

Informe de Laboratorio Proyecto Final

Tema: Proyecto Final

Nota

Estudiante	Escuela	Asignatura
Christian Mestas Zegarra cmestasz@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Fundamentos de la Programación 2 Semestre: II Código: 1701213

Laboratorio	Tema	Duración
Proyecto Final	Proyecto Final	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2023 - B	Del 22 Enero 2024	Al 02 Febrero 2024

1. Tarea

- **1:** Presentar opciones para partida rápida, personalizada, continuas partida (uso de archivos guardados).
- **2:** Mostrar juego desarrollándose en un tablero, se muestran los datos de cada personaje.
- **3:** Se muestra que se instancié un personaje determinado, como caballero, lancero, arquero o espdachín.
- **4:** Mostrar la opción de poder volver a iniciar una partida o guardar la que se estaba jugando.
- **5:** Mostrar el archivo utilizado para guardar la partida.
- **6:** Mostrar el empleo de la base de datos.

2. Equipos, materiales y temas utilizados

- Sistema Operativo Microsoft Windows 10 Pro 64 bits
- Visual Studio Code 1.82.2
- Java Development Kit 17.0.1
- JavaFX sdk 21.0.1
- Git 2.41.0.windows.1
- Windows PowerShell 5.1.19041.3031
- Cuenta en GitHub con el correo institucional.
- Programación Orientada a Objetos
- HashMap de Objetos
- Agregación y composición
- Herencia y polimorfismo
- Interfaces
- Miembros de clase e instancia
- Interfaz gráfica de usuario
- Bases de datos
- Archivos

3. URL de Repositorio Github

- URL del Repositorio GitHub para clonar o recuperar.
- <https://github.com/cmestasz/fp2-23b.git>
- URL del proyecto final en el Repositorio GitHub.
- https://github.com/cmestasz/fp2-23b/tree/main/fase03/proyecto_final

