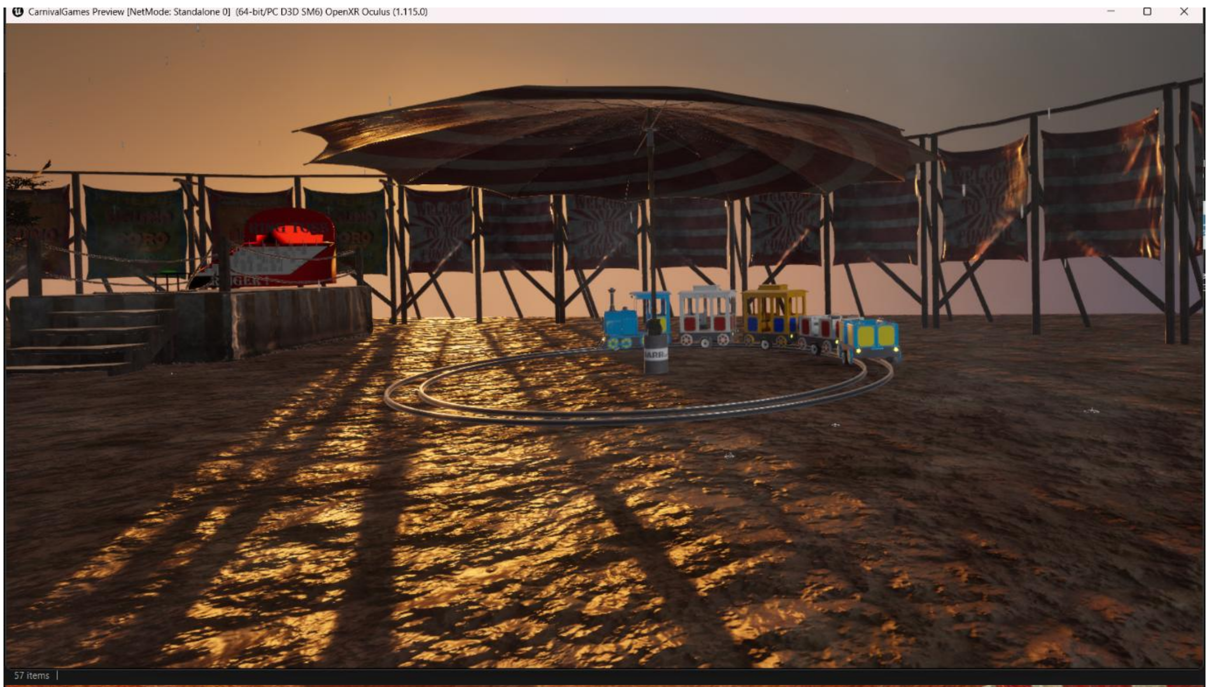
The following screenshots of our final build were taken on VR Preview for ease.

