**<h1 class="title">Pack It Perfect!</h1>**

Pack it Perfect is a quirky and challenging VR puzzle game where your objective is to fit a variety of odd-shaped items into a single suitcase. But it’s not just about finding the right space, some items react when placed next to each other. For example, certain objects might expand, melt, or trigger strange interactions depending on what they’re touching. Figuring out how to pack everything without setting off chaos is the real puzzle.

**<h2>About the Game</h2>**

This VR game was made in Unity on my internship at AlterEyes, together with the other interns (2 coders and 1 artist) we had full control of design and made this game from prototype to production and also ready for release. Unfortunaly it isn't released yet but it should be available soon on Meta Quest Store!

**<h2>What I Did</h2>**

During my time on Pack It Perfect, I contributed extensively to both the gameplay systems and the overall technical stability of the project. My work spanned UI/UX design, game mechanics, save systems, and bug fixing, with a strong focus on problem-solving and refactoring for long-term maintainability.

**<h3> Gameplay Systems & Mechanics </h3>**

**<h4> Rule System </h4>**

This system first iteration during prototype was not written by me, but I did refactor and rewrite it during production phase.

Items consists of three types, default (nothing special), Rule Holder (the item that will do something when triggered with the correct trigger type, each rule holder has a Rule scriptable Object attached to it), and the trigger (the item that will trigger the rule in the rule holders if their trigger type is the correct). Trigger types is an Enum that will tell the rile holder rand trigger at which trigger type they should react to, the values could be: cold, fire, predator, ….

Each rule has different ways it needs to react, for example, the expand rule, only needs to react when there is something near it to trigger, so boolean approach, but other items needs to react every time something happens, so like a trigger approach.

We made it in a way where we only need to write the logic of what it needs to do and it will automatically do the approach it is designed to do through a simple boolean with the scriptable Object.

**<h4> Expansion Rule (objects expand in size near specific neighbors) </h4>**

Goal: Certain objects should instantly grow when placed near specific “trigger” objects, doubling their size to create more interesting packing challenges.

The rule system scans for valid the placed object’s neighbours to check if it’s an expansion trigger. When a match is found, the expanding object scales its mesh and colliders proportionally fast to give it a snappy feeling. The scaling logic bypasses Unity’s default transform scaling for some colliders, using manual resizing to avoid hierarchy bugs (especially with nested mesh colliders).

Once the object grows, it runs a bounds check if it’s outside of the suitcase. If any part exceeds those limits, the object is repositioned or ejected out i fit doesn’t find space. If other items overlap the expanded item, those items will be thrown out instead.

<h5> **Blowfish – More Than Just Size** </h5>

**Goal:** Variant of the expansion rule, but with a visual twist. The blowfish “inflates” and changes its mesh to a puffed-up model.

**<h4> Beach Ball – Throw to Grow, Squeeze to Shrink </h4>**

**Goal:** Make players *physically* interact with the beachball for it to change sizes. Throw it up for size increase, compress it with two hands to shrink.

I tracked the ball’s upwards velocity to detect throws, triggering an expansion when it crossed a speed and small time threshold. For shrinking, I measured the distance between the player’s hands while holding the ball. When that distance dropped below a set threshold, the ball smoothly scaled down.

**<h4> Ice Cube – Heat & Cold</h4>**

**Goal:** Ice melts unless kept cold, but heat overrides cold, even if both are nearby.

Each ice cube constantly scans its surroundings for “temperature sources”, these are triggers with heat or cold type. Cold sources stops the melting, so without a cold source nearby it will metl. Hot sources makes the icu cube mellt instantly, even if there is a cold source, so you have to watch out where you place hot objects.

**<h4> Custom Distance Grabber </h4>**

**Goal:** Give players the ability to grab objects from far away.

Even though we used the Auto Hands VR package to handle the hands and grabbing, their Distance grabber wasn’t working the way we needed it to, so based on the Auto hands package, I created a simple ray-based targeting system with a LineRenderer beam to know which item you’re about to grab, haptic feedback on lock, and grab confirmation when the player pressed the grab button. When grabbing the item will fly towards the hand in a fast and satisfying way.

**<h4> UI & Player Experience**

UI Development with Doozy – Learned and implemented the Doozy UI package to create flexible and scalable menus. Built the Level Selector, interactive UI elements, and Completion Menu that displays stars earned after a level.

**<h4> Level Progression & Save Data </h4>**

**Goal:** Keep track of stars earned and lock/unlock levels accordingly

**Implementation:**  
Used the AlterEyes save system to store per-level star counts. Added unique **LevelIDs** to each ScriptableObject so the save data always matches the correct level, even if the list order changes. Grouped levels into “collections” so designers could set star requirements for each group without coding.

**Challenges:**

* Early on, reordering levels erased save progress — fixed permanently with ID system.
* Needed to handle missing/invalid save files gracefully so first-time players could start instantly.