4. Actividades con el repositorio GitHub

commits.bash

```
1 $ git add .gitignore
2
3 $ git commit -m ".gitignore actualizado para solo publicar la carpeta src"
4 [main e0cc937] .gitignore actualizado para solo publicar la carpeta src
5 1 file changed, 3 insertions(+)
6
7 $ git add .
8
9 $ git commit -m "Ejemplo dado en la documentacion de JavaFX"
10 [main 118a77f] Ejemplo dado en la documentacion de JavaFX
11 9 files changed, 155 insertions(+)
12 create mode 100644 .vscode/settings.json
13 create mode 100644 fase03/proyecto_final/VIDEOGAME/.vscode/launch.json
14 create mode 100644 fase03/proyecto_final/VIDEOGAME/.vscode/settings.json
15 create mode 100644 fase03/proyecto_final/VIDEOGAME/README.md
16 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/FX/Controller.java
17 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/FX/Main.fxml
18 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/FX/Videogame.java
19 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/FX/style.css
20 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/FX/test.fxml
21 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/4.png
22
23 $ git push -f
24 Enumerating objects: 24, done.
25 Counting objects: 100% (24/24), done.
26 Delta compression using up to 4 threads
27 Compressing objects: 100% (19/19), done.
28 Writing objects: 100% (21/21), 3.69 MiB | 1.04 MiB/s, done.
29 Total 21 (delta 1), reused 0 (delta 0), pack-reused 0
30 remote: Resolving deltas: 100% (1/1), done.
31 To https://github.com/cmestasz/fp2-23b.git
32 + af4068d...118a77f main -> main
33
34 $ git add .
35
36 $ git commit -m "Menú principal"
37 [main e9a3e21] Menú principal
38 9 files changed, 92 insertions(+), 34 deletions(-)
39 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/FX/Main Menu.fxml
40 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/barrack.png
41 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/barrack.psd
42 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/waiting.png
43 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/waiting.psd
44
45 $ git add .
46
47 $ git commit -m "Controlador del menú principal"
48 [main 84ae869] Controlador del menú principal
49 5 files changed, 131 insertions(+), 14 deletions(-)
50 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/FX/MainMenu/Main Menu.fxml
51 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/FX/MainMenu/MainMenuController.java
52 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/Utils/Resolution.java
```

```
53 $ git add .
54
55
56 $ git commit -m "Modelo de servidor para manejar varias instancias locales del videojuego"
57 [main 5e5d1a0] Modelo de servidor para manejar varias instancias locales del videojuego
58 4 files changed, 133 insertions(+), 56 deletions(-)
59 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/FX/Main Menu.fxml
60 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/FX/MainMenu/MainMenuServer.java
61
62 $ git add .
63
64 $ git commit -m "Servidor completo"
65 [main 1d626a9] Servidor completo
66 1 file changed, 33 insertions(+), 11 deletions(-)
67
68 $ git add .
69
70 $ git commit -m "Funcionalidad de servidor para el menú principal completa"
71 [main abc9d8d] Funcionalidad de servidor para el menú principal completa
72 5 files changed, 198 insertions(+), 58 deletions(-)
73 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/FX/MainMenu/Connection.java
74 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/FX/MainMenu/Operation.java
75 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/Utils/Utils.java
76
77 $ git push
78 Enumerating objects: 76, done.
79 Counting objects: 100% (76/76), done.
80 Delta compression using up to 4 threads
81 Compressing objects: 100% (59/59), done.
82 Writing objects: 100% (65/65), 30.02 KiB | 2.73 MiB/s, done.
83 Total 65 (delta 32), reused 0 (delta 0), pack-reused 0
84 remote: Resolving deltas: 100% (32/32), completed with 7 local objects.
85 To https://github.com/cmestasz/fp2-23b.git
86 118a77f..abc9d8d main -> main
87
88 $ git add .
89
90 $ git commit -m "Inicio del juego vinculado entre ambas instancias"
91 [main a9a8d44] Inicio del juego vinculado entre ambas instancias
92 9 files changed, 132 insertions(+), 64 deletions(-)
93 rename fase03/proyecto_final/VIDEOGAME/src/FX/{ => MainGame}/Controller.java (94%)
94 rename fase03/proyecto_final/VIDEOGAME/src/FX/{ => MainGame}/Main.fxml (98%)
95 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/FX/test.fxml
96
97 $ git add .
98
99 $ git commit -m "Cambio de estructura y vinculo entre ambos controladores"
100 [main 0be0029] Cambio de estructura y vinculo entre ambos controladores
101 8 files changed, 77 insertions(+), 35 deletions(-)
102 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/FX/MainGame/Controller.java
103 rename fase03/proyecto_final/VIDEOGAME/src/FX/MainGame/{Main.fxml => Main Game.fxml} (98%)
104 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/FX/MainGame/MainGameController.java
105 rename fase03/proyecto_final/VIDEOGAME/src/FX/{MainMenu/MainMenuServer.java =>
    MainServer.java} (96%)
106 rename fase03/proyecto_final/VIDEOGAME/src/{FX/MainMenu => Utils}/Connection.java (96%)
107 rename fase03/proyecto_final/VIDEOGAME/src/{FX/MainMenu => Utils}/Operation.java (90%)
```

```
108 $ git add .
109
110
111 $ git commit -m "Comentarios descriptivos en las partes confusas"
112 [main 38e2aaf] Comentarios descriptivos en las partes confusas
113 3 files changed, 21 insertions(+), 12 deletions(-)
114
115 $ git add .
116
117 $ git commit -m "Estructura del controlador del juego principal"
118 [main 8c4d1bd] Estructura del controlador del juego principal
119 11 files changed, 86 insertions(+), 82 deletions(-)
120 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/FX/MainGame/Main Game.fxml
121 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/FX/MainGame/MainGame.fxml
122 rename fase03/proyecto_final/VIDEOGAME/src/FX/MainMenu/{Main Menu.fxml => MainMenu.fxml}
123 (100%)
124 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/4.png
125 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/barrack.png
126 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/barrack.psd
127 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/tile.png
128 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/tile.psd
129
130 $ git push
131 Enumerating objects: 72, done.
132 Counting objects: 100% (72/72), done.
133 Delta compression using up to 4 threads
134 Compressing objects: 100% (57/57), done.
135 Writing objects: 100% (61/61), 14.12 KiB | 2.02 MiB/s, done.
