Eating Prototype

Time to complete: **X weeks**

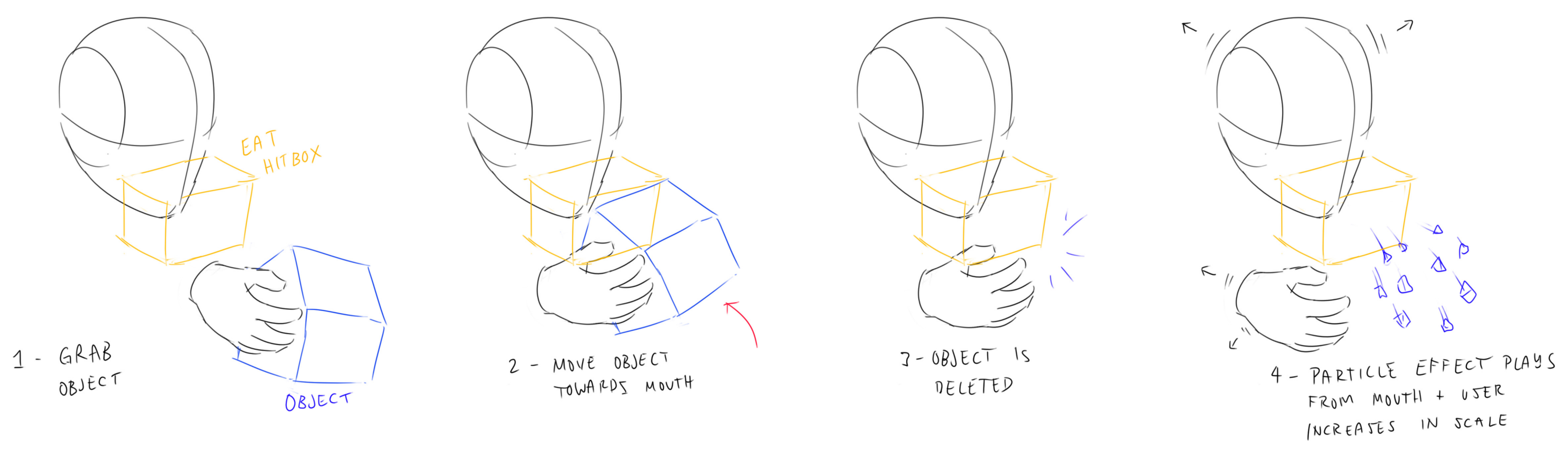
Goals

- Allow users to move grabbable objects into an "eating" hitbox attached to the user's headpose, triggering the scaling effect.
- Evaluate how the experience feels when the separate elements are combined together.

Description

The core feature behind this experience is allowing users to 'eat' the objects that they're able to grab. Any object that can be grabbed can also be eaten. To 'eat' an object, the user must simply move the object into their headpose, deleting the object, triggering a simple particle effect from the user's 'mouth', and increasing their scale by a set amount depending on the object eaten.

Visual Reference


EAT HITBOX
1 - GRAB OBJECT
OBJECT
2 - MOVE OBJECT TOWARDS MOUTH
3 - OBJECT IS DELETED
4 - PARTICLE EFFECT PLAYS FROM MOUTH + USER INCREASES IN SCALE

User Flow

The order of actions the user should be able to follow:

1. User starts in level 1 of the mockup environment
2. They find a grabbable object in their environment
3. They move the object into their mouth
4. The object is eaten (Removed)
   a. A particle effect plays from the user's mouth
   b. Their scale increases by a determined amount depending on the object eaten and their current scale amount
5. They are able to grab + eat new objects with their larger scale
6. The user continues to eat objects until they eat everything in the prototype

Adjustable Elements

To best test this prototype, we should be able to adjust certain elements:

- The scale + position of the eat hitbox
- The scale value increase for each object
- All adjustable elements from previous prototypes
- Scaling restrictions

Developer Recommendations

-

Game feel

Forms of feedback that will make the interaction feel more satisfying and will better help us answer our prototype question:

- Eating SFX
    - Crunching sound
    - Chewing sound
    - Om nom nom sound
- A particle effect from the user's mouth

Potential Directions

Predictions for how this mechanic could be expanded upon to improve the experience:

- Explore

Testing Instructions

Where?

