BUBO

Bubo is a fantasy-themed VR adventure game where players embody an owl tasked with mail delivery. While navigating environmental hazards (storms and volcanoes) with intuitive flight mechanics and dynamic challenges, players must balance speed and precision to ensure safe package delivery and maximize rewards.

Made with

Unity · C# · blender · Ps

Design & Code by

Erin Hao

GitHub

https://github.com/SugaCube777/Bubo-VR

Video

https://vimeo.com/1042411228?share=copy#t=0

Bubo – an owl VR simulation game

Design

Game Play

Players take on the role of an owl and fly to deliver mail while avoiding hazards and managing resources like energy and time.

Game Loop

- Accept Mail
- Fly to Alternative Destination
- Fly to Destination
- Navigate Hazards
- Deliver Mail
- Receive Rewards
- Upgrade Gears

Systems & Mechanics Player Fly Behaviour

- Unlike traditional teleportation navigation in VR games, players control the main character by simulating bird behaviors.
- Players flap their wings to ascend and perform other actions such as gliding, tilting, and descending with intuitive wing movements.

Environment System

The environment introduces dynamic hazards that impact gameplay, including:

- Rain: Increases weight, causing a speed debuff and potential damage to packages.
- Lightning: Strikes during storms, leading to loss of control and a 2–4 second fall, damaging certain packages.
- Updrafts: Provide a constant upward force, reducing weight and increasing speed.
- Geyser Eruptions: Create turbulence within range, potentially damaging packages.
- Electricity Towers: Cause instant mission failure upon collision.
- Electromagnetic Interference: Leads to a 10-second disorientation, disabling the directional guide after falling to the ground.

Delivery System

- Sender: The origin of a package.
- Receiver: The recipient of a package, which can vary based on the narrative and include multiple recipients.
- Quest: Each package is tied to a quest, guiding players and driving narrative progression.
- Inventory & Packages: Players store packages in an inventory, each with attributes such as weight, time sensitivity or fragility. A package's condition is affected by time and hazards, ultimately influencing the reward.

Respawn

Other Mechanics

- Characters: Interactable agents that can send or receive packages, initiate dialogues, and trigger effects.
- Dialogues: Display dialogue UI with AI-generated cinematic voiceovers for enhanced immersion.

Bubo – an owl VR simulation game

DESIGN World Map / Level Design

Island & Tree House Site

River Cave & Poacher's Hut

Magic Academy & Town

Ancient Forest & Herbalist's Garden

Grassy Field & illegal hunting site

Map Key

- 🟢 Easy Area I: Windy & Drizzling – Slows player flight speed.
- 🟡 Easy Area II: Fences, Railheads & Electric Wires – Obstacles to navigate.
- 🔵 Medium Area I: Geysers – Risk of being lifted above clouds, harmful to live animal packages.
- 🔴 Medium Area II: Lightning Storms – Requires dodging lightning strikes.
- 🟣 Hard Area I: Oceanic Storms – Heavy rain, strong wind, and lightning, hazardous for flight and packages.
- 🟠 Hard Area II: Volcanic Lava – Severe threat to package safety.