136 Total 61 (delta 29), reused 0 (delta 0), pack-reused 0
137 remote: Resolving deltas: 100% (29/29), completed with 5 local objects.
138 To https://github.com/cmestasz/fp2-23b.git
139 abc9d8d..8c4d1bd main -> main
140
141 $ git add .
142
143 $ git commit -m "Estructura visual del juego principal"
144 [main 49914f7] Estructura visual del juego principal
145 6 files changed, 124 insertions(+), 35 deletions(-)
146 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/FX/MainGame/Board.java
147
148 $ git add .
149
150 $ git commit -m "Modelo de base de datos y cambios en el menu principal"
151 [main 5b878e5] Modelo de base de datos y cambios en el menu principal
152 6 files changed, 135 insertions(+), 37 deletions(-)
153 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/Utils/DBConnector.java
154 rename fase03/proyecto_final/VIDEOGAME/src/Utils/{Connection.java => ServerConnection.java}
155 (91%)
156
157 $ git push
158 Enumerating objects: 48, done.
159 Counting objects: 100% (48/48), done.
160 Delta compression using up to 4 threads
161 Compressing objects: 100% (29/29), done.
162 Writing objects: 100% (31/31), 21.49 KiB | 4.30 MiB/s, done.
163 Total 31 (delta 13), reused 0 (delta 0), pack-reused 0
```

```
162 remote: Resolving deltas: 100% (13/13), completed with 9 local objects.
163 To https://github.com/cmestasz/fp2-23b.git
164    8c4d1bd..5b878e5  main -> main
165
166 $ git add .
167
168 $ git commit -m "Clase que permite enviar y solicitar datos a la base de datos"
169 [main e1375c5] Clase que permite enviar y solicitar datos a la base de datos
170    3 files changed, 74 insertions(+), 14 deletions(-)
171
172 $ git add .
173
174 $ git commit -m "Implementacion de la clase DBConnector"
175 [main acca9f6] Implementacion de la clase DBConnector
176    5 files changed, 101 insertions(+), 70 deletions(-)
177
178 $ git add .
179
180 $ git commit -m "Clases para el juego principal"
181 [main dfbc1be] Clases para el juego principal
182    17 files changed, 186 insertions(+), 23 deletions(-)
183    create mode 100644 fase03/proyecto_final/VIDEOGAME/src/FX/MainGame/BoardGUI.java
184    create mode 100644 fase03/proyecto_final/VIDEOGAME/src/FX/MainGame/Classes/Archer.java
185    create mode 100644 fase03/proyecto_final/VIDEOGAME/src/FX/MainGame/Classes/Knight.java
186    create mode 100644 fase03/proyecto_final/VIDEOGAME/src/FX/MainGame/Classes/Soldier.java
187    create mode 100644 fase03/proyecto_final/VIDEOGAME/src/FX/MainGame/Classes/Spearman.java
188    create mode 100644 fase03/proyecto_final/VIDEOGAME/src/FX/MainGame/Classes/Swordsman.java
189    create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/tile_archer.png
190    create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/tile_archer.psd
191    create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/tile_knight.png
192    create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/tile_knight.psd
193    create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/tile_spearman.png
194    create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/tile_spearman.psd
195    create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/tile_swordsman.png
196    create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/tile_swordsman.psd
197
198 $ git add .
199
200 $ git commit -m "Eleccion de reino y cambios en el menu principal"
201 [main 790dc08] Eleccion de reino y cambios en el menu principal
202    5 files changed, 119 insertions(+), 32 deletions(-)
203    create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/settings.png
204
205 $ git add .
206
207 $ git commit -m "El tablero ahora se conecta entre ambos jugadores"
208 [main 5f9c49f] El tablero ahora se conecta entre ambos jugadores
209    13 files changed, 227 insertions(+), 95 deletions(-)
210    delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/FX/MainGame/BoardGUI.java
211    create mode 100644 fase03/proyecto_final/VIDEOGAME/src/Utils/SerializableColor.java
212
213 $ git add .
214
215 $ git commit -m "Envio de mensajes y conexion en el juego principal"
216 [main 45de600] Envio de mensajes y conexion en el juego principal
217    10 files changed, 270 insertions(+), 75 deletions(-)
```

```
218 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/Utils/MainGameOperation.java
219 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/Utils/MainMenuOperation.java
220 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/Utils/Operation.java
221
222 $ git add .
223
224 $ git commit -m "Nuevo sistema de mensajes"
225 [main 5207444] Nuevo sistema de mensajes
226 3 files changed, 35 insertions(+), 16 deletions(-)
227
228 $ git add .
229
230 $ git commit -m "Chat de colores y mejor tratado"
231 [main 1afcb77] Chat de colores y mejor tratado
232 3 files changed, 67 insertions(+), 26 deletions(-)
233
234 $ git add .
235
236 $ git commit -m "Representacion visual de los soldados en ambos tableros"
237 [main cc5ebe5] Representacion visual de los soldados en ambos tableros
238 13 files changed, 185 insertions(+), 78 deletions(-)
239 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/Utils/BetterColor.java
240 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/Utils/SerializableColor.java
241 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/Utils/Tile.java
242 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/Utils/VideogameConstants.java
243 rename fase03/proyecto_final/VIDEOGAME/src/img/{tile.png => tile_tile.png} (100%)
244 rename fase03/proyecto_final/VIDEOGAME/src/img/{tile.psd => tile_tile.psd} (100%)
245
246 $ git push
247 Enumerating objects: 173, done.
248 Counting objects: 100% (173/173), done.
249 Delta compression using up to 4 threads
250 Compressing objects: 100% (145/145), done.
251 Writing objects: 100% (154/154), 247.29 KiB | 7.98 MiB/s, done.
252 Total 154 (delta 87), reused 0 (delta 0), pack-reused 0
253 remote: Resolving deltas: 100% (87/87), completed with 10 local objects.
254 To https://github.com/cmestasz/fp2-23b.git
255 5b878e5..cc5ebe5 main -> main
256
257 $ git add .
258
259 $ git commit -m "Implementacion de los tipos de soldado y sus posibles acciones"
260 [main 70d26d2] Implementacion de los tipos de soldado y sus posibles acciones
261 23 files changed, 52 insertions(+), 43 deletions(-)
262 rename fase03/proyecto_final/VIDEOGAME/src/{FX => }/MainServer.java (99%)
263 rename fase03/proyecto_final/VIDEOGAME/src/{FX => }/Videogame.java (98%)
264 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/action_charge.png
265 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/action_charge.psd
