

## Reporte Sprint #0

### Instrucciones

#### Objetivos

- Tomar decisiones sobre el proyecto de desarrollo de software SOS.
- Aprender pruebas unitarias y programación de GUI en el lenguaje de tu elección.

#### Entregables y políticas de calificación

Lean el documento “descripción del Proyecto 3S2” cuidadosamente y toma las decisiones para el desarrollo del software.

Usen el siguiente template para completar tu reporte.

#### 1. Decisiones claves para el proyecto SOS of the SOS (2 puntos)

|   |   |
|---|---|
| Lenguaje de programación orientado a objetos                | Python  |
| Librería GUI (recomendable)                                 | tkinter   |
| IDE (Integrated Development Environment)                    | pycharm   |
| Framework xUnit (JUnit for Java por ejemplo)                | unittest  |
| Guía de estilo de programación (debe ser leído con cuidado) | POO   |
| Sitio de alojamiento del proyecto                           | <a href="https://github.com/kritzanyeraldin/CC--3S2/tree/main/Proyecto-3S2">https://github.com/kritzanyeraldin/CC--3S2/tree/main/Proyecto-3S2</a> |
| Otras decisiones si procede                                 |   |

Ejemplos de guía de estilo de programación:

- Guía de estilo de Java Google: <https://google.github.io/styleguide/javaguide.html>
- Guía de estilo de C++ Google: <https://google.github.io/styleguide/cppguide.html>
- Guía de estilo Python Google: <https://google.github.io/styleguide/pyguide.html>

#### 2. Pruebas unitarias (8 puntos)

Encuentren un tutorial sobre el framework de pruebas unitarias que has elegido y escriban al menos dos pruebas xUnit de un programa que hayas escrito o encontrado en otro lugar. Adjunta aquí (1) la captura de pantalla de la ejecución de tu programa.

Programamos el juego de michi y realizamos dos pruebas que chequea los valores ‘O’ y ‘X’

```

main.py x test_main.py x C:\...\test_main.py x
1
2 class michi:
3     def __init__(self):
4         pass
5     def check(self, tablero):
6         j=0
7         pos=0
8         vacio=" "
9         for i in range(0,3):
10            while j!=3:
11                if(tablero[i][j]==tablero[i][2]):
12                    pos=i
13                    bo=True
14                else:
15                    bo=False
16                    pos=0
17                break
18            j=j+1
19        if(bo==True):
20            return tablero[pos][pos]
21        else:
22            return vacio

