

Programming Languages and Translators
COMS W4115
Stephen A. Edwards
Fall 2007

Orange White Green Animation Language
White Paper

Meenakshi Sripal
September 25, 2007

Introduction

Hinduism is the world's oldest religion that was originated in India. It is composed of numerous, diverse beliefs, traditions, and idols/deities. Today it is still survived by Hindus celebrating wide-variety of Indian festivals, thus preserving the India's culture.

Hinduism has many festivals throughout the year. The table below lists several popular Hindu festivals along with deities that are worshipped.

Festival	Deity
Durga Puja	Durga
Diwali	Lord Rama, Lord Ganesha, Goddess Lakshmi, Lord Mahabali
Holi	Lord Krishna
Ganesh Chaturthi	Lord Ganesha
Gowri Habba	Goddess Gowri
Rama Navami	Lord Rama
Krishna Janmashtami	Lord Krishna

Since religious practices are an integral part of everyday life, it is important to perform religious rituals in the right way to seek blessings or salvation from God. However, observation of rituals greatly vary among regions, villages, and individuals.

Motivation

Due to the complex nature of Indian religious beliefs, many young and old Indian people tend to deviate from performing religious rituals regularly and can easily forget the procedures. Therefore, I propose to develop a new programming language called Orange White Green Animation Language (OWGAL). The main objective of the OWGAL is to help users to create one or more routines (programs) that will produce flash animation files demonstrating how to conduct the ritual for different deities for different occasions/festivals. Then those animation files can be saved and shared among others to educate them how to express devotion to God/deities. Another goal of this high-level language is to help the beginners or novice devotees visualize certain objects that are needed for worship and how to perform rituals. OWGAL will be designed in a way that it is easy to understand with user-friendly syntax aiming users with little programming skills.

Properties of OWGAL

OWGAL language will follow the objected oriented design. The new animation library will be created consisting of pre-loaded images of deities, objects, and food. In addition, it is aimed to be dynamic allowing users to add additional deities, objects and food. Part of the animation library will be reused, such as open source codes that generate Flash animation files. Ming is planned to be used to compile the animation files turning them into .swf files. OWGAL language will have four program constructs: keyword and its property, setup method, and perform method.

KEYWORDS, PROPERTIES

The bold words below represent keywords that will be part of valid syntax. Properties are subset of values for each keyword.

Deity

Hindus normally pray to God through multiple idols or deities. Instances of deities include Shiva, Ganesha, Krishna, Rama, Lakshmi, Durga, and more. The valid Deity keyword will limit to pre-defined set of images that will be located in the animation library. In addition, the language will allow users to add their own deity by importing a new deity image (.gif, .jpeg) and adding to the animation library.

Syntax: `Deity(name=name)`

Properties: *name* - name of desired idol to be worshipped

Object

There are additional objects needed/required to perform a religious ritual. Objects include diya (candle lamp), flowers, garland, incense sticks, and kumkum (sacred powder), just to name a few.

Syntax: `Object(name=name, num=num, color=color, target=target, where=where)`

Properties:

name - name of the actual object to be used in ritual

num - how many items of the specified object

color - the color of the specified object

where - indicates which part of the idol to be placed with (such as feet, head, neck)

Offering

Worshippers often offer different things to their idols: flowers, clothes, holy thread, water, and food.

Syntax: `Offering(name=name, num=num, target=target, color=color, where=where, location=location)`

Properties:

num - number of times to offer

target - valid for certain keywords such as flower object, indicating to whom or where you're putting flowers

where - indicates which part of the idol to be placed with (such as feet, head, neck)

location - indicates where on the shrine to offer objects like food

Action

Devotees may perform the following chores during the worship: lighting a lamp, offering food to the deities, scripture reading, singing devotional hymns, meditation, chanting mantras, put garland around deities, apply kumkum (sacred powder) on deities' forehead, perform aarti (camphor offered to deity), and give deities holy bath. There will be pre-defined functions that will allow users to algorithmically perform rituals. Properties of the Action keyword are inputs into the pre-defined functions.

Syntax: `Action(name=name, pournum=pournum, target=target, withwhat=withwhat)`

Properties:

name - name of action to be done
withwhat - only valid if keyword *name* equals bath, value indicating to be used while giving the deity bath
pournum - how many times to pour
roundnum - how many times to go around in front of the deity holding the diya

Food

Food is the popular offering given to God and eaten by devotees after the worship is over. For some rituals for certain festivals, especially Ganesh Chaturthi, devotees give deities a bath, using even food! Food examples are: apple, banana, ghee (liquid butter), milk, tumeric, oranges just to name a few.

Syntax: `Food(name=name, num=num)`

Properties:

name - name of food
num - number of food items

SETUP METHOD

Setup method is the predefined subroutine that users can specify in the beginning of the user-defined routine to indicate what items are needed to prepare and which deity will be the focus of the worship. It will be responsible for loading the specified objects onto the standard shrine and establishing OpenGL.

PERFORM METHOD

Perform is another predefined subroutine that users can specify right after Setup method. This is where the real action is defined how to do the worship.

Program Structure

```
EventName.Routine {  
    //Setup method  
    Setup {  
        ...  
    }  
    //Perform method  
    Perform {  
        ...  
    }  
    Exit  
}
```

Sample Code

```
Holi.Routine1 {  
    //Setup method  
  
    Setup{  
        Deity(name=Krishna)  
        Object(name=diya, num=2, color=gold).Light  
        Object(name=incensestick, num=2).Light  
        Offering(name=flower, num=1, color=red,  
            target=Deity(name=Krishna), where=feet)  
        Offering(name=(Food(name=ghee)), num=1,
```

```
        location=frontcenter)
    }

//Perform method

Perform{
    Action(name=bath, withwhat=water, pournum=3
           target=Deity(name=Krishna), where=head)
    Action(name=aarti, roundnum=3 target=Deity(name=Krishna))
}
Exit
}
```