266 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/action_dismount.png
267 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/action_dismount.psd
268 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/action_mount.png
269 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/action_mount.psd
270 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/action_move.png
271 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/action_move.psd
272 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/action_schiltrom.png
273 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/action_schiltrom.psd
```



```
274 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/action_shoot.png
275 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/action_shoot.psd
276 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/action_sworddance.png
277 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/action_sworddance.psd
278
279 $ git add .
280
281 $ git commit -m "Implementacion de movimientos y ataques, se cancelaron bastantes planes que
    eran demasiado ambiciosos"
282 [main 4fc2a3f] Implementacion de movimientos y ataques, se cancelaron bastantes planes que
    eran demasiado ambiciosos
283 26 files changed, 320 insertions(+), 107 deletions(-)
284 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/action_attack.png
285 rename fase03/proyecto_final/VIDEOGAME/src/img/{action_shoot.psd => action_attack.psd} (61%)
286 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/action_charge.png
287 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/action_charge.psd
288 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/action_dismount.png
289 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/action_dismount.psd
290 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/action_mount.png
291 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/action_mount.psd
292 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/action_schiltrom.png
293 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/action_schiltrom.psd
294 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/action_shoot.png
295 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/action_sworddance.png
296 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/action_sworddance.psd
297
298 $ git add .
299
300 $ git commit -m "Primera prueba de toda la funcionalidad completa"
301 [main 6520ee5] Primera prueba de toda la funcionalidad completa
302 6 files changed, 85 insertions(+), 62 deletions(-)
303
304 $ git push
305 Enumerating objects: 98, done.
306 Counting objects: 100% (98/98), done.
307 Delta compression using up to 4 threads
308 Compressing objects: 100% (73/73), done.
309 Writing objects: 100% (76/76), 71.05 KiB | 5.46 MiB/s, done.
310 Total 76 (delta 46), reused 0 (delta 0), pack-reused 0
311 remote: Resolving deltas: 100% (46/46), completed with 13 local objects.
312 To https://github.com/cmestasz/fp2-23b.git
313    cc5ebe5..6520ee5  main -> main
314
315 $ git add .
316
317 $ git commit -m "Correccion de los ultimos errores"
318 [main 924da03] Correccion de los ultimos errores
319 12 files changed, 134 insertions(+), 72 deletions(-)
320
321 $ git push
322 Enumerating objects: 46, done.
323 Counting objects: 100% (46/46), done.
324 Delta compression using up to 4 threads
325 Compressing objects: 100% (23/23), done.
326 Writing objects: 100% (24/24), 25.04 KiB | 6.26 MiB/s, done.
327 Total 24 (delta 16), reused 0 (delta 0), pack-reused 0
```



```
328 remote: Resolving deltas: 100% (16/16), completed with 16 local objects.
329 To https://github.com/cmestasz/fp2-23b.git
330     6520ee5..924da03  main -> main
331
332 $ git add .
333
334 $ git commit -m "Implementacion de más utilidades"
335 [main ealidca9] Implementacion de más utilidades
336     5 files changed, 27 insertions(+), 13 deletions(-)
337
338 $ git push
339 Enumerating objects: 28, done.
340 Counting objects: 100% (28/28), done.
341 Delta compression using up to 4 threads
342 Compressing objects: 100% (14/14), done.
343 Writing objects: 100% (15/15), 4.20 KiB | 1.40 MiB/s, done.
344 Total 15 (delta 11), reused 0 (delta 0), pack-reused 0
345 remote: Resolving deltas: 100% (11/11), completed with 10 local objects.
346 To https://github.com/cmestasz/fp
347
348 $ git add .
349
350 $ git commit -m "Ultimas correcciones de la logica y funciones"
351 [main f0cf82b] Ultimas correcciones de la logica y funciones
352     2 files changed, 4 insertions(+), 1 deletion(-)
353
354 $ git push
355 Enumerating objects: 20, done.
356 Counting objects: 100% (20/20), done.
357 Delta compression using up to 4 threads
358 Compressing objects: 100% (10/10), done.
359 Writing objects: 100% (11/11), 3.40 KiB | 1.70 MiB/s, done.
360 Total 11 (delta 8), reused 0 (delta 0), pack-reused 0
361 remote: Resolving deltas: 100% (8/8), completed with 7 local objects.
362 To https://github.com/cmestasz/fp2-23b.git
363     ealidca9..f0cf82b  main -> main
364
365 $ git add .
366
367 $ git commit -m "Correcciones de la base de datos"
368 [main 12becf8] Correcciones de la base de datos
369     3 files changed, 15 insertions(+), 7 deletions(-)
370
371 $ git push
372 Enumerating objects: 23, done.
373 Counting objects: 100% (23/23), done.
374 Delta compression using up to 4 threads
375 Compressing objects: 100% (11/11), done.
376 Writing objects: 100% (12/12), 1.12 KiB | 1.12 MiB/s, done.
377 Total 12 (delta 8), reused 0 (delta 0), pack-reused 0
378 remote: Resolving deltas: 100% (8/8), completed with 8 local objects.
379 To https://github.com/cmestasz/fp2-23b.git
380     f0cf82b..12becf8  main -> main
381
382 $ git add .
383
```

```
384 $ git commit -m "Implementacion de mejoras visuales y forma final"
385 [main fd709bc] Implementacion de mejoras visuales y forma final
386 47 files changed, 155 insertions(+), 99 deletions(-)
387 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/White_Hammer.png
388 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/White_Moving.png
389 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/background_beach.png
390 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/background_beach.psd
391 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/background_data.png
392 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/background_data.psd
393 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/background_desert.png
394 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/background_desert.psd
395 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/background_forest.png
