**Quaddle 2.0.1 Update: Prototype**

*Main Points*

* In addition to Quaddle 1.0, two types of objects have been added: a *Pedestal Object* and *Shape Object*
* **Pedestal Object**
  + *Feature Dimensions*
    - Character – An image displayed on the body of the object.

Kinds of Characters: 2 Chinese Characters

* + - Arm Ends – The shape of the arms that extend from it’s body

Kinds of Arm Ends: Flat, Blunt, Pointed, Flared

* + What is unique about a pedestal object?
    - It is always a box. The surface where the character is displayed is therefore always constant, so the characters will be easily readable and comparable. It permits the new ‘Character’ Dimension while controlling for (A) Mapping difficulty and (B) the Monkey’s abilities to recognize a character.
* **Shape Object**
  + *Feature Dimensions*
    - Shape – A unique shape
    - Kinds of Shapes: Cone, Pyramid, Hedra, Pentagon, Gengon, Carrot, Spring
  + What is unique about a shape object?
    - Funnily enough, there is not any incredibly purposeful and unique reason for its incorporation. However, in a stacking structure, it adds a simple yet noticeable feature since, unlike the Quaddle and Pedestal, it is *only* a shape. They are not complex like the others with characters or multiple arm shapes, and thus make a feature that is easily discernable.
* **The Totem Pole Structure**
  + Reminder: This simply is a way to stack the generated objects on top of each other.
  + In order to implement it, two components needed to change: (1) *the object table representation* and (2) *the code structure*.
* **The New Object Table**
  + Different Kinds of Objects
    - Because there are different kinds of objects, if you want to generate a specific kind of object, you simply denote it by the first letter of the object name followed by the feature dimension values of that object.

**Example**

*Quaddle*:“Q11111”, *Pedestal*:“P11”, *Shape*: “S1”

* + Representing Stacks of objects
    - We need to be able to denote stacks for the Totem Structure. All that requires is a string that tells the program to take pictures, export, and delete the just created stack and to begin a new one. I represented this as three dashed or “---“. Finally, I added a string type that denotes the end of object generation, or the end of the file. This string is three asterisks, or “\*\*\*”.

**Example Object Tables**

An Object Table that Generates 5 Quaddle 2.0.1s with a stack height of two each:

//Begin Object Table

S5

S3

---

S7

Q21121

---

S5

S6

---

Q12124

P21

---

S7

S3

\*\*\*

//End Object Table

An Object Table that Generates 2 Quaddle 2.0.1s with a stack height of four:

//Begin Object Table

P11

Q22113

Q12212

Q22223

P11

---

S7

S2

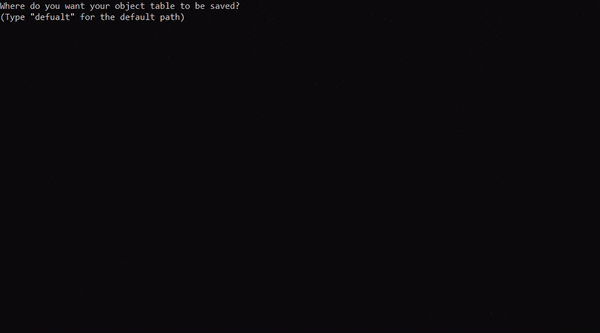
Q22222

S2

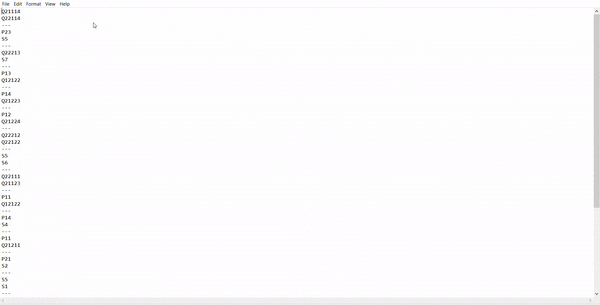
P12

\*\*\*//End Object Table

I have set up a small executable in c++ that generates random objects for testing.