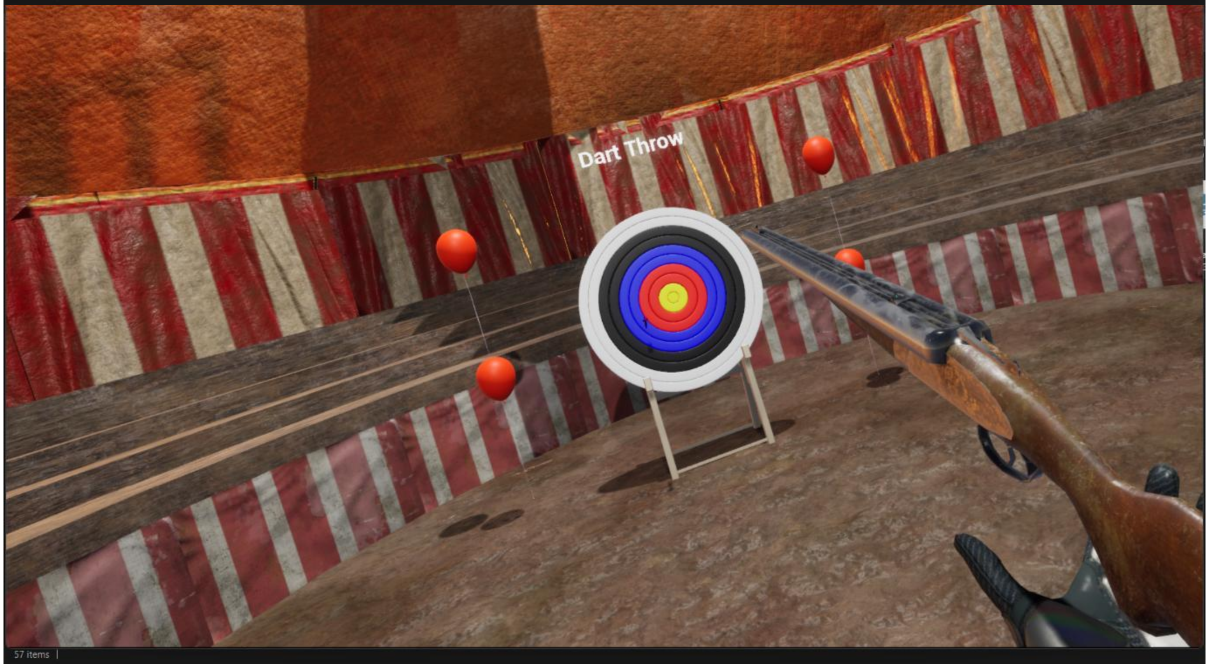

CarnivalGames Preview [NetMode: Standalone 0] (64-bit/PC D3D SM6) OpenXR Oculus (1.115.0)
57 items