396 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/background_forest.psd
397 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/background_meadow.png
398 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/background_meadow.psd
399 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/background_mountain.png
400 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/background_mountain.psd
401 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/beach.jpg
402 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/desert.png
403 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/icon_big_attack.png
404 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/icon_big_defence.png
405 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/icon_big_helmet.png
406 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/icon_big_target.png
407 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/meadow.jpg
408 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/mountain.jpg
409
410 $ git push
411 Enumerating objects: 94, done.
412 Counting objects: 100% (94/94), done.
413 Delta compression using up to 4 threads
414 Compressing objects: 100% (58/58), done.
415 Writing objects: 100% (59/59), 43.14 MiB | 2.47 MiB/s, done.
416 Total 59 (delta 16), reused 0 (delta 0), pack-reused 0
417 remote: Resolving deltas: 100% (16/16), completed with 13 local objects.
418 To https://github.com/cmestasz/fp2-23b.git
419     12becf8..fd709bc  main -> main
420
421 $ git add .
422
423 $ git commit -m "Correcciones para el ejecutable y proyecto exportado"
424 [main 83baf9f] Correcciones para el ejecutable y proyecto exportado
425 9 files changed, 67 insertions(+), 25 deletions(-)
426 create mode 100644 fase03/proyecto_final/VIDEOGAME/SERVER.jar
427 create mode 100644 fase03/proyecto_final/VIDEOGAME/VIDEOGAME.jar
428 create mode 100644 fase03/proyecto_final/VIDEOGAME/data/dblogin.dat
429 create mode 100644 fase03/proyecto_final/VIDEOGAME/src/Main.java
430
431 $ git add .
432
433 $ git commit -m "Ejecutables del videojuego en una carpeta separada"
434 [main 68308c2] Ejecutables del videojuego en una carpeta separada
435 46 files changed, 569 insertions(+), 1 deletion(-)
436 create mode 100644 fase03/proyecto_final/EJECUTABLES/SERVER.jar
437 rename fase03/proyecto_final/{VIDEOGAME/SERVER.jar => EJECUTABLES/VIDEOGAME.jar} (77%)
438 create mode 100644 fase03/proyecto_final/EJECUTABLES/data/dblogin.dat
439 create mode 100644 fase03/proyecto_final/SERVER/.vscode/settings.json
```

```
440 create mode 100644 fase03/proyecto_final/SERVER/README.md
441 create mode 100644 fase03/proyecto_final/SERVER/SERVER.jar
442 create mode 100644 fase03/proyecto_final/SERVER/src/FX/MainGame/Board.java
443 create mode 100644 fase03/proyecto_final/SERVER/src/FX/MainGame/Classes/Archer.java
444 create mode 100644 fase03/proyecto_final/SERVER/src/FX/MainGame/Classes/Knight.java
445 create mode 100644 fase03/proyecto_final/SERVER/src/FX/MainGame/Classes/Soldier.java
446 create mode 100644 fase03/proyecto_final/SERVER/src/FX/MainGame/Classes/Spearman.java
447 create mode 100644 fase03/proyecto_final/SERVER/src/FX/MainGame/Classes/Swordsman.java
448 rename fase03/proyecto_final/{VIDEOGAME => SERVER}/src/MainServer.java (99%)
449 create mode 100644 fase03/proyecto_final/SERVER/src/Utils/BetterColor.java
450 create mode 100644 fase03/proyecto_final/SERVER/src/Utils/DBConnector.java
451 create mode 100644 fase03/proyecto_final/SERVER/src/Utils/MainGameOperation.java
452 create mode 100644 fase03/proyecto_final/SERVER/src/Utils/MainMenuOperation.java
453 rename fase03/proyecto_final/{VIDEOGAME => SERVER}/src/Utils/ServerConnection.java (100%)
454 create mode 100644 fase03/proyecto_final/SERVER/src/Utils/Utils.java
455 create mode 100644 fase03/proyecto_final/SERVER/src/Utils/VideogameConstants.java
456 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/White_Hammer.png
457 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/White_Moving.png
458 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/action_attack.psd
459 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/action_move.psd
460 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/background_beach.psd
461 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/background_data.psd
462 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/background_desert.psd
463 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/background_forest.psd
464 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/background_meadow.psd
465 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/background_mountain.psd
466 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/beach.jpg
467 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/desert.png
468 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/icon_big_attack.png
469 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/icon_big_defence.png
470 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/icon_big_helmet.png
471 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/icon_big_target.png
472 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/meadow.jpg
473 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/mountain.jpg
474 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/tile_archer.psd
475 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/tile_knight.psd
476 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/tile_spearman.psd
477 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/tile_swordsman.psd
478 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/tile_tile.psd
479 delete mode 100644 fase03/proyecto_final/VIDEOGAME/src/img/waiting.psd
480
481 $ git push
482 Enumerating objects: 62, done.
483 Counting objects: 100% (62/62), done.
484 Delta compression using up to 4 threads
485 Compressing objects: 100% (41/41), done.
486 Writing objects: 100% (48/48), 137.06 MiB | 1.74 MiB/s, done.
487 Total 48 (delta 15), reused 0 (delta 0), pack-reused 0
488 remote: Resolving deltas: 100% (15/15), completed with 7 local objects.
489 remote: warning: See https://gh.io/lfs for more information.
490 remote: warning: File fase03/proyecto_final/EJECUTABLES/VIDEOGAME.jar is 50.32 MB this is
    larger than GitHub's recommended maximum file size of 50.00 MB
491 remote: warning: File fase03/proyecto_final/VIDEOGAME/VIDEOGAME.jar is 86.40 MB this is
    larger than GitHub's recommended maximum file size of 50.00 MB
492 remote: warning: GH001: Large files detected. You may want to try Git Large File Storage -
    https://git-lfs.github.com.
```