`wp-the-hungry-one--growing : 154` in `8-GrabbableObjectsPrototype`

The scene combines eating functionality with previous prototypes.

Input

Trigger press to grab an object

Touchpad touch to activate teleport

Touchpad press to teleport

For debugging, only works if the Level Configurations scale amount is not set to 0 in the settings:

B to grow
A to shrink
Left menu to reset the scale

In order to adjust the eating range outside of play mode:

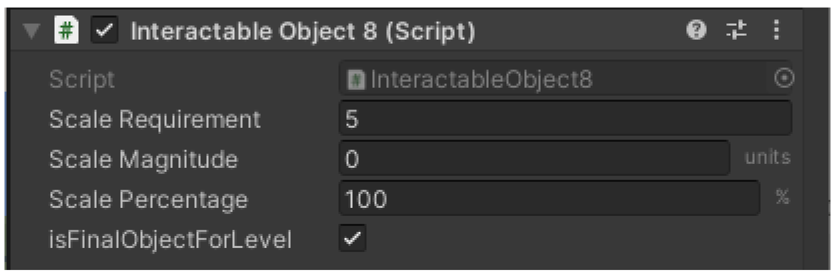
- Go to the scene hierarchy in Unity.
- Go to `VRTK_SDKManager → VRTK_SDKSetup → Oculus` and enable the Oculus object. It will allow you to see the eating range (defined by a cube attached to the user's head).
- Go to `VRTK_SDKSetup → Oculus → OVRCameraRig → CenterEyeAnchor` and find the `EatHitbox` object in the scene
- Adjust `Transform.Scale` values for `EatHitbox` as you see fit
    - If you adjust `Transform.Position`, note that the position will change relative to the headset

Eating range is defined by a transparent purple cube.

NEW: In order to set Interactable Object as the final object for the scene:

This is an important addition since implementation of a scale cap.

If you want the object to be the final object for level, make sure the property `isFinalObjectForLevel` is on like this:



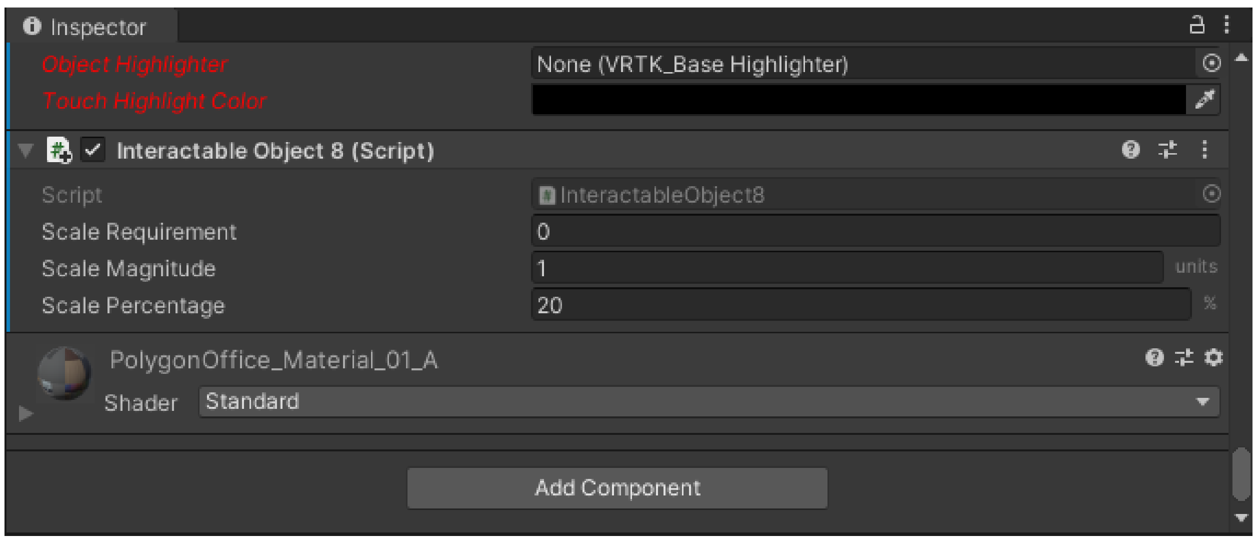
Interactable Object 8 (Script)

