**Memory V3 Reflection Activity**

**Q1 Assume that you want to extend the information of your memory game by keeping track of how many times in total the player exposed and viewed each hidden tile before finding a match. How do you have to change the code to keep track of how many times in total the player clicked on each hidden tile?**

|  |
| --- |
| def select(self, position):  # handle the click action  # position passed is event.pos, indicaing the osition of the click  if self.rect.collidepoint(position):  if not self.is\_exposed:  self.count += 1 # this is the line added to count how many times the tile is exposed, in which the self.count is initialize as 0  self.is\_exposed = True  return True  return False |

**Q2 Based on the extension in the previous question, change the draw method of tiles so that when a tile is exposed, the number of times it has been viewed is shown in the top left corner of the tile. (Note you can draw text over top of the tile image by using draw\_string after blit).**

|  |
| --- |
| def draw(self):  # draw the tile  surface = self.window.get\_surface()  if self.is\_exposed:  surface.blit(self.image, self.position)  self.window.draw\_string(str(self.count), self.x, self.y) #this is the line added  else:  surface.blit(Tile.default\_image, self.position)    pygame.draw.rect(surface, self.br\_color, self.rect, self.br\_width) |

**Q3. Based on the extension in the previous question, extend the code again so that the number of times a tile has been viewed is shown in the following way: 1) The number should only be shown for the last selected tile, i.e., when the first tile is exposed, the number is shown; then, when the second tile is selected, the first tile should not show the number anymore, but the second tile should show the number - until a match is found, in which case the numbers should be shown permanently in both matching tiles.**

|  |
| --- |
| Add a new instance attribute named show\_counts to Tile class to control whether it should display the counts in draw method  def handle\_mouseup(self, event):  # handle a click event performed by the player  for row in self.grid:  for tile in row:  if tile.select(event.pos): // set the show\_counts to True in the select method  if self.clicked is None:  self.clicked = tile  else:  if self.clicked != tile:  self.recover.append(self.clicked)  self.recover.append(tile)  # set the show\_counts of the self.clicked to false here  # also set the show\_counts to false in the recover method  self.clicked = None |