Bubo – an owl VR simulation game
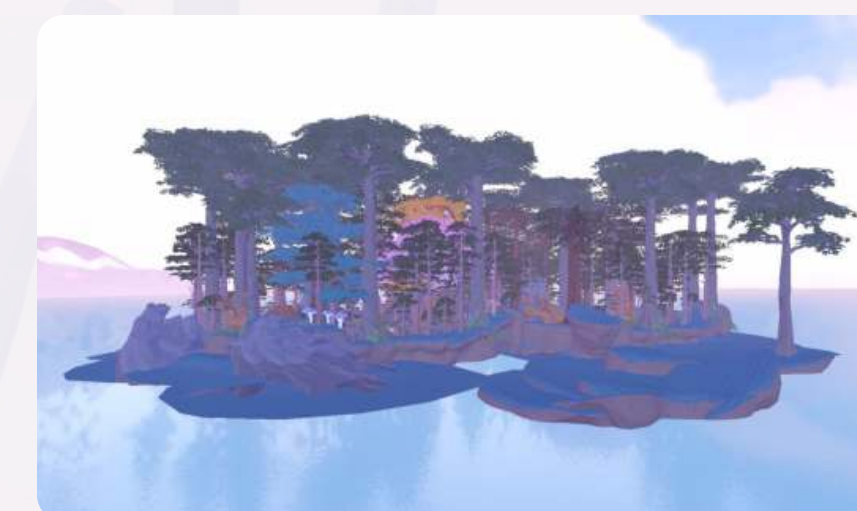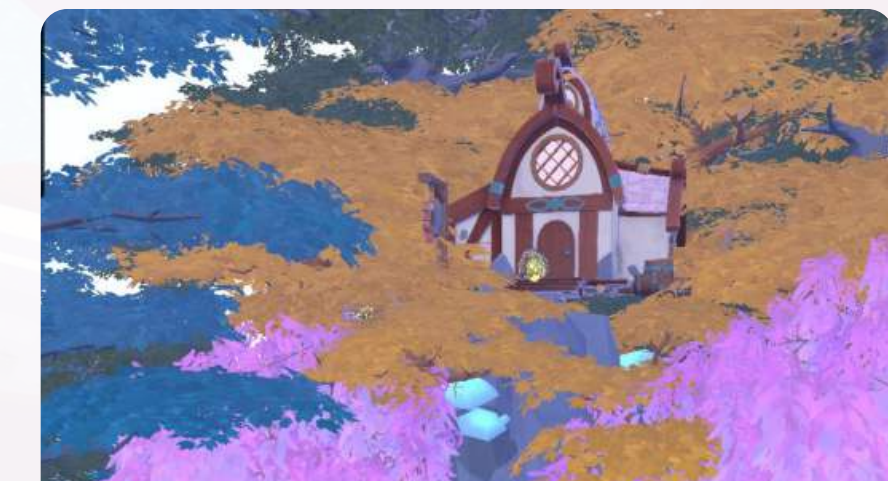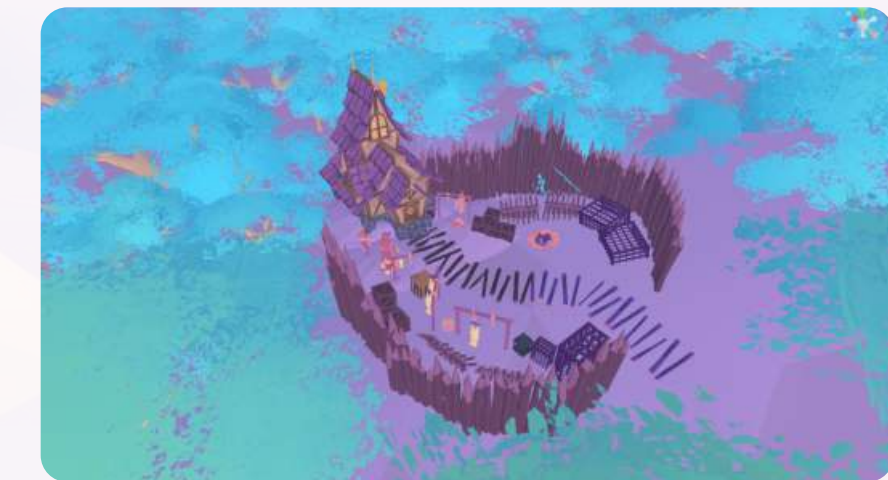
DESIGN Quests

Quests I Back to school Letters

Description + Sender:
- Sent from Magic Academy office
- Letters for students to prepare for the upcoming school year, and admission letter for new student

Recipients
- Professor's house
- New prefect and her best friend (light drizzle)
- New student (windy)

Delivery Items
- 3 Back to school letters & Prefect badge, 1 admission letter (vulnerable to water, no time limit)

Dialogue 1:
- "The Prefect! you've been made new Prefect, I knew it!"
- "But it would be a lot on top of my exams."
- "Oh you're bragging. Seems like I can get away with some rule breakings can't I?"
- "Don't you dare, I'll turn you in to professor!"