Script	InteractableObject8
Scale Requirement	5
Scale Magnitude	0 units
Scale Percentage	100 %
isFinalObjectForLevel	✓

Note that the **scale requirement for the final object = scale cap for the level.**

In order to modify the amount the user is scaled by when they eat an object:

- Go to the object in the scene.
- Find `InteractableObject8` script attached to the object as a component.



Object Highlighter: None (VRTK_Base Highlighter)
Touch Highlight Color

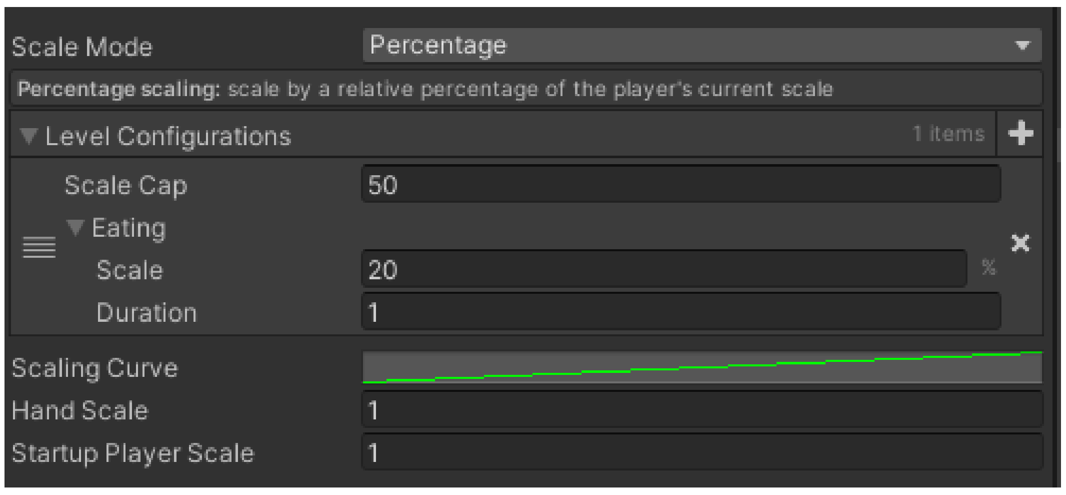
Interactable Object 8 (Script)
Script: InteractableObject8
Scale Requirement: 0
Scale Magnitude: 1 units
Scale Percentage: 20 %

PolygonOffice_Material_01_A
Shader: Standard

- Modify scale magnitude and/or scale percentage. Either value will be used depending on which **scale mode** you chose.
- Note that the **changes to the values in the script cannot be saved while you are in play mode**. You have to stop playing the scene and modify these values outside of play mode.