Commit

.gitignore actualizado para solo publicar la carpeta src

Commit


Ejemplo dado en la documentacion de JavaFX


Commit

Menú principal

Commit

Controlador del menú principal

 main

 cimestasz committed last week


1 parent: e9a3e21 commit: 84ae869


☐ Showing 5 changed files with 131 additions and 14 deletions.

Controlador del menú principal

Commit

Modelo de servidor para manejar varias instancias locales del videojuego

 main

 cimestasz committed last week


1 parent: 84ae869 commit: 5e5d1a8


☐ Showing 4 changed files with 133 additions and 56 deletions.

Modelo de servidor para manejar varias instancias locales del videojuego

Commit

Servidor completo

 main

 cimestasz committed last week


1 parent: 5e5d1a8 commit: 1d626a9


Showing 1 changed file with 33 additions and 11 deletions.

Servidor completo

Commit

Funcionalidad de servidor para el menú principal completa

 main

 cimestasz committed last week


1 parent: 1d626a9 commit: abc9d8d


☐ Showing 5 changed files with 198 additions and 58 deletions.

Funcionalidad de servidor para el menú principal completa

Commit

Inicio del juego vinculado entre ambas instancias

 main

 cimestasz committed last week


1 parent: abc9d8d commit: a9a8044


☐ Showing 9 changed files with 132 additions and 64 deletions.

Inicio del juego vinculado entre ambas instancias

Commit

Cambio de estructura y vinculo entre ambos controladores

 main

 cimestasz committed last week

1 parent: a9a8044 commit: 0be0829

☐ Showing 8 changed files with 77 additions and 35 deletions.

Cambio de estructura y vinculo entre ambos controladores

Commit

Comentarios descriptivos en las partes confusas

main

cmestasz committed last week

1 parent: 0be8029 commit: 3862aef

Showing 3 changed files with 21 additions and 12 deletions.

Whitespace

Ignore whitespace

Split

Unified

Comentarios descriptivos en las partes confusas

Commit

Estructura del controlador del juego principal

main

cmestasz committed last week

1 parent: 38e2aef commit: 8c4d1bd

Showing 11 changed files with 86 additions and 82 deletions.

Whitespace

Ignore whitespace

Split

Unified

Estructura del controlador del juego principal

Commit

Estructura visual del juego principal

main

cmestasz committed last week

1 parent: 8c4d1bd commit: 49914f7

Showing 6 changed files with 124 additions and 35 deletions.

Whitespace

Ignore whitespace

Split

Unified

Estructura visual del juego principal

Commit

Modelo de base de datos y cambios en el menu principal

main

cmestasz committed last week

1 parent: 49914f7 commit: 5b878e5

Showing 6 changed files with 135 additions and 37 deletions.