Dialogue 2:
- "Let's see, if you are taking herbology this year, you won't be able to take my class"
- "Come on mum, I don't want to be in your class anyways, you're going to give me a Pass regardless"
- "Alright, but do choose your electives according to your future direction. sleep early, we have shopping to do tomorrow"

Dialogue 3:
- "An admission letter? addressed to our house."
- "Is that what I think it is?"
- "Read it read it!"
- "Dear Parents, we are pleased o inform you that..."

Quests III Poacher's Business

Description + Sender:
Sent from illegal hunting sites
(Grassy field, River cave, Tree house)
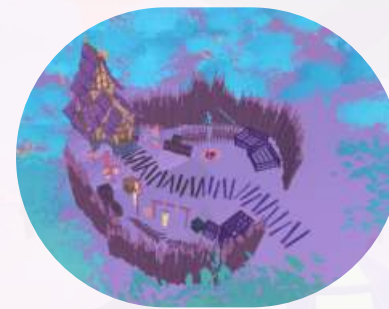
They definitely belong back at their home, but they are worth a lot. They might have been in capture for a long time and need nurturing, maybe if someone could give proper care......

Recipients
- Poacher's shack (Geyser region)
- School Creature Care (Stormy path with lightning hazard)

Delivery Items
Ginkgo lynx, Singing salamander, Glowing parakeet egg (vulnerable to altitude over cloud)

Dialogue 1:
- "My lucky animals! this one for the dark Order, this one good for a coat, ah who would refuse a pelt like this? This one from Albania ---"
- "Who would want this noisy thing"
- "Mind your own business or I'll turn you into one"

Dialogue 2:
- "Poor creatures! Are they from poachers? "
- "Leave them to me, I'll make sure to take good care of them, don't you worry"

Quests IV Urgent Scholarly Exchange

Description + Sender:
Sent from Herbalist's Garden
This is a important recipe and related herbs for dragon pox cure.

Recipients
School potionare (volcanic region with Lava, Geyser, and Rain)

Delivery Items
Ancient herbal book, fresh & dried herbs (vulnerable to both heat and water)

Dialogue (sender):
- "Make sure these herbs get to the potionare safely"
- "Oh and don't forget to pick up the enchanted flower from the garden! "
- "Come around for tea again next time! "

Dialogue (receiver):
- "Look the owl has returned with post!"
- "Must be the recipe for the currr-!!"
- "It's called a cure! C U R E, now could you move away and let her in please?"

ART

Visual Style

A stylized 3D fantasy theme with vibrant and immersive visuals.

3D Scene

A seamless fantasy world featuring castles and villages with detailed indoor design, forests, volcanoes, beaches, mountains, and other diverse landscapes.

Animation Rigging

Chain IK constraints are utilized to achieve realistic wing movements for the player model.

3D Characters

The player model consists of a first-person rigged wing system, with movement mapped to the wings for realistic and expressive gestures. NPCs are fantasy-themed and include detailed animations.

UI Design

Minimalist, fantasy-themed design with retro-style patterns and textures.

Sound Design

- **Sound Style:** Epic medieval fantasy music that enhances the immersive experience.
- **Dialogues:** AI-generated cinematic voice overs with British accents, perfectly matching the game's thematic setting.