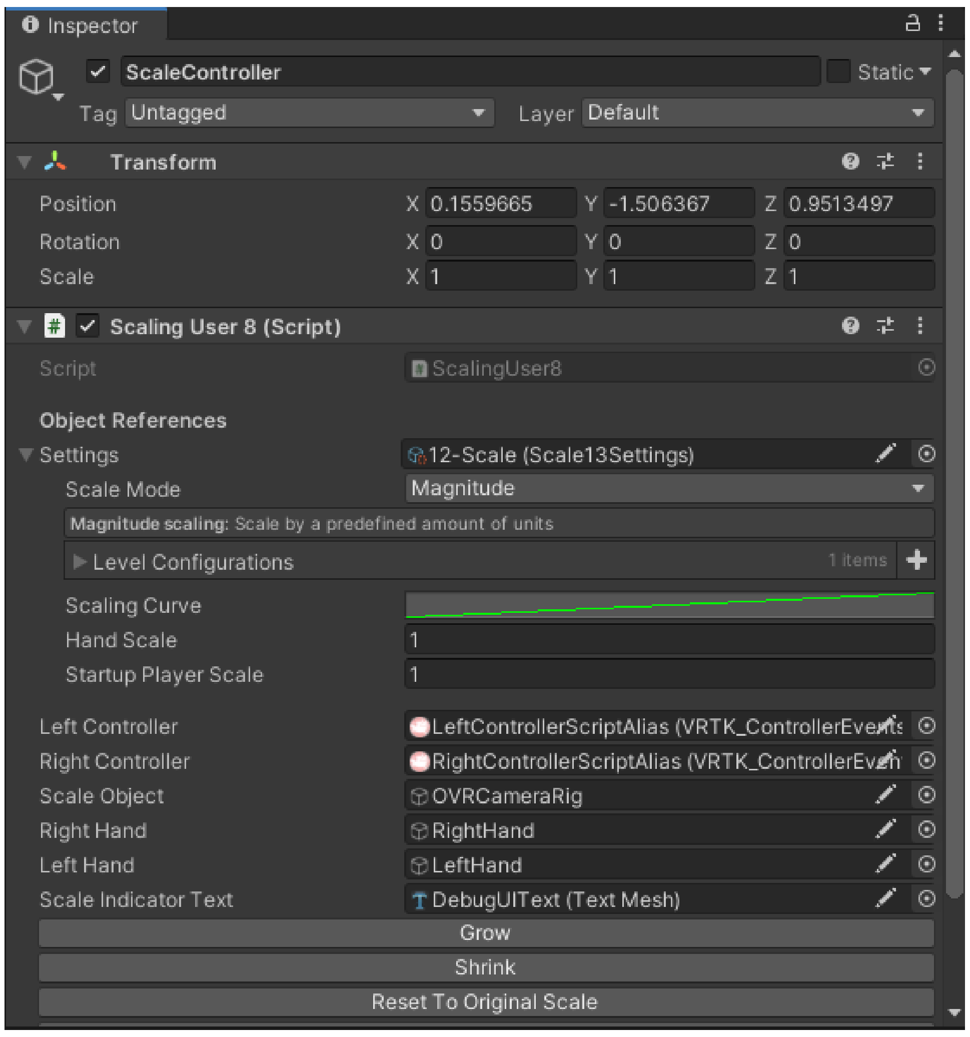
In order to test different scale modes (a set amount vs percentage of user's scale):

- Go to `12-Scale` Settings (located under `Settings → Scale13` or under `8-GrabbableObjects` in Wonder window)



Scale Mode: Percentage
Percentage scaling: scale by a relative percentage of the player's current scale
Level Configurations — 1 items
Scale Cap: 50
Eating
Scale: 20 %
Duration: 1
Scaling Curve
Hand Scale: 1
Startup Player Scale: 1