```

```

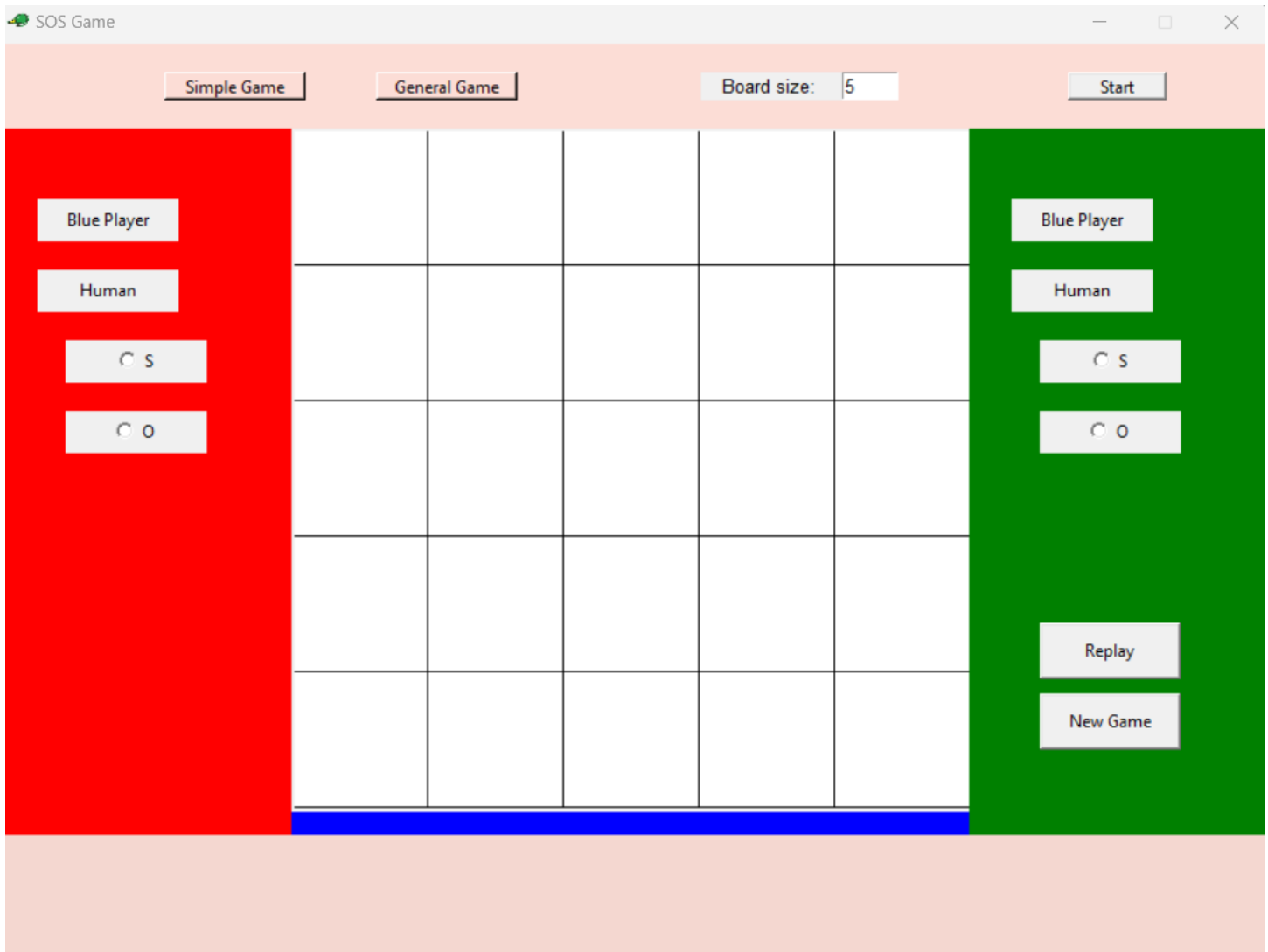
main.py x test_main.py x C:\...\test_main.py x
1 from unittest import TestCase
2 from main import michi
3
4 class Testmichi(TestCase):
5     def test_check_0(self):
6         tablero = [["X", "0", "0"], ["0", "0", "0"], ["0", "X", "X"]]
7         m = michi()
8         esperado="0"
9         resultado = m.check(tablero)
10        self.assertEqual(esperado, resultado)
11    def test_check_X(self):
12        tablero = [["X", "0", "0"], ["0", "X", "0"], ["X", "X", "X"]]
13        m = michi()
14        esperado="X"
15        resultado = m.check(tablero)
16        self.assertEqual(esperado, resultado)

```

### 3. Programación GUI (10 puntos)

Escriban un programa GUI en el lenguaje que hayas elegido para tu proyecto SOS. La GUI de tu programa debe incluir texto, líneas, una casilla de verificación y botones de opción. Si bien se recomienda considerar la GUI para el tablero de juego SOS, no es obligatorio. En esta tarea, cualquier programa GUI de tu propio trabajo es aceptable.

Adjunten aquí (1) la captura de pantalla de la ejecución de tu programa y (2) el código fuente de tu programa.



```
main.py x Manager.py screens.py
1 from Manager import Manager
2
3 if __name__ == '__main__':
4     app = Manager()
5     app.mainloop()
```

```
SOS main
main.py Manager.py x screens.py
1 import tkinter as tk
2 from tkinter import PhotoImage
3 from screens import Container
4 class Manager(tk.Tk):
5     def __init__(self, *args, **kwargs):
6         super().__init__(*args, **kwargs)
7         self.title("SOS Game")
8         self.geometry("900x650")
9         icon = PhotoImage(file="imagenes/duck.png")
10        self.iconphoto(True, icon)
11        self.resizable(False, False)
12        container = tk.Frame(self)
13        container.pack(
14            side=tk.TOP,
15            fill=tk.BOTH,
16            expand=True)
17        container.configure(bg="red")
18
19        self.frames = {}
20        f = Container
21        frame = f(container, self)
22        self.frames[f] = frame
23        # frame.grid(row=0,column=0,sticky=tk.NSEW)
24        self.show_frame(Container)
25
26    def show_frame(self, container):
27        frame = self.frames[container]
28        frame.tkraise()
29
30 1 usage Kritzan Yeraldin
31
32 SOS > sprint0 > Manager.py
```

PC SOS main

main.py Manager.py screens.py

import tkinter as tk

2 from tkinter.messagebox import \*

3 from constantes import style

4 from Board import Board

5

3 usages Kritzan Yeraldin

6 class Container(tk.Frame):