REPLICA

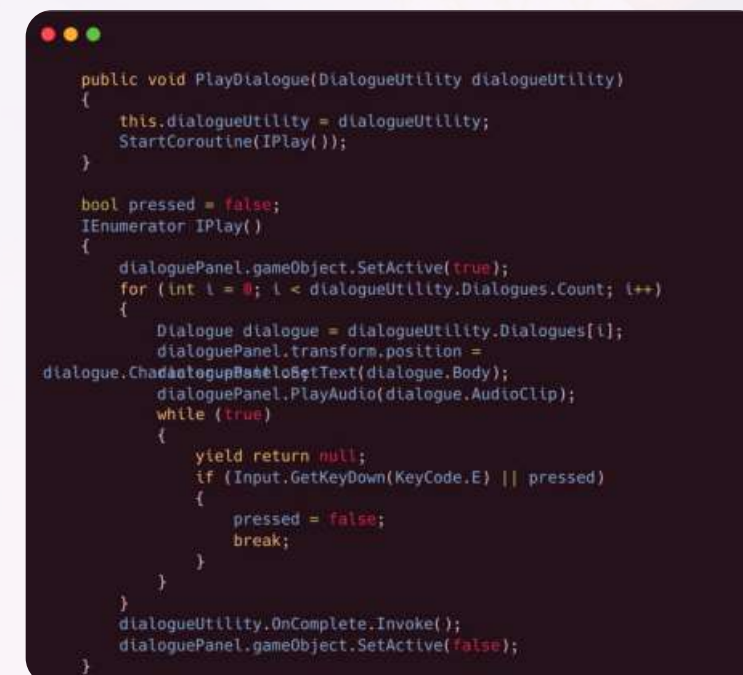