Whitespace

Ignore whitespace

Split

Unified

Modelo de base de datos y cambios en el menu principal

Commit

Clase que permite enviar y solicitar datos a la base de datos

main

cmestasz committed 5 days ago

1 parent: 5b878e5 commit: e1375c5

Showing 3 changed files with 74 additions and 14 deletions.

Whitespace

Ignore whitespace

Split

Unified

Clase que permite enviar y solicitar datos a la base de datos

Commit

Implementacion de la clase DBConnector

main

cmestasz committed 5 days ago

1 parent: e1375c5 commit: acca9f6

Showing 5 changed files with 101 additions and 70 deletions.

Whitespace

Ignore whitespace

Split

Unified

Implementacion de la clase DBConnector


Christian Mestas


Fundamentos de la Programación 2

Página 14

Commit

Clases para el juego principal

 main

 cmestasz committed 5 days ago

1 parent: acca9f6 commit: d9fc1be

Showing 17 changed files with 186 additions and 23 deletions.

Whitespace

Ignore whitespace


Split


Unified

Clases para el juego principal

Commit

Eleccion de reino y cambios en el menu principal

 main

 cmestasz committed 5 days ago

1 parent: d9fc1be commit: 790dc98

Showing 5 changed files with 119 additions and 32 deletions.

Whitespace

Ignore whitespace


Split


Unified

Eleccion de reino y cambios en el menu principal

Commit

El tablero ahora se conecta entre ambos jugadores

 main

 cmestasz committed 4 days ago

1 parent: 790dc98 commit: 5f9c49f

Showing 13 changed files with 227 additions and 95 deletions.

Whitespace

Ignore whitespace


Split


Unified

El tablero ahora se conecta entre ambos jugadores

Commit

Envio de mensajes y conexion en el juego principal

 main

 cmestasz committed 4 days ago

1 parent: 5f9c49f commit: 45de088

Showing 10 changed files with 270 additions and 75 deletions.

Whitespace

Ignore whitespace


Split


Unified

Envio de mensajes y conexion en el juego principal

Commit

Nuevo sistema de mensajes

 main

 cmestasz committed 4 days ago

1 parent: 45de088 commit: 5207444

Showing 3 changed files with 35 additions and 16 deletions.

Whitespace

Ignore whitespace


Split


Unified

Nuevo sistema de mensajes

Commit

Chat de colores y mejor tratado

 main

 cmestasz committed 4 days ago

1 parent: 5207444 commit: 1afcb77

Showing 3 changed files with 67 additions and 26 deletions.

Whitespace

Ignore whitespace


Split


Unified

Chat de colores y mejor tratado


Commit

Representacion visual de los soldados en ambos tableros

 main

 cimestasz committed 4 days ago

1 parent: 1a6b777 commit: cc5eb65

 Showing 13 changed files with 185 additions and 78 deletions.

Whitespace

Ignore whitespace


Split


Unified

Representacion visual de los soldados en ambos tableros


Commit

Implementacion de los tipos de soldado y sus posibles acciones

 main

 cimestasz committed 2 days ago

1 parent: cc5eb65 commit: 78b26d2

 Showing 23 changed files with 52 additions and 43 deletions.

Whitespace

Ignore whitespace


Split


Unified

Implementacion de los tipos de soldado y sus posibles acciones


Commit

Implementacion de movimientos y ataques, se cancelaron bastantes plan...
...es que eran demasiado ambiciosos

 main

 cimestasz committed 2 days ago

1 parent: 78b26d2 commit: 4fc2a3f

 Showing 26 changed files with 320 additions and 107 deletions.

Whitespace

Ignore whitespace


Split


Unified

Implementacion de movimientos y ataques, se cancelaron bastantes planes que eran demasiado ambiciosos


Commit

Primera prueba de toda la funcionalidad completa

 main

 cimestasz committed 2 days ago

1 parent: 4fc2a3f commit: 6528ee5

 Showing 6 changed files with 85 additions and 62 deletions.

Whitespace

Ignore whitespace


Split


Unified

Primera prueba de toda la funcionalidad completa


Commit

Correccion de los ultimos errores

 main

 cimestasz committed yesterday

1 parent: 6528ee5 commit: 924da83

 Showing 12 changed files with 134 additions and 72 deletions.

Whitespace

Ignore whitespace


Split


Unified

Correccion de los ultimos errores


Commit

Implementacion de más utilidades

 main

 cimestasz committed yesterday

1 parent: 924da83 commit: ea10ca9

 Showing 5 changed files with 27 additions and 13 deletions.

Whitespace

Ignore whitespace


Split


Unified

Implementacion de más utilidades

Commit

Ultimas correcciones de la logica y funciones

 main

 cimestasz committed yesterday

1 parent: ea1dca9 commit: f8cf82b

Showing 2 changed files with 4 additions and 1 deletion.