7 Kritzan Yeraldin

7 def \_\_init\_\_(self, parent, controller):

8 super().\_\_init\_\_(parent)

9 #dimension de la vista login

10 self.pack()

11 self.place(x=0, y=0, width=900, height=660)

12 self.configure(bg=style.beige)

13 #edita desde aqui

14 self.controller = controller

15 self.init\_widgets()

16

17 self.frames={}

18 self.create\_frames\_button = tk.Button(self, text="Start", command=self.create\_frames)

19 self.create\_frames\_button.place(x=760, y=20, width=70, height=20)

20

1 usage Kritzan Yeraldin

21 def create\_frames(self):

22 f = Container2

23 frame = f(self, self)

24 self.frames[f] = frame

25 frame.tkraise()

26

```
SOS ▾  main ▾
main.py  Manager.py  screens.py ×
25     frame.tkraise()
26
27     1 usage  Kritzan Yeraldin
28     def show_frame(self, container):
29         frame=self.frames[container]
30         frame.tkraise()
31
32     Kritzan Yeraldin
33     def control(self):
34         self.show_frame(Container2)
35
36     1 usage  Kritzan Yeraldin
37     def init_widgets(self):
38         #frame1
39         self.frame1=tk.Frame(self,bg=style.beige)
40         self.frame1.place(x=0,y=0,width=760, height=60)
41
42         self.simple_game_button=tk.Button(self.frame1,text="Simple Game",bg='#FCDDDD6')
43         self.simple_game_button.place(x=120,y=20,width=100,height=20)
44
45         self.general_game_button=tk.Button(self.frame1,text="General Game",bg='#FCDDDD6')
46         self.general_game_button.place(x=270,y=20,width=100,height=20)
47
48         label_board_size=tk.Label(self.frame1,text='Board size: ',font='Arial 10')
49         label_board_size.place(x=500,y=20,width=100,height=20)
50
51         self.entry_board_size=tk.Entry(self.frame1,font='Arial 10')
52         self.entry_board_size.place(x=600,y=20,width=40,height=20)
```