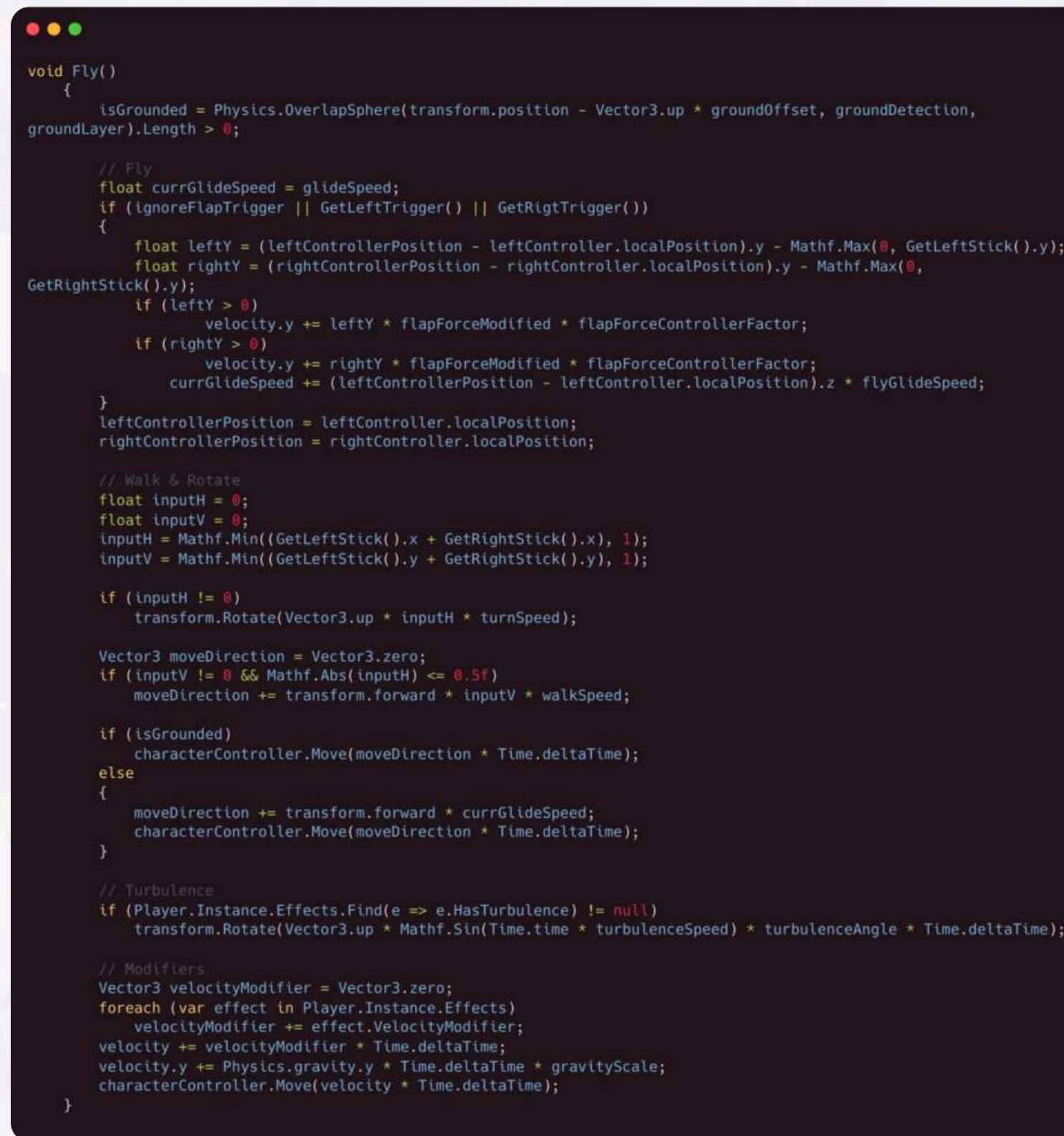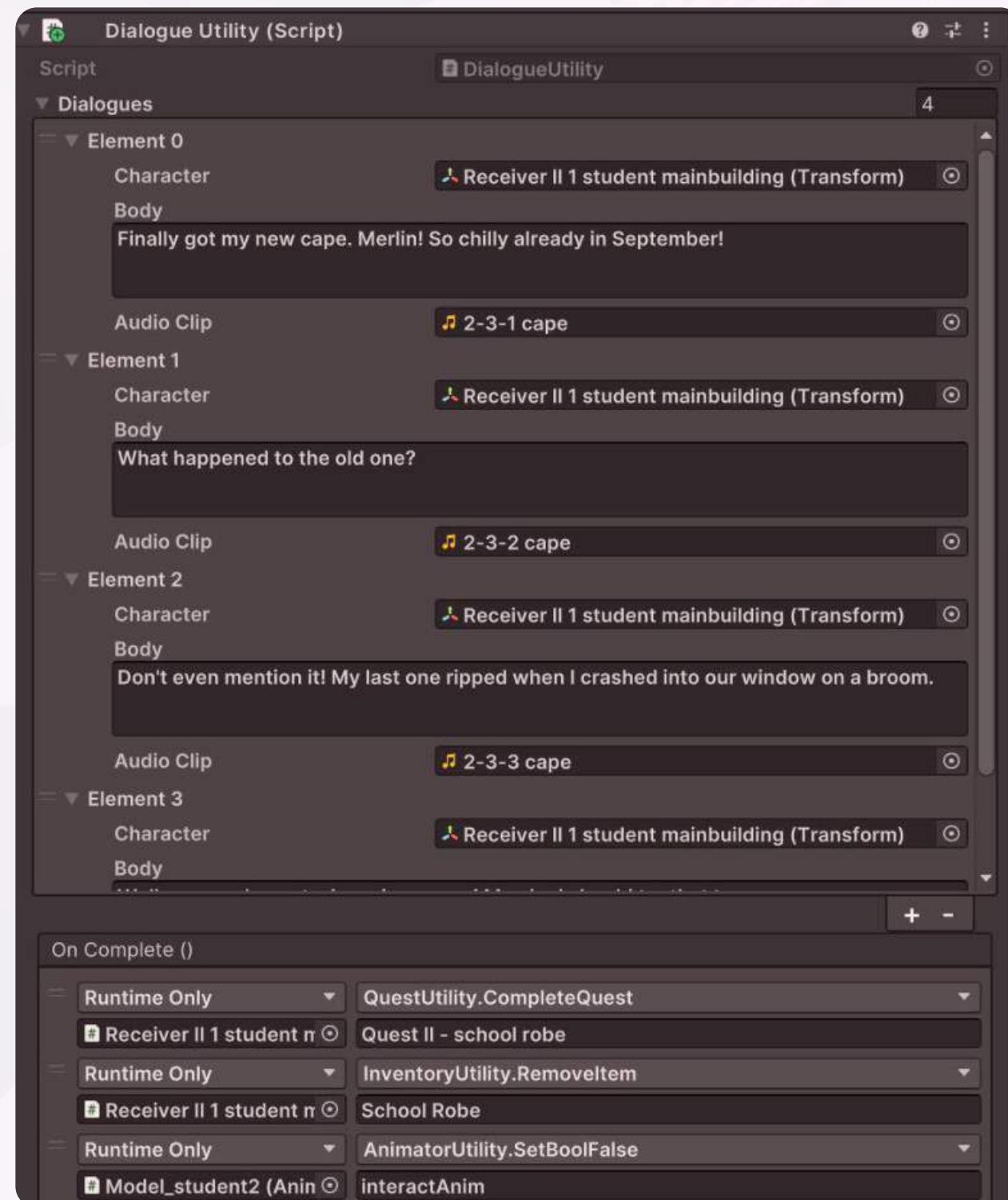