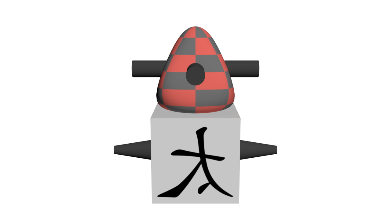
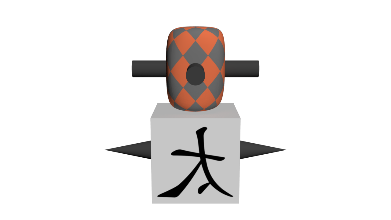
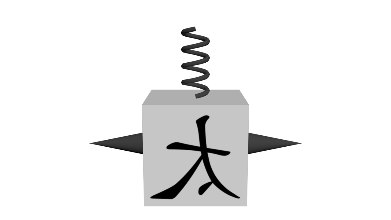
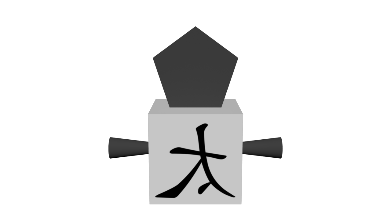
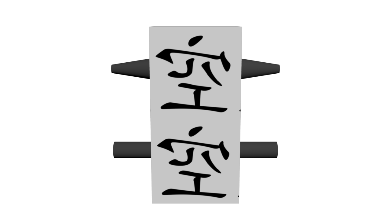
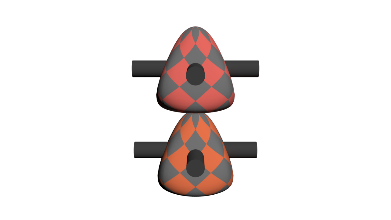
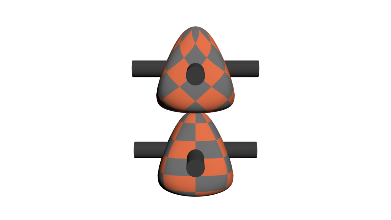
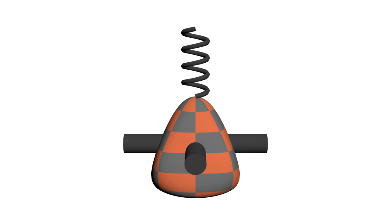
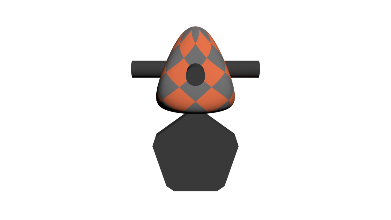
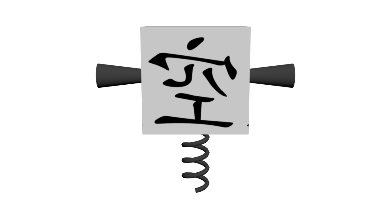
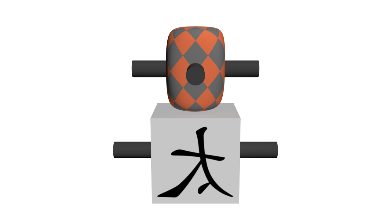
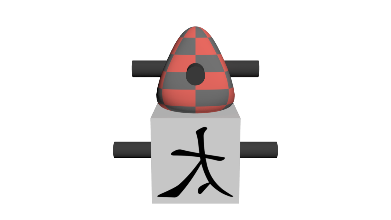
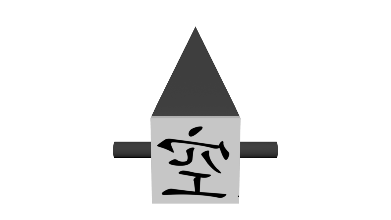
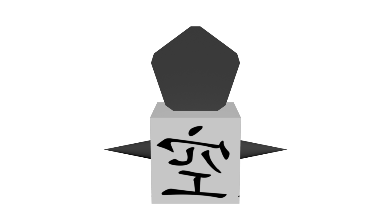
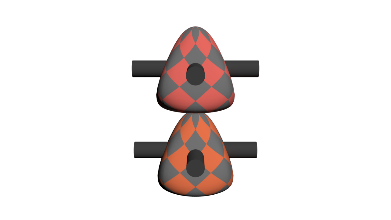
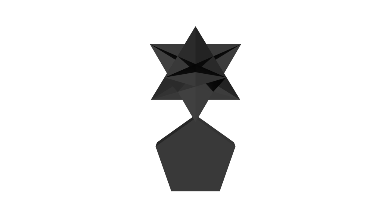
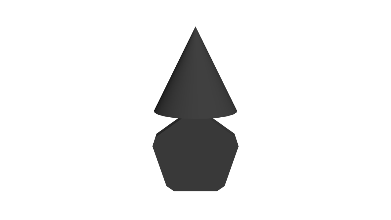
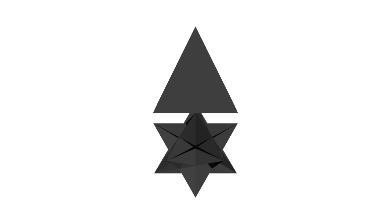
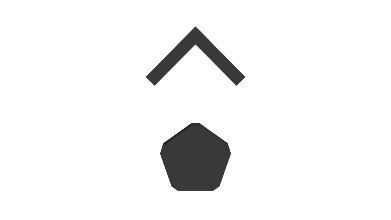


This Generated:



Example Output

After I generated the previous example for the object table generator, I ran it through the MaxScript these are the objects that were created.



I omitted the 45 degree angle pictures for the sake of space, but as you can see this is what the general out put looks like

* Code Structure highlights
  + \*\*\*Not all features from Quaddle 1.0 have been implemented (e.g. hats and transparency)
  + A better scoped script.
  + If you have more questions about the structure, please ask!

Conclusions/Opinion

**Most Important**: With this basic structure in place, what are some possible feature dimensions that may help this object look better *and* perhaps re-orient it? Should some be taken away? Should some be implemented differently?

I think the general structure is great; it is a good foundation to keep building these objects. However, they are not good-looking objects. A suggestion is to re-orient the totem pole structure to go horizontal with an arm that connects the two generated objects. I do not know if this is a good solution; it is just an example. Another thing that should be noted is the current build does not extend the Quaddle object, rather it creates two separate object types. If we wish to maintain the integrity of the ability to apply a slider to the different components developed in Quaddle 1.0, then I can rework the structure. This would raise questions about the character dimension implementation since right now, in the code, it develops an independent pedestal object (in other words, it is not defined as a part of a Quaddle). If we were to implement the box as an extension to the Quaddle, we would not want the character to be mapped to an object body that would make it hard to recognize. So, we would still need to signify “square Quaddle.” These are some of the new challenges with Quaddle 2.0.1. With this in mind let’s Discuss! Onto Quaddle 2.0.2.