**3D Loading Screen – Immersive Transitions**

**Goal:** Replace flat loading fades with a VR-friendly, immersive transition.

**Implementation:**  
Placed a giant 3D sphere around the player’s camera, animated its opacity for fade in/out, and swapped scenes during the blackout moment. This prevents motion sickness by keeping the player’s perspective consistent.

**Challenges:**

* The sphere had to be perfectly centered on the camera at all times — solved with a LateUpdate position lock.
* Had to prevent the player from seeing “outside” the sphere before fade completed.

**Bug Fixes & Refactoring – The Unseen Hero Work**

**Goal:** Keep the game stable and maintainable while adding new features.

**Implementation:**

* Fixed merge conflict disasters that broke prefabs, lost script references, and reset assets.
* Refactored the rule/trigger framework to separate logic types for easier debugging.
* Adjusted dependency injection setup for more reliable scene loading.

**Challenges:**

* VR bugs are often hard to reproduce — I built custom debug tools to log headset position, object states, and recent interactions for easier diagnosis.
* Balancing feature work with refactoring without slowing the team down.

If we publish this **exact level of detail** on your portfolio, it’s going to make it very clear that you:

1. Understand the design intention of each feature.
2. Can explain *exactly* how you implemented it.
3. Know where the pain points were and how you solved them.

[**https://www.gamesjobsdirect.com**](https://www.gamesjobsdirect.com)

[**https://www.flega.be/members**](https://www.flega.be/members)

[**https://gamedevmap.com**](https://gamedevmap.com)

[**https://www.eventsforgamers.com**](https://www.eventsforgamers.com)

**Onderwerp: Sollicitatie Unity Gameplay Programmer – [Jouw Naam]**

Beste **[Naam van de recruiter / contactpersoon],**

Met veel enthousiasme solliciteer ik naar de functie van Unity Gameplay Programmer bij **[Bedrijfsnaam].** De combinatie van mijn ervaring met Unity en C# en mijn passie voor het ontwikkelen van meeslepende gameplay-ervaringen maakt dat ik ervan overtuigd ben een waardevolle bijdrage te kunnen leveren aan uw team.

Tijdens mijn eerdere projecten heb ik gewerkt **aan [noem 1–2 voorbeelden, bv. character controllers, AI-systemen, physics-interacties of UI]**. Daarbij heb ik zowel zelfstandig als in multidisciplinaire teams gewerkt, waarbij ik gewend ben om nauw samen te werken met designers en artists om gameplay-ideeën te vertalen naar functionele en plezierige features.

Wat mij aanspreekt **in [Bedrijfsnaam]** is **[noem iets specifieks, bv. de focus op innovatieve gameplay, het genre van hun games of hun reputatie].** Ik ben ervan overtuigd dat mijn technische vaardigheden en creatieve insteek goed aansluiten bij de visie van jullie studio.

In de bijlage vindt u mijn CV en een link naar mijn portfolio: [link naar GitHub/portfolio]. Ik licht mijn ervaring en motivatie graag verder toe in een persoonlijk gesprek.

Alvast hartelijk dank voor uw tijd en overweging. Ik kijk ernaar uit om van u te horen.

Met vriendelijke groet,

[Voornaam Achternaam]  
[Telefoonnummer]  
[E-mailadres]  
[Portfolio/GitHub/LinkedIn link]

Onderwerp: Sollicitatie Unity Gameplay Programmer – [Jouw Naam]

Beste [Naam van de recruiter / contactpersoon],

Met veel enthousiasme solliciteer ik naar de functie van Unity Gameplay Programmer bij [Bedrijfsnaam]. Onlangs ben ik afgestudeerd aan Howest – Digital Arts & Entertainment, waar ik mij heb gespecialiseerd in game development en gameplay programmering. Tijdens mijn opleiding en stage heb ik een sterke basis opgebouwd in Unity en C#, gecombineerd met ervaring in het ontwikkelen van boeiende en technisch stabiele gameplay-systemen.

Tijdens mijn internship bij AlterEyes heb ik meegewerkt aan VR-projecten, waarbij ik zowel gameplay-features als technische optimalisaties heb gerealiseerd. Deze ervaring heeft me niet alleen mijn programmeervaardigheden verder laten ontwikkelen, maar ook geleerd hoe belangrijk samenwerking tussen disciplines is om creatieve ideeën om te zetten in een kwalitatieve game-ervaring.

Wat mij aanspreekt in [Bedrijfsnaam] is [noem iets specifieks: bv. jullie innovatieve aanpak, focus op immersieve gameplay of het type games dat jullie ontwikkelen]. Ik ben ervan overtuigd dat mijn achtergrond, gecombineerd met mijn passie voor gameplay programmering, goed aansluit bij de visie van uw team.

In de bijlage vindt u mijn CV en een link naar mijn portfolio: [link naar GitHub/portfolio]. Ik licht mijn ervaring en motivatie graag verder toe in een persoonlijk gesprek.

Alvast hartelijk dank voor uw tijd en overweging. Ik kijk ernaar uit om van u te horen.

Met vriendelijke groet,

[Voornaam Achternaam]

[Telefoonnummer]

[E-mailadres]

[Portfolio/GitHub/LinkedIn link]