

Shape Recognition Game Template Documentation

Introduction

This Unity Asset is a shape recognition game template designed to recognize and interact with various shapes. It uses a machine learning model (trained with ML ONNX) to accurately recognize shapes in real-time. The game offers a simple interface for retrieving recognized shapes during gameplay.

Features

- **Shape Recognition with ML ONNX:** Utilizes a machine learning model trained with ONNX to recognize a variety of predefined shapes, including:
 - Circle
 - Z
 - V
 - W
 - I
 - Heart
 - Infinity
 - E
 - Horizontal Line Up
 - Horizontal Line Down
 - Diagonal Line
 - Question Mark
 - A
 - X
 - Triangle
 - Square
 - N
- **Simple Integration:** Drag and drop the ShapeInputCanvas into your game scene, and access the recognized shape with a single line of code.
- **Customizable:** Easily check the recognized shape with ShapeInput.GetShape() during gameplay.

Setup

1. **Import the Template:** Import the Unity package into your project.
2. **Add ShapeInputCanvas:** Drag and drop the ShapeInputCanvas prefab into your scene.
3. **Access Recognized Shapes:** Use the following code to get the recognized shape during gameplay:

```
var shape = ShapeInput.GetShape();
```

This will return the shape recognized by the system.

Shape Recognition Details

The system is trained using a machine learning model in ONNX format to recognize the following shapes:

1. **Circle:** A round shape with no corners or edges.
2. **Z:** A letter Z shape with sharp angles.
3. **V:** A “V” shaped figure.
4. **W:** A “W” shaped figure.
5. **I:** A straight vertical line resembling the letter “I”.
6. **Heart:** A symmetrical shape resembling a heart.
7. **Infinity:** A figure eight symbol representing the concept of infinity.
8. **E:** The letter “E”.
9. **Horizontal Line Up:** A line that is oriented horizontally and points upwards.
10. **Horizontal Line Down:** A line that is oriented horizontally and points downwards.
11. **Diagonal Line:** A line drawn at a diagonal angle.
12. **Question Mark:** A “?” shape used to signify inquiry.
13. **A:** The letter “A”.
14. **X:** The letter “X”.
15. **Triangle:** A three-sided polygon.
16. **Square:** A four-sided shape with equal sides.
17. **N:** The letter “N”.

Gameplay Instructions

1. **Objective:** The objective of the game is to correctly identify and interact with the shapes presented.
2. **How to Play:** The game will present shapes to the player, and they need to match the shapes using the recognition system.

3. **Score System:** Points are awarded based on speed and accuracy.

Customization

- **Shape Recognition:** The system automatically detects the shapes based on the trained ONNX model. Simply call `ShapeInput.GetShape()` to access the recognized shape.
- **Gameplay Logic:** Adjust the gameplay flow, scoring, and difficulty based on your game's requirements.

Requirements

- Unity 2020 or later
- ONNX runtime support for Unity
- Compatible with both 2D and 3D projects
- Mobile or Desktop platform support

Known Issues

- Shape recognition may require fine-tuning for certain edge cases. Configuration options are available in the `ShapeInput` script.

Support

For support, bug reports, or suggestions, please contact us at gamesommy@gmail.com.