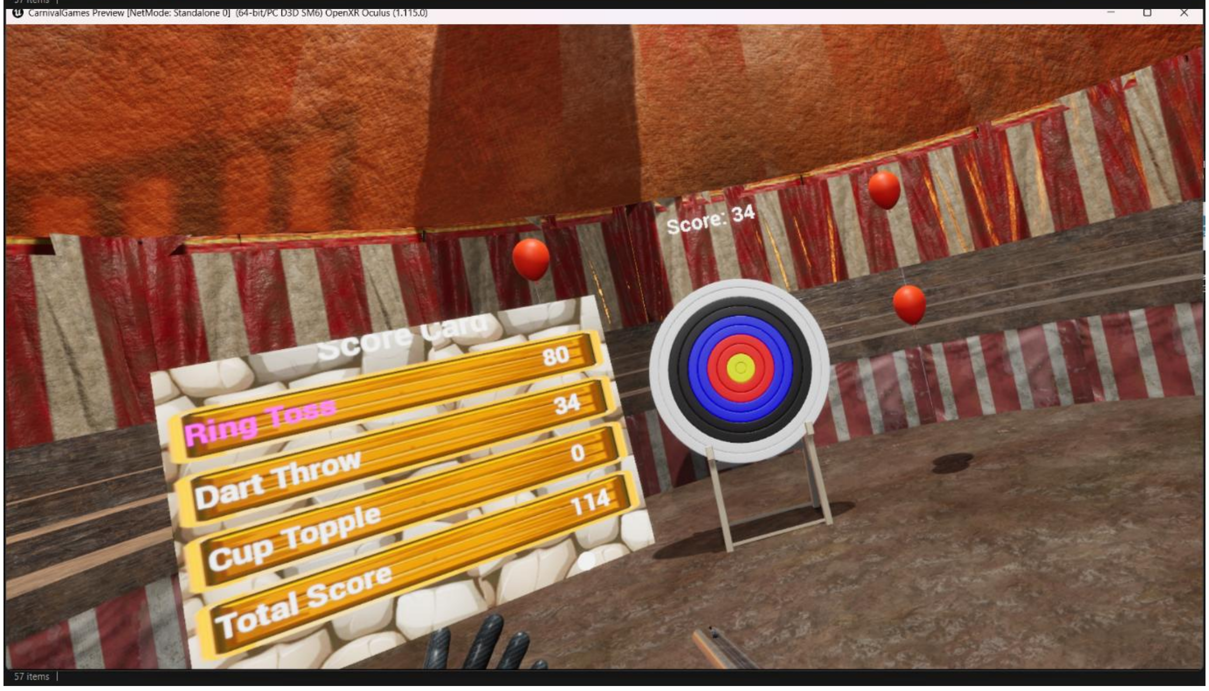

CarnivalGames Preview [NetMode: Standalone 0] (64-bit/PC D3D SM6) OpenXR Oculus (1.115.0)
WARNING
KEEP A SAFE
DISTANCE FROM
THE CAGES

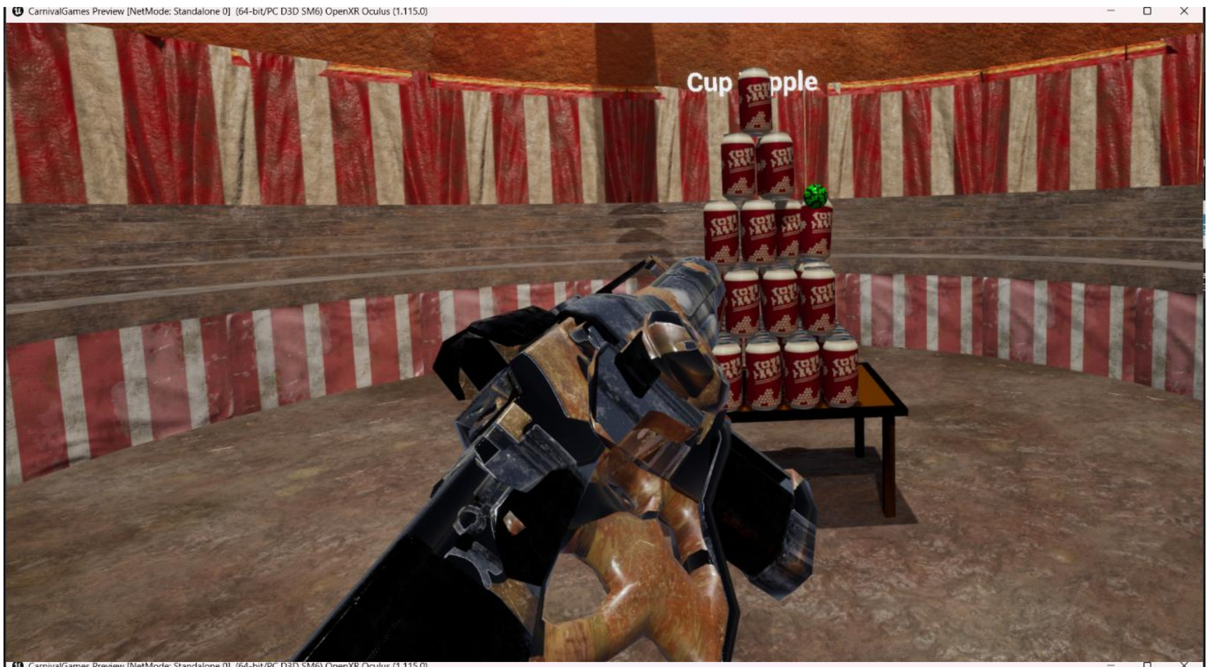

CarnivalGames Preview [NetMode: Standalone 0] (64-bit/PC D3D SM6) OpenXR Oculus (1.115.0)
BRUNO
RORO
BRUNO
1000 LBS
RING TOSS
RINGER !
WELCOME
TO THE
FUNFAIR
57 items

