Game: Memory

Version: 1

**Description**

**After starting the program, a window with title “Memory” and black background will be displayed on the screen. And in the window a grid with 16tiles will be shown on the left side of the window. Each tile of the grid has image, bounded by black 4 pixels black border. 8 images will be randomly distributed to the 16 tiles, each appears twice.** At the start of the beginning, each tile will be covered in a default image. The default image has blue background and a red ‘?’. **The right side of the window will be filled by black rectangle.** And on the top right of the window, the time used will be displayed in white font, in seconds.

When the player clicks on a specific covered tile, the image distributed to it will be exposed. When the player clicks on another covered tile, the tile image of the second tile is also revealed. If the exposed images are different, they will be covered by default image again after a 1 second delay. If the exposed images are same, they will be exposed permanently. Any click outside a covered tile will be ignored. The player can keep searching for identical images until all of the tiles are exposed permanently. When this happens, the game is over and the timer is stopped. **When the close box is clicked, the window closes.**

**Functional Tests**

1. **Start the program**
   1. **Does a window appear?**
      1. **Is the window with black background?**
      2. **Does the window have a title 'Memory'?**
   2. **Is a grid with 16 tiles displayed on the left side of the window?**
      1. **Does each tile contain an image?**
      2. **Does each tile have a black border?**
   3. **Is there a rectangle on the right of the screen?**
      1. **Is it black?**
      2. **Does it fill the rest part of the window?**
2. **Click the close box.**
   1. **Does the window close?**
3. **Restart the game**
   1. **Is the distribution of the images different from that of the previous game process?**

**Main Algorithm**

create window

create game

play game

close window

**Class Game**

***Instance Attributes***

window # the window on which to draw

pause\_time # pause time between drawing frames

close\_clicked # indicates if close button was clicked

continue\_game # indicates if game should continue

**grid\_size # (int, int) indicates the number of rows and columns respectively**

**images # [images] contains the images**

**grid # [[Tile]]**

***Instance Methods/Blocks***

**initialize instance**

# initialize/create all instance attributes

**get\_images**

**create\_grid**

**get\_images**

**for image in dirname:**

**load image**

**make a list containing 2 copies of every images**

**randomize the list**

**create\_grid**

**for i in grid\_size[0]:**

**create\_row**

**add the row to the grid**

**create\_row**

**for j in grid\_size[1]:**

**create a tile**

**add the tile to row**

**play game**

while not close\_clicked

# ‘play’ a single frame

handle next event

draw the current frame

if continue\_game:

update all game objects

decide if game should continue

pause before next iteration/frame

**handle event**

get next event from the event queue

if event type == QUIT

close\_clicked = True

# check more events as required

pass

**update game objects**

# update Game objects to new position in next frame

pass

**draw frame**

erase the window

# draw the Game objects

for row in grid:

for tile in row:

tile.draw

update the window

**decide if game should continue**

# check if game should continue or not

pass

Class Tile

Class Attributes

window # indicates the window on which the tiles will be displayed

br\_color # str indicate the border color

br\_width # int indicate the width of the border

Instance Attributes

image # indicates the image that the tile has

position # (int, int) indicate the position where the tile is drawn

Class Methods

set\_window

Instance Method

initialize instance

# initialize the tile based on the image and position

draw

draw the image

draw the black border