- Modify `Scale Mode`. You can switch between scale modes during playing.
- Note that a **quick way to check which settings are used for the scene** is to:
  - Find `ScaleController` object in the scene
  - Check which settings are inputted in `ScalingUser8 → ObjectReferences → Settings`
  - You can modify the settings used for the scene by inputting your own settings in that field (make sure your settings are derived from Scale13! You can check if that's the case in the Wonder window)



```
ScaleController                               Static
Tag Untagged                    Layer Default

Transform
Position              X 0.1559665   Y -1.506367   Z 0.9513497
Rotation              X 0           Y 0           Z 0
Scale                 X 1           Y 1           Z 1

Scaling User 8 (Script)
Script                ScalingUser8

Object References
Settings              12-Scale (Scale13Settings)
  Scale Mode          Magnitude
  Magnitude scaling: Scale by a predefined amount of units
  Level Configurations                          1 items
  Scaling Curve
  Hand Scale          1
  Startup Player Scale 1

Left Controller       LeftControllerScriptAlias (VRTK_ControllerEvents
Right Controller      RightControllerScriptAlias (VRTK_ControllerEven
Scale Object          OVRCameraRig
Right Hand            RightHand
Left Hand             LeftHand
Scale Indicator Text  DebugUIText (Text Mesh)
```

What are Level Configurations used for in the settings?

- Level Configurations were used to adjust **scale amount per level** for normal growth (when you eat an object) and growth spurt growth (when you reach a certain size and want to grow by a much larger amount to transition between levels)
- We are transitioning to **adjusting scale per object instead of per level.**

How can Level Configurations still be used?

- `Level Configurations → Eating` is only used for growing with button presses on your controller. (debug purposes only)
- `Level Configurations → Growth Spurt` can be used as normal
  - It will check if the user reached a certain scale threshold, and will scale the user up by a specified scale amount and for a specified duration
- If you want to use either of the values, **make sure Scale is not set to 0.**

Testing

Reviewed by @Quinn MacDonald @Peter Kao 02 03 2021

- From testing, we've found that it will be the most satisfying to base the level transitions off of eating particular objects rather than based off of them reaching particular scales
  - **Feature added — Scale Restrictions**
    - **We want to restrict the player's scale for each level until they've eaten the 'final object' in each level. For example, the user shouldn't be able to surpass a scale of '5' during level 1, until they've eaten the largest object in level (the bed).**

Reviewed by @Anya Leonova 02 04 2021

- I love that we have so many things in the house that we can eat, it's great
- Was wondering why I couldn't eat the clouds when I was on the same eye level with them
- Some cars couldn't be grabbed
- Cool feeling when I went to grab a car and accidentally grabbed a whole building (I had no idea that building was grabbable)

- I wanna eat my house :D (eating just the roof was also great, but left me wanting more)
- It would be fun to have custom sounds for eating certain objects (I remember how you would hear people screaming in Katamari if you were big enough to "grab" the houses and buildings)
- Feeling of accomplishment and amazement when I ate the moon and EVERYTHING disappeared

Reviewed by @Peter Kao @Quinn MacDonald 02 11 2021

- Smooth locomotion felt better — implement smooth locomotion properly so it scales with the user and doesn't have collision
- Footsteps SFX would add a lot for giving the user a sense of scale — The SFX should scale with the user in its pitch and reverb
- Remove rocks from under the bridge — Don't position edible objects underneath other objects
- Animated objects will bring a lot of life to the environment — Driving cars, chimney smoke, etc.
- Reward at the end will be important for production — Scaling up into the universe

Iterations

- [x] ~~Create a hitbox for eating detection~~

▼ `wp-the-hungry-one--growing : 90` at `9-EatHitbox`

1. The hitbox is rendered with a transparent purple material, which makes it easy to see in the scene
2. Hitbox is located under VRTK_SDKSetup → Oculus → OVRCameraRig → CenterEyeAnchor → **EatHitbox**