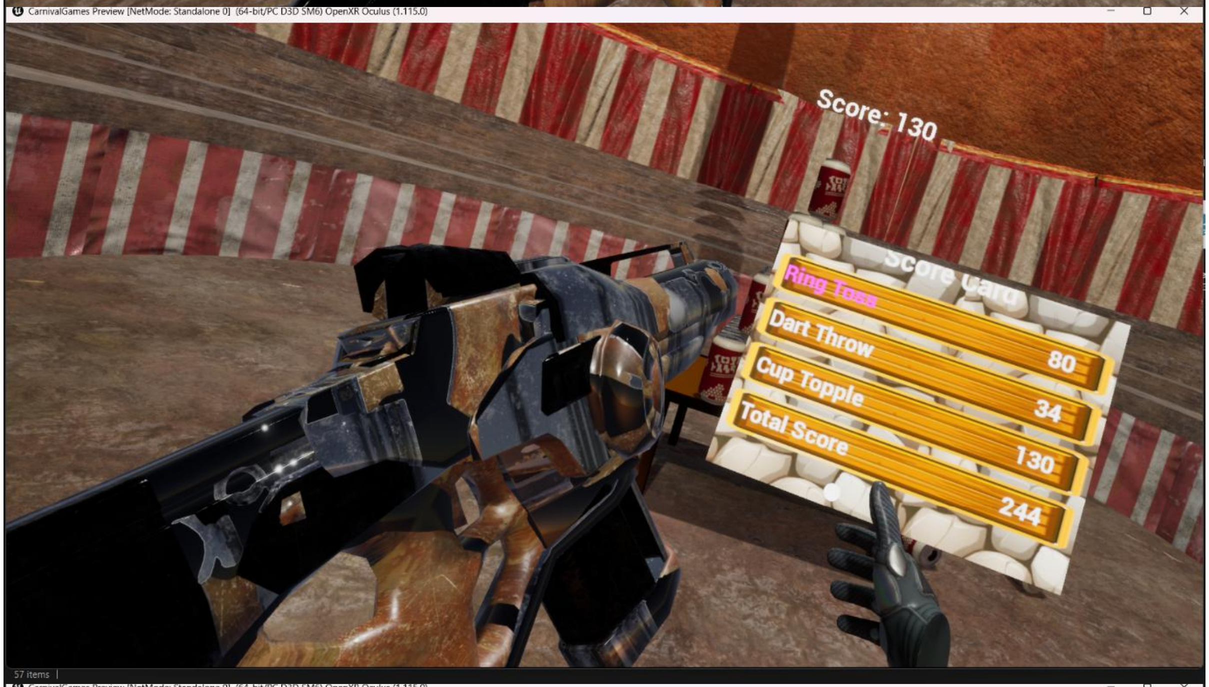

CarnivalGames Preview [NetMode: Standalone 0] (64-bit/PC D3D SM6) OpenXR Oculus (1.115.0)
Score: 80
Score Card
Ring Toss 80
Dart Throw 0
Cup Topple 0
Total Score 80
WELCOME
TO THE
FUNFAIR
57 items