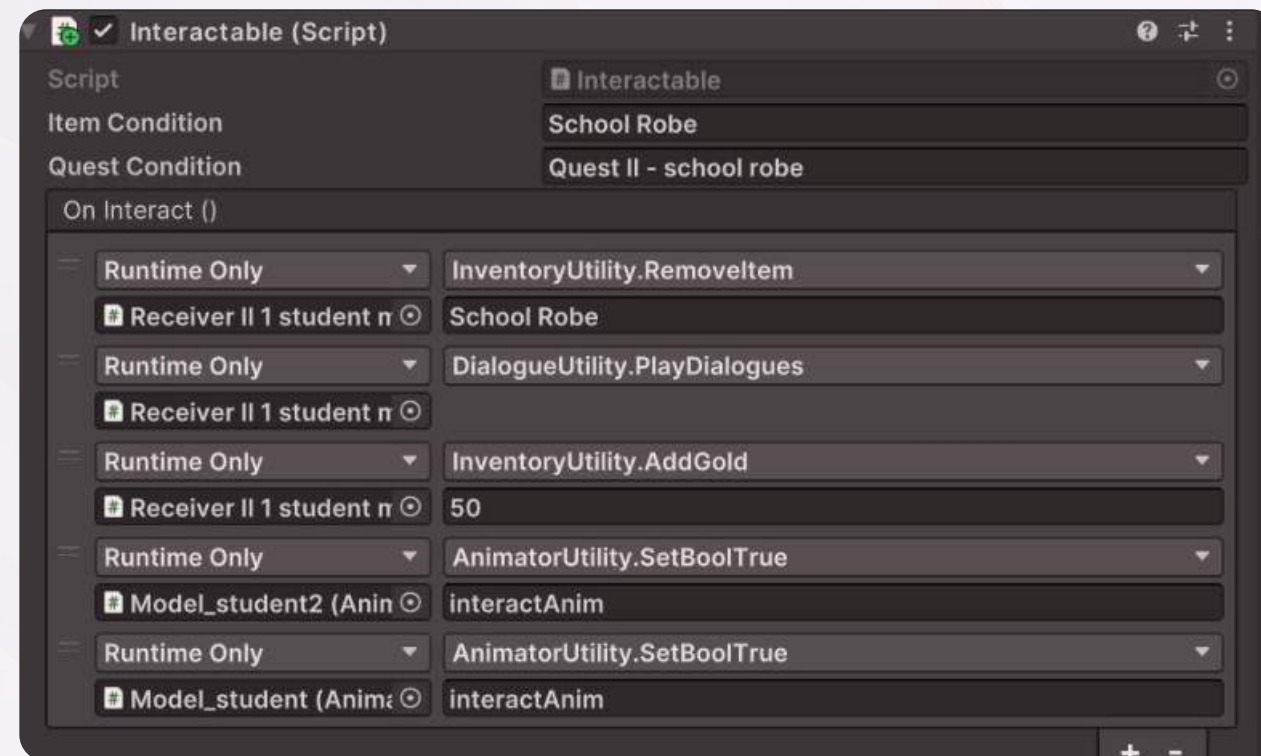*Bubo — an owl VR simulation game*