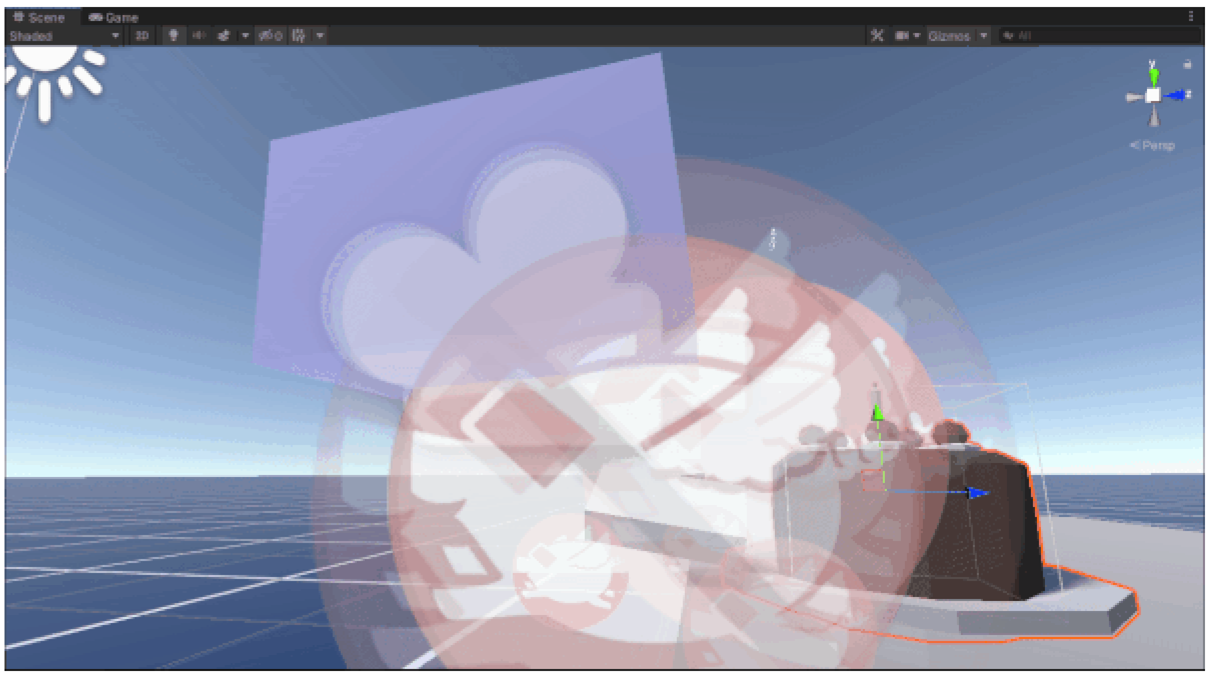
In order to adjust the hitbox outside of play mode:

- Enable `Oculus` object under `VRTK_SDKSetup → Oculus`
- Find `EatHitbox` object in the scene (under `VRTK_SDKSetup → Oculus → OVRCameraRig → CenterEyeAnchor`)
- Adjust Transform values for `EatHitbox` as you see fit

- [x] ~~Detect when the grabbed object is in the eat hitbox (corresponds to eating)~~

▼ `wp-the-hungry-one--growing : 91` at `9-EatHitbox`

An object gets deleted when it's eaten. The feature was added to facilitate visual feedback in the game.



- [x] ~~Delete the object and scale the user up when the object is eaten~~

▼ `wp-the-hungry-one--growing : 93` at `10-EatAndScale`

- [x] ~~Add different scale amounts for each object that can be eaten (including choice between percentage and magnitude)~~

▼ `wp-the-hungry-one--growing : 117-122, 126-129` in `12-AdjustScalePerObject` and `130` in `8-GrabbableObjectsScene`

Each object is modifiable through `InteractableObject` script that is attached to every interactable object in the scene.