PC SOS main

main.py Manager.py screens.py

```
49         self.entry_board_size.place(x=600,y=20,width=40,height=20)
50
51     2 usages Jaime Acuña +1 *
52     class Container2(tk.Frame):
53         Kritzan Yeraldin +1
54         def __init__(self,parent,container):
55             super().__init__(parent)
56             self.container=container
57             self.pack()
58             self.place(x=0,y=60,width=900,height=600)
59             self.configure(bg=style.beige2)
60             self.condition()
61
62     1 usage Kritzan Yeraldin +1 *
63     def condition(self):
64         if int(self.container.entry_board_size.get()) >= 3:
65             label=tk.Label(self,text='agui2')
66             label.place(x=0,y=0,width=20,height=20)
67             #self.create_board()
68             self.init_widgets()
69         else:
70             showerror(message="Tamaño invalido.")
71
72     1 usage Jaime Acuña +1 *
73     def create_board(self):
74
75         #Se actualiza el tamaño del contenedor del tablero
76         self.frame_board.update()
77
78         #Crea un tablero vacio
79         """
```

```
SOS main
main.py Manager.py screens.py x
69 def create_board(self):
70
71     #Se actualiza el tamaño del contenedor del tablero
72     self.frame_board.update()
73
74     #Crea un tablero vacío
75     """
76     self.board_size toma el valor del tamaño del tablero que se ingresa en la interfaz después de apretar start
77     lo que se quiere es que el tamaño del tablero se adapte a su contenedor en este caso sería self.frame_board
78     así que se toma el mínimo valor entre ancho y altura y luego se divide entre el tamaño del tablero ingresado
79     de esta forma se logra contener el tablero en el frame board sea cual sea el tamaño que se ingrese.
80     """
81     self.board = Board(int(self.container.entry_board_size.get()))
82     self.board_size=self.board.get_board_size()
83
84     frame_width = self.frame_board.winfo_width()
85     frame_height=self.frame_board.winfo_height()
86     #print(f'frame {frame_width} \n cell')
87     self.cell_size = min(frame_width,frame_height)/self.board_size
88     #print(self.cell_size)
89
90
91     self.canvas_width = self.board_size * self.cell_size
92     self.canvas_height = self.board_size * self.cell_size
93
94     self.canvas = tk.Canvas(self.frame_board, width=self.canvas_width, height=self.canvas_height)
95     self.canvas.place(x=0,y=0)
96
97     #dibuja los rectangulos del tablero
98     for row in range(self.board_size):
99         for col in range(self.board_size):
100
101             Container2 > init_widgets()
```

```
SOS main
main.py Manager.py screens.py x
96
97     #dibuja los rectangulos del tablero
98     for row in range(self.board_size):
99         for col in range(self.board_size):
100             x0 = col * self.cell_size
101             y0 = row * self.cell_size
102             x1 = (col + 1) * self.cell_size
103             y1 = (row + 1) * self.cell_size
104             self.canvas.create_rectangle(x0, y0, x1, y1, fill='white', tags='cell')
105
106             #self.canvas.tag_bind('cell', '<Button-1>', self.on_cell_clicked)
107
108     Jaime Acuña
109 def on_cell_clicked(self, event):
110     x = self.canvas.canvasx(event.x)
111     y = self.canvas.canvasy(event.y)
112
113     col = int(x // self.cell_size)
114     row = int(y // self.cell_size)
115
116     if self.board[row][col] == '':
117         self.board[row][col] = 'S'
118         self.canvas.create_text(
119             (col + 0.5) * self.cell_size,
120             (row + 0.5) * self.cell_size,
121             text='S',
122             font=('Arial', 32),
123             fill='red'
124         )
125
126 Container2 > init_widgets()
SOS > print0 > screens.py
```



```
SOS main
main.py Manager.py screens.py x
125
126 1 usage Jaime Acuña *
127
128 def init_widgets(self):
129
130     #frame_blue_player
131     self.frame_blue_player=tk.Frame(self,bg='red')
132     self.frame_blue_player.place(x=0, y=0,width=210,height=500)
133
134     #frame_tablero
135     self.frame_board=tk.Frame(self,bg='blue')
136     self.frame_board.place(x=210, y=0,width=480,height=500)
137     #print(f'frame {self.frame_board.winfo_width()}')
138
139     #frame_red_player
140     self.frame_red_player=tk.Frame(self,bg='green')
141     self.frame_red_player.place(x=690, y=0,width=210,height=500)
142
143     #frame_blue_player(210x500)
144     label_blue_player=tk.Label(self.frame_blue_player,text='Blue Player')
145     label_blue_player.place(x=30,y=50,width=100,height=30)
146
147     label_human=tk.Label(self.frame_blue_player,text='Human')
148     label_human.place(x=30,y=100,width=100,height=30)
149
150     radioValue = tk.IntVar()
151     self.S_red_player = tk.Radiobutton(self.frame_blue_player, text='S',variable=radioValue,value=1)
152     self.S_red_player.place(x=50,y=150,width=100,height=30)
153     self.O_red_player = tk.Radiobutton(self.frame_blue_player,text='O',variable=radioValue,value=0)
154     self.O_red_player.place(x=50,y=200,width=100,height=30)
155
Container2 > init_widgets()
SOS > sprint0 > screens.py
```

```
SOS main
main.py Manager.py screens.py x
148
149
150     radioValue = tk.IntVar()
151     self.S_red_player = tk.Radiobutton(self.frame_blue_player, text='S',variable=radioValue,value=1)
152     self.S_red_player.place(x=50,y=150,width=100,height=30)
153     self.O_red_player = tk.Radiobutton(self.frame_blue_player,text='O',variable=radioValue,value=0)
154     self.O_red_player.place(x=50,y=200,width=100,height=30)
155
156     #frame_red_player
157     label_red_player=tk.Label(self.frame_red_player,text='Blue Player')
158     label_red_player.place(x=30,y=50,width=100,height=30)
159
160     label_human=tk.Label(self.frame_red_player,text='Human')
161     label_human.place(x=30,y=100,width=100,height=30)
162
163     radioValue = tk.IntVar()
164     self.S_blue_player = tk.Radiobutton(self.frame_red_player, text='S',variable=radioValue,value=1)
165     self.S_blue_player.place(x=50,y=150,width=100,height=30)
166     self.O_blue_player = tk.Radiobutton(self.frame_red_player,text='O',variable=radioValue,value=0)
167     self.O_blue_player.place(x=50,y=200,width=100,height=30)
168
169     self.replay_button = tk.Button(self.frame_red_player, text='Replay')
170     self.replay_button.place(x=50,y=350,width=100,height=40)
171     self.new_game_button = tk.Button(self.frame_red_player,text='New Game')
172     self.new_game_button.place(x=50,y=400,width=100,height=40)
173
174     # creamos el tablero
175     self.after(1000,self.create_board())
176
177
178
Container2 > init_widgets()
```