Tech

Event-Driven Programming

Utilizes UnityEvent and UnityAction, following the observer design pattern, to manage interactables. Scripted utility components enable drag-and-drop actions, allowing for prototyping and rapid level building and without the need for additional scripting for each interactable.

Interactable (Script)

Script	Interactable
Item Condition	School Robe
Quest Condition	Quest II - school robe

On Interact ()

- Runtime Only — InventoryUtility.RemoveItem — Receiver II 1 student m ⊙ School Robe
- Runtime Only — DialogueUtility.PlayDialogues — Receiver II 1 student m ⊙
- Runtime Only — InventoryUtility.AddGold — Receiver II 1 student m ⊙ 50
- Runtime Only — AnimatorUtility.SetBoolTrue — Model_student2 (Anin ⊙ interactAnim
- Runtime Only — AnimatorUtility.SetBoolTrue — Model_student (Anima ⊙ interactAnim

Dialogue Utility (Script)

Script	DialogueUtility
Dialogues	4

Element 0
- Character: Receiver II 1 student mainbuilding (Transform)
- Body: Finally got my new cape. Merlin! So chilly already in September!
- Audio Clip: 2-3-1 cape

Element 1
- Character: Receiver II 1 student mainbuilding (Transform)
- Body: What happened to the old one?
- Audio Clip: 2-3-2 cape

Element 2
- Character: Receiver II 1 student mainbuilding (Transform)
- Body: Don't even mention it! My last one ripped when I crashed into our window on a broom.
- Audio Clip: 2-3-3 cape

Element 3
- Character: Receiver II 1 student mainbuilding (Transform)
- Body:

On Complete ()

- Runtime Only — QuestUtility.CompleteQuest — Receiver II 1 student m ⊙ Quest II - school robe
- Runtime Only — InventoryUtility.RemoveItem — Receiver II 1 student m ⊙ School Robe
- Runtime Only — AnimatorUtility.SetBoolFalse — Model_student2 (Anin ⊙ interactAnim

```
void Fly()
{
    isGrounded = Physics.OverlapSphere(transform.position - Vector3.up * groundOffset, groundDetection,
    groundLayer).Length > 0;

    // Fly
    float currFlapSpeed = glideSpeed;
    if (ignoreFlapTrigger || GetLeftTrigger() || GetRigtTrigger())
    {
        float leftY = (leftControllerPosition - leftController.localPosition).y - Mathf.Max(0, GetLeftStick().y);
        float rightY = (rightControllerPosition - rightController.localPosition).y - Mathf.Max(0,
        GetRightStick().y);
        if (leftY > 0)
            velocity.y += leftY * flapForceModified * flapForceControllerFactor;
        if (rightY > 0)
            velocity.y += rightY * flapForceModified * flapForceControllerFactor;
        currGlideSpeed += (leftControllerPosition - leftController.localPosition).z * flyGlideSpeed;
    }
    leftControllerPosition = leftController.localPosition;
    rightControllerPosition = rightController.localPosition;

    // Walk & Rotate
    float inputH = 0;
    float inputV = 0;
    inputH = Mathf.Min((GetLeftStick().x + GetRightStick().x), 1);
    inputV = Mathf.Min((GetLeftStick().y + GetRightStick().y), 1);

    if (inputH != 0)
        transform.Rotate(Vector3.up * inputH * turnSpeed);

    Vector3 moveDirection = Vector3.zero;
    if (inputV != 0 && Mathf.Abs(inputH) <= 0.5f)
        moveDirection += transform.forward * inputV * walkSpeed;

    if (isGrounded)
        characterController.Move(moveDirection * Time.deltaTime);
    else
    {
        moveDirection += transform.forward * currGlideSpeed;
        characterController.Move(moveDirection * Time.deltaTime);
    }

    // Turbulence
    if (Player.Instance.Effects.Find(e => e.HasTurbulence) != null)
        transform.Rotate(Vector3.up * Mathf.Sin(Time.time * turbulenceSpeed) * turbulenceAngle * Time.deltaTime);

    // Modifiers
    Vector3 velocityModifier = Vector3.zero;
    foreach (var effect in Player.Instance.Effects)
        velocityModifier += effect.VelocityModifier;
    velocity += velocityModifier * Time.deltaTime;
    velocity.y += Physics.gravity.y * Time.deltaTime * gravityScale;
    characterController.Move(velocity * Time.deltaTime);
}
```

```
public void PlayDialogue(DialogueUtility dialogueUtility)
{
    this.dialogueUtility = dialogueUtility;
    StartCoroutine(IPlay());
}

bool pressed = false;
IEnumerator IPlay()
{
    dialoguePanel.gameObject.SetActive(true);
    for (int i = 0; i < dialogueUtility.Dialogues.Count; i++)
    {
        Dialogue dialogue = dialogueUtility.Dialogues[i];
        dialoguePanel.transform.position =
dialogue.Character.transform.position;
        dialoguePanel.SetText(dialogue.Body);
        dialoguePanel.PlayAudio(dialogue.AudioClip);
        while (true)
        {
            yield return null;
            if (Input.GetKeyDown(KeyCode.E) || pressed)
            {
                pressed = false;
                break;
            }
        }
    }
    dialogueUtility.OnComplete.Invoke();
    dialoguePanel.gameObject.SetActive(false);
}
```

```
[System.Serializable]
public class Effect
{
    public string ID = "Effect";
    public float Duration = 0; // s
    public bool IsUnique = false;

    // Fly
    public float FalpForceModifer = 0; // %
    public bool BlockControl = false;
    public Vector3 VelocityModifier =
Vector3.zero;
    public bool HasTurbulence = false;

    // Inventory
    public float WeightModifer = 0; // %
    public float SoundnessDamage = 0; // /s

    // UI
    public bool BlockNavigationUI = false;
}
```