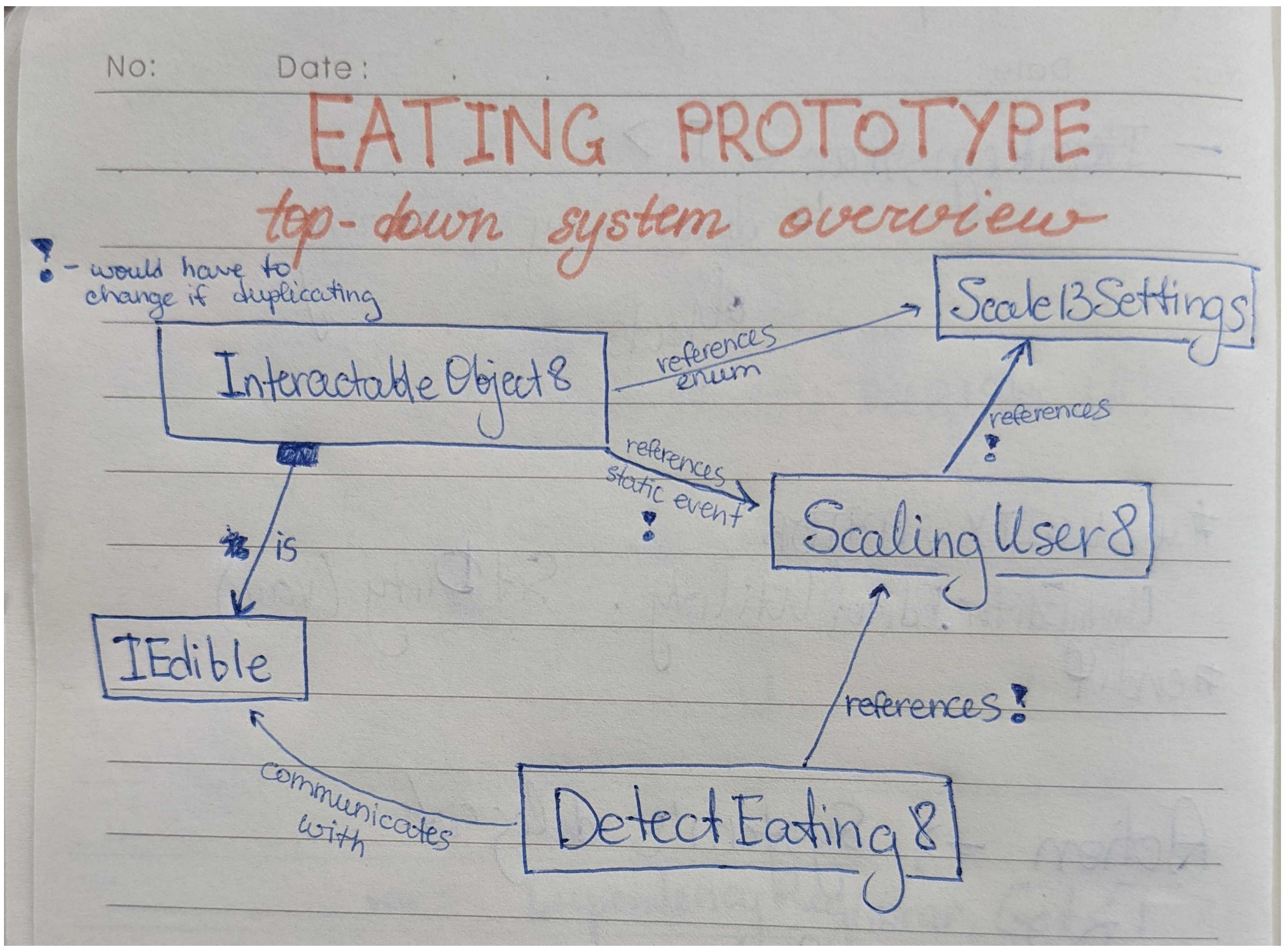
- [x] ~~Add particle effects when eating happens~~

▼ `wp-the-hungry-one--growing : 98` at `11-EatAndScaleParticles`

- [x] ~~Add sound effects when eating happens~~

▼ `wp-the-hungry-one--growing : 98` at `11-EatAndScaleParticles`

System Overview



Visual representation of dependencies between scripts. (does **not** follow UML or any other guides for creating diagrams like this)

`IEdible`

- Returns the scale the user gains from eating IEdible
- **Usage:** implement this interface for a desired class