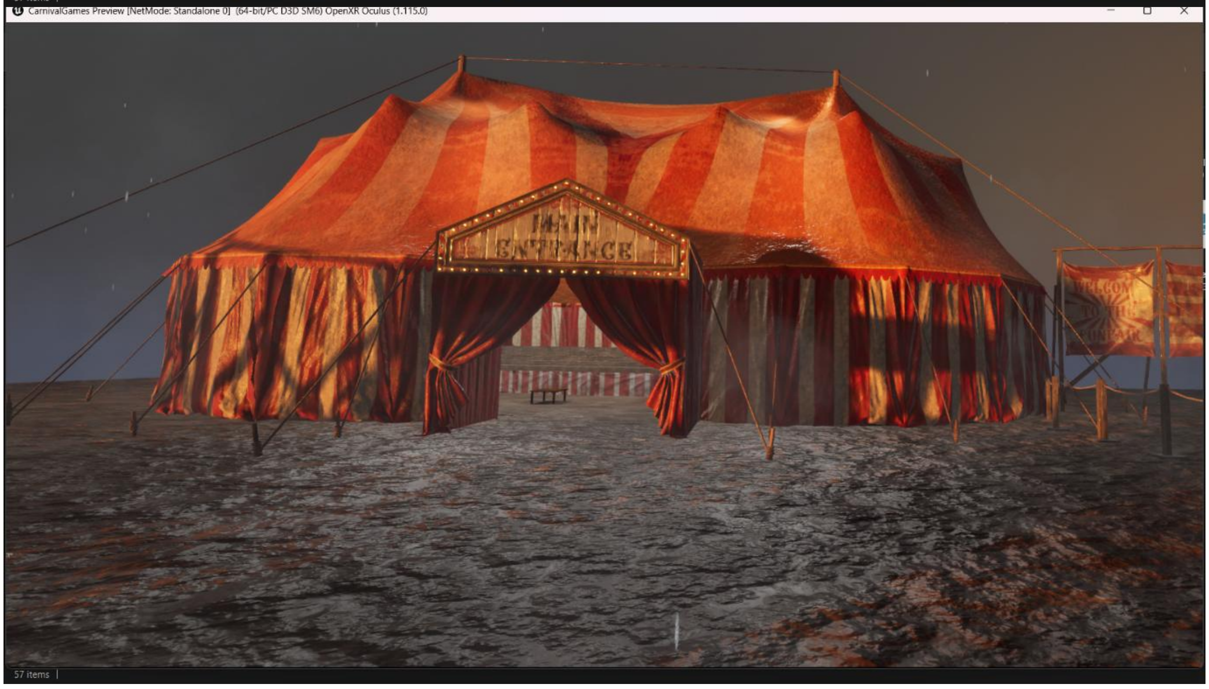

CarnivalGames Preview [NetMode: Standalone 0] (64-bit/PC D3D SM6) OpenXR Oculus (1.115.0)
57 items


Dart Throw
57 items


CarnivalGames Preview [NetMode: Standalone 0] (64-bit/PC D3D SM6) OpenXR Oculus (1.115.0)
Score: 34
Score Card
Ring Toss 80
Dart Throw 34
Cup Topple 0
Total Score 114
57 items


CarnivalGames Preview [NetMode: Standalone 0] (64-bit/PC D3D SM6) OpenXR Oculus (1.115.0)
Cup Topple


CarnivalGames Preview [NetMode: Standalone 0] (64-bit/PC D3D SM6) OpenXR Oculus (1.115.0)
Score: 130
Score Card
Ring Toss
Dart Throw
80
Cup Topple
34
Total Score
130
244
57 items


CarnivalGames Preview [NetMode: Standalone 0] (64-bit/PC D3D SM6) OpenXR Oculus (1.115.0)
MAIN ENTRANCE
57 items

7. References

- Unreal Engine Documentation
- Unreal Engine VR Template (OpenXR)
- Unreal Engine Marketplace (Fab) – free and paid assets
- Blender – custom 3D asset creation
- Cosmos Leartes Studios – environment assets
- HOYO-MiX (HoYoverse)– main background music
- Pixabay – free audio effects

8. Plagiarism Declaration

We have read and understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at: http://www.tcd.ie/calendar

We declare that this assignment, together with any supporting artefacts, is offered for assessment as our original and unaided work, except insofar as any advice and/or assistance from any other named person in preparing it and any reference material used are duly and appropriately acknowledged.

Name	Signature	Date
Priyansh Nayak	P. Nayak	4/1/26
Akshit Shrivastava	A. Shrivastava	4/1/26
Viraj Vilas Bhor	V. Bhor	4/1/26
Sarvesh Suresh Jaiswal	S. Jaiswal	4/1/26
Yokesh Muthu Kathiravan	Y. M. Kathiravan	4/1/26