Potion Volume Simulation Documentation

---

1. Introduction

This Unity project is an educational simulation to teach the concept of volume for cubes and cuboids through an interactive potion-filling activity. The user selects a shape, customizes its dimensions, and then fills it with "potion" from a tap bucket, observing how volume translates into capacity. The experience includes synchronized voice-overs (VOs), visual effects, and real-time feedback.

Unity Version: 2022.3.21f1 (2D template)

Core Features:

- Shape selection (cube or cuboid) with dimension inputs
- Interactive sliding of shape into scene
- Voice-over narration guiding the user
- Tap bucket pre-fill and dynamic fill animation
- Smooth tweening animations using DOTween
- Perfect-fill feedback with shimmer particle effect

---

2. Project Structure

```
Assets/
├── Audio/                # VO clips: voReady, voPouring, voPerfect
├── Materials/
├── Prefabs/              # (Optional) prefab templates for shapes
├── Scenes/
│   ├── Welcome.unity     # Screen 1
│   ├── CustomizeShape.unity  # Screen 2
│   └── FillPotion.unity  # Screen 3
├── Scripts/
│   ├── ShapeCustomizer.cs
│   └── FillPotionScene.cs
├── Sprites/              # Bucket, ramp, cube, cuboid sprites
└── UI/                   # Canvas, buttons, input fields, images
```

---

3. Scene Breakdown

3.1. Scene 1: Welcome Screen

- **UI Elements:** Title, "Start" button, AudioSource
- **Flow:** On Start button click, `ShapeCustomizer.LoadNextScene()` triggers VO via `audioSource.Play()` then `SceneManager.LoadScene("CustomizeShape")` after VO length.

3.2. Scene 2: Customize Shape

- **UI Layout:**
  - Two buttons: Cube, Cuboid
  - Input fields:
    - Cube: one side `a`
    - Cuboid: `l`, `b`, `h`
  - Shape preview (Image component)
  - "Next" button enabled via `ValidateInputs()`
- **Script:** `ShapeCustomizer.cs`
  - Manages shape selection, dimension validation, PlayerPrefs storage
  - On Next: stores `Shape`, `L`, `B`, `H` and loads `FillPotion` scene

3.3. Scene 3: Fill Potion

- **UI & Visuals:**
  - Tap bucket with `bucketFillOverlay` (Image, Filled Vertical)
  - Off-screen `cubeObject` & `cuboidObject` (UI groups) placed at `spawnPoint`
  - Each shape group contains a `FillOverlay` Image (Filled Vertical)
  - `spawnPoint` and `targetPoint` empty Transforms to animate entry
  - Slider, StartPourButton, ResetButton, FeedbackText
  - Shimmer ParticleSystem for perfect fill effect
- **Audio:** `voReady`, `voPouring`, `voPerfect` via `AudioSource`
- **Script:** `FillPotionScene.cs`
  - On Start: reads `Shape`, dimensions from PlayerPrefs, calculates `volume = L*B*H`
  - Places active shape at `spawnPoint`, tweens to `targetPoint`
  - Pre-fills bucket (`bucketFillOverlay.fillAmount = volume/100f`)
  - On shape arrival: plays `voReady`, then `voPouring`, then enables Start button
  - On StartPour: begins fill in Update loop, animates `activeFillOverlay` and bucket fill via `Mathf.Lerp`
  - Updates value label via DOTween to show current liters (

---

4. Scripts Overview

4.1. ShapeCustomizer.cs

```
public class ShapeCustomizer : MonoBehaviour {
  public TMP_InputField inputA, inputL, inputB, inputH;
  public Image shapePreview;
  public Sprite cubeSprite, cuboidSprite;
  public Button nextButton;
  enum ShapeType { Cube, Cuboid }
  // SelectShape(), ValidateInputs(), OnNext() saving PlayerPrefs
}
```

4.2. FillPotionScene.cs

```
public class FillPotionScene : MonoBehaviour {
  public Slider fillSlider;
  public TMP_Text valueLabel, feedbackText;
  public Button startPourButton, resetButton;
  public Image bucketFillOverlay;
  public ParticleSystem shimmer;
  public GameObject cubeObject, cuboidObject;
  public Image cubeFillOverlay, cuboidFillOverlay;
  public Transform spawnPoint, targetPoint;
  public AudioSource audioSource;
  public AudioClip voReady, voPouring, voPerfect;

  private float volume;
  private bool isFilling, isFilled, canPressStart;
  private Image activeFillOverlay;

  void Start() { /* ... instantiate + tween + VOs ... */ }
  public void StartPouring() { /* enable fill */ }
  void Update() { /* fill logic */ }
  public void ResetScene() { /* reset all values + replay VOs */ }
}
```

---

5. DOTween Integration

- **Install:** Unity → Window → Package Manager → My Assets → DOTween → Import
- **Setup:** Unity → Tools → Demigiant → DOTween Utility Panel → Setup DOTween
- **Usage:** `using DG.Tweening;`
  - `transform.DOMove(targetPoint.position, 1.5f)`
  - `DOTween.To(() => ..., x => ..., currentFill, 0.3f)`

---

6. Voice-Over & Audio

- All VO clips placed in `Assets/Audio/`
- `AudioSource` on `AudioManager` GameObject
- Clips assigned in Inspector
- Timing controlled via `Invoke()` after VO lengths

---

7. Inspector Assignments

- **ShapeCustomizer (CustomizeShape Scene):**
  - Input Fields, Buttons, Preview Image, Sprites → Inspector
- **FillPotionScene (FillPotion Scene):**
  - `spawnPoint`, `targetPoint`, `cubeObject`, `cuboidObject`,
  - `cubeFillOverlay`, `cuboidFillOverlay`, `bucketFillOverlay`,
  - `fillSlider`, `valueLabel`, `feedbackText`, `startPourButton`, `resetButton`,
  - `shimmer`, `audioSource`, `voReady`, `voPouring`, `voPerfect`

---

8. Future Enhancements

- Animated potion stream from bucket to shape via particle trails
- Touch-friendly slider for mobile
- Additional practice scenes with computed missing dimensions
- Visual equation overlay (`3 x 3 x 3 = 27`)
- Save user performance and scores

End of Documentation.