**Karl Palmer**

Assabet Valley Regional Technical High School

Computer Programming and Web Development

**1 Introduction**

The purpose of this document will be used to give detailed instruction on how a software will be developed to take in x numbers, display and add that amount of dice, and display the total. The first section in this group is the Introduction, which will go over the basic job that the system will perform as well as explain the purpose of the document. The second section that the document has is the design specification, which is what the system needs to do and what it needs to function. The design specification is split up into 4 different categories which specifies exactly what is needed. Data Design is what constants, variables, and other data is required for the system to work, the functional design is how the system will perform the specific actions that it needs to work, the non-functional design is how the system should perform based on the functional requirements, and the final category being user interface design which covers how the user interface of the system will be implemented.

In this program its purpose is to roll x amount of six sided die and add/show all the dice values together, then output the total values to the user. To do this the main program will have buttons for subtracting/adding die from the die amount then another button that causes a roll method to occur. The roll method will calculate the necessary size of the dice for the amount the user wants to fit on the screen and with that size it will generate a random number for each dice and with this number it will display the dice on the screen and add it to the total. The total will be displayed next to the roll button.

**2 Design Specification**

**2.1 Data Design**

How the main data design; constants, variables, etc., of the system ***Dice Roller 9001*** shall be implemented***.***

1. int WIDTH = 1000
2. int HEIGHT = 500
3. Color WHITE = rgb(255, 255, 255)
4. Color BLACK = rgb(0, 0, 0)
5. String[] DIEFACES = all image locations for six sided die
6. Font diceAmtText = Product Sans, size 30
7. Mouse mouse = mouse data
8. \_image\_library = saved loaded images
9. Int diceAmt = 1
10. total = 0

**2.2 Functional Design**

How the main functional requirements of the system ***Dice Roll 9001*** shall be implemented:

image function get\_image(path)

image = \_image\_library.get(path)

if image == null then

image = load image with pygame library

end if

return image

end get\_image

procedure button(type, img, xmin, xmax, ymin, ymax)

display button at (xmin, ymin)

if mouse pressed && mouseX > xmin && mouseX < xmax && mouseY > ymin && mouseY < ymax then

if type == ‘sub’ && diceAmt != 1 then

diceAmt -= 1

else if type==’add’ then

diceAmt += 1

else if type==’roll’ then  
 roll()

end if

end if

end button

procedure drawbg

window.fill(WHITE)

draw BLACK rectangle on the bottom of the screen with a height of 70 and width of WIDTH

button(‘sub’, ‘images/Sub.png’, 10, 60, HEIGHT – 60, HEIGHT - 10)

button(‘add’, ‘images/Add.png’, 70, 120, HEIGHT – 60, HEIGHT - 10)

if diceAmt > 1 then  
 display diceAmt + ‘dice’ to the user

else then

display diceAmt + ‘die’ to the user

end if

button(‘roll’, ‘images/Roll.png’, WIDTH – 100, WIDTH – 16, HEIGHT – 60, HEIGHT - 10)

display total to the user

end drawbg

procedure roll()

total = 0

rows = squareroot(diceAmt)

cols = math.ceil(diceAmt / rows)

posx = 0

posy = 0

if diceAmt == 1 then

dicesize = 400

else then

dicesize = WIDTH / col

end if

for die in diceAmt + 1

diefacenum = random.randint(0, 5)

total += doefacenum + 1

dieface = get\_Image(DIEFACES[diefacenum], diesize, diesize)

display dieface

posx += diesize

if posx > WIDTH-diesize then

posy += diesize

posx = 0

end if

update display

delay(1/diceAmt)

end for

end roll

main program

while True

event = pygame.event.wait()

if event.type == QUIT then

pygame.quit()

sys.exit()

else then

mouseX = get mouse posX

mouseY = get mouse posY

drawbg()

update display

end if

end while

end main program

**2.3 Non-functional Design**

How the most important non-functional require­ments for the system ***Dice Roll 9001*** shall be implemented:

Hardware:

1 GHz or faster 32-bit (x86) or 64-bit (x64) processor

1 GB RAM (32-bit) or 2 GB RAM (64-bit)

16 GB available hard disk space (32-bit) or 20 GB (64-bit)

DirectX 9 graphics device with WDDM 1.0 or higher driver

Operating System:

Windows: Windows 7 or later

Mac: Max OS X 10.6 and later w/ Intel processor

Linux: Ubuntu, Red Hat w/ 250 MB of free space, openGL drivers installed, and Java Runtime Environment installed

**2.4 User Interface Design**

How the user interface require­ments for the system ***Dice Roll 9001*** shall be implemented:

|  |
| --- |
| A GUI with add/sub dice button and roll button with displays of dice amount and dice roll total |
|  |

**References**

A list of all references used for your system requirements should be here.

PyGame, sys, os, random, time, math libraries