Scripts

Game Management
GameManager.cs
Effect.cs

Player
Player.cs
PlayerFlyBehaviour.cs
PlayerRespawnPoint.cs

Interactables & Characters
Interactable.cs
InteractableManager.cs
Character.cs
CharacterManager.cs
DialogueManager.cs
DialoguePanel.cs
DialogueUtility.cs

Misc
AnimatorUtility.cs
UIBlackscreen.cs

Inventory System
Inventory.cs
InventoryPanel.cs
InventoryUtility.cs
InventoryPanelItem.cs
ItemConfigs.cs

Environment System
EnvironmentBuffer.cs
EnvironmentGeyser.cs
EnvironmentJam.cs
EnvironmentLightning.cs
EnvironmentRain.cs
EnvironmentVolcano.cs

Quest System
QuestArrow.cs
QuestConfigs.cs
QuestManager.cs
QuestUtility.cs

Bubo — an owl VR simulation game

Development

Conceptualization

How to train your dragon: Borrowing the thrill and exhilaration of flight mechanics, the game recreates the excitement of soaring through the skies.

Harry Potter, Fantastic Beast and where to find them: Enriched with magical fantasy world-building, the game features a creative narrative with fictional creatures, adding depth and wonder to its universe.

Sky Children of Light: Incorporating an explorative and almost meditative gameplay experience, the game offers players a chance to connect with the serene and enchanting aspects of its world.

AI Generated Sprites

Organized Quest to Build Narrative

Interactive Design

Thumbstick to quick switch to next tracking item

Details about Package

package dmg level (water / heat)

Y to call out inventory
X to close
A to view detail

Player UI -- in flight

time — Tracking to — money

Inventory snaps to righthand side when called out

speed
height

wind direction?

South East

Package #

Receiver: character, mail box, chimney, etc.

3D cursor?

Tracking to Package 1

Drag & drop?
L/R grip

Total Weight : 6.3 / 20

Tracking New Package

currency + xxxx

Text text text

Total Weight : 2.9 / 20

Quest Flow Drafts

Actual Interaction in Game

Flight Experience

Bubo — an owl VR simulation game