`ScalingUser8`

- Scales user up and down
- **Usage:** attach to a random object in the scene, and fill out necessary object references

`InteractableObject8`

- Controls interactability of the object
- Implements `IEdible` interface, and returns a scale value based on the scale mode used (magnitude or percentage)
- **Usage:** attach to a `VRTK_InteractableObject` that we want to pick up and eat

`DetectEating8`

- Detects when an edible object entered the eating range of the player
- **Usage:** attach this script to an object with a collider set to trigger

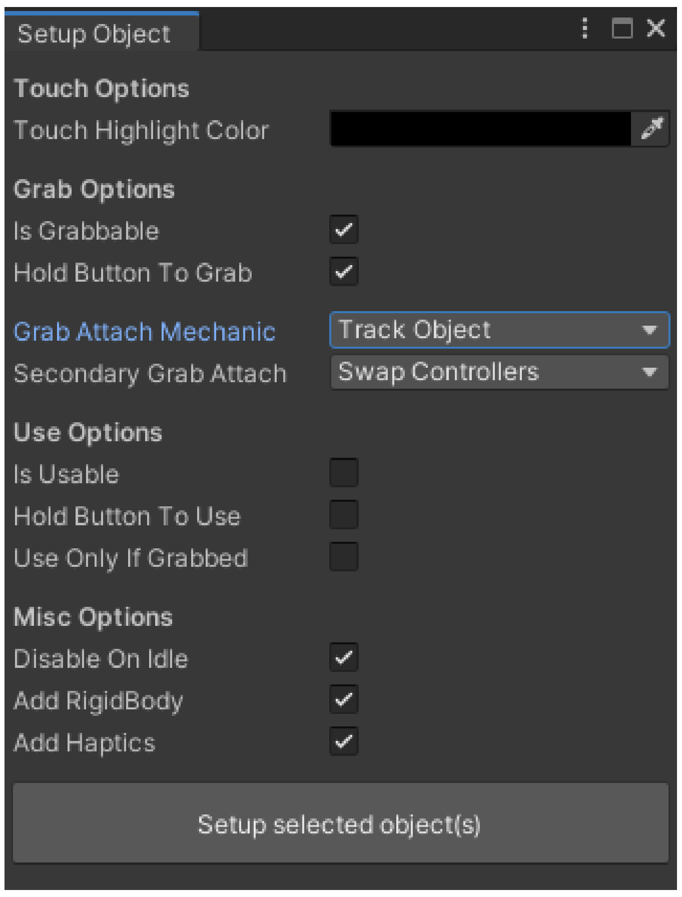
`Scale13Settings`

- The most up to date scale settings as of this prototype
- **Usage**: `ScalingUser8` takes in `Scale13Settings` instance as a parameter. Can be modified through the scene

Insights

- It's easier to find consistent scaling across each level when the user scales by a percentage rather than a set amount — We should probably try to use percentage based scaling in production.
- The ability to cast shadow on the environment gives the user a greater sense of presence in the moment
- In order to set up interactable object:

- Select the object(s) in the scene that you want to make interactable
- Go to Windows → VRTK → Setup Interactable Object
- Make sure the pop-up window has the following settings:



Setup Object

Touch Options

Touch Highlight Color	⬛

Grab Options

Is Grabbable	✓
Hold Button To Grab	✓
Grab Attach Mechanic	Track Object
Secondary Grab Attach	Swap Controllers

Use Options

Is Usable	
Hold Button To Use	
Use Only If Grabbed	

Misc Options

Disable On Idle	✓
Add RigidBody	✓
Add Haptics	✓

- Click "Setup selected object(s)". There will be no visual feedback, but you should see new VRTK-related scripts on your object(s) now.
- Go to `VRTK_TrackObjectGrab` component on your object(s) and make sure "Precision Grab" is turned on. (otherwise the objects might fly away from you)
- Include the custom InteractableObject script as a component on the object.
- You're all set! The object should be interactable and edible.