Whitespace

Ignore whitespace


Split


Unified

Ultimas correcciones de la logica y funciones

Commit

Correcciones de la base de datos

 main

 cimestasz committed yesterday

1 parent: f8cf82b commit: 12becf8

Showing 3 changed files with 15 additions and 7 deletions.

Whitespace

Ignore whitespace


Split


Unified

Correcciones de la base de datos

Commit

Implementacion de mejoras visuales y forma final

 main

 cimestasz committed 10 hours ago

1 parent: 12becf8 commit: fd789dc

Showing 47 changed files with 155 additions and 99 deletions.

Whitespace

Ignore whitespace


Split


Unified

Implementacion de mejoras visuales y forma final

Commit

Correcciones para el ejecutable y proyecto exportado

 main

 cimestasz committed 1 hour ago

1 parent: fd789dc commit: 83baef9f

Showing 9 changed files with 67 additions and 25 deletions.

Whitespace

Ignore whitespace


Split


Unified

Correcciones para el ejecutable y proyecto exportado

Commit

Ejecutables del videojuego en una carpeta separada

 main

 cimestasz committed 1 hour ago

1 parent: 83baef9f commit: 68388c2

Showing 46 changed files with 569 additions and 1 deletion.

Whitespace

Ignore whitespace


Split


Unified

Ejecutables del videojuego en una carpeta separada

Commit

Sistema de guardado y cargado de partidas

 main

 cimestasz committed 1 minute ago

1 parent: df9144e commit: 2186d87

Showing 13 changed files with 93 additions and 16 deletions.

Whitespace

Ignore whitespace

Split

Unified

Sistema de guardado y cargado de partidas

5. Código desarrollado

5.1. Servidor

MainServer.java

```
1
2 import Utils.*;
3 import java.io.*;
4 import java.util.*;
5 import javax.swing.JOptionPane;
6 import FX.MainGame.Board;
7
8 public class MainServer extends Thread implements MainMenuOperation, MainGameOperation {
9     private ArrayList<ServerConnection> connectionsList = new ArrayList<ServerConnection>();
10    private HashMap<Integer, Long> lastModifiedMap = new HashMap<Integer, Long>(); // Guarda
        CUALQUIER archivo que es recibido o cambiado
11    private int totalConnections;
12    private boolean active = true;
13    private int tickRate;
14
15    private HashMap<String, int[]> matches = new HashMap<String, int[]>();
16
17    public MainServer(int tickRate) {
18        this.tickRate = tickRate;
19    }
20
21    public void run() {
22        File directory = new File("connections");
23        if (!directory.exists()) {
24            directory.mkdirs();
25        }
26        try {
27            while (active) {
28                int newTotalConnections = 0;
29                for (File file : directory.listFiles())
30                    if (file.getName().endsWith(".dat"))
31                        newTotalConnections++;
32                System.out.println(newTotalConnections);
33
34                if (totalConnections != newTotalConnections) {
35                    for (int id = totalConnections; id < newTotalConnections; id++) {
36                        // Se crea una nueva conexión y se agrega a la lista.
37                        ServerConnection connection = new ServerConnection(totalConnections);
38                        System.out.println("connecting: " + connection);
39                        connectionsList.add(connection);
40                        lastModifiedMap.put(totalConnections, connection.getLastModified());
41                    }
42                    totalConnections = newTotalConnections;
43                }
44
45                for (int id = 0; id < totalConnections; id++)
46                    respond(id);
47
48                sleep(tickRate);
49            }
50        } catch (Exception e) {
51            e.printStackTrace();
52        }
53    }
54 }
```

```
50     }
51
52     for (ServerConnection connection : connectionsList) {
53         connection.deleteDataConnection();
54         connection = null;
55     }
56
57     } catch (Exception e) {
58         e.printStackTrace();
59     }
60 }
61
62 public void end() {
63     active = false;
64 }
65
66 private void respond(int id) {
67     ServerConnection connection = connectionsList.get(id);
68     System.out.println("responding: " + connection);
69     long lastModified = connection.getLastModified();
70
71     // Se verifica si la conexión ha sido modificada desde la última respuesta.
72     if (lastModifiedMap.get(id) != lastModified) {
73         try {
74             if (!connection.isInitialized())
75                 connection.initialize();
76
77             DataInputStream in = connection.getDataInputStream();
78             int operation = in.readInt();
79             String code = Utils.readString(in);
80
81             int[] ids;
82             int idOther;
83             ServerConnection other;
84             DataOutputStream toHost;
85             DataOutputStream toGuest;
86             DataOutputStream toOther;
87             switch (operation) {
88                 case OPERATION_CREATE:
89                     // Se almacena la información de la conexión que ha creado un nuevo
90                     // código.
91                     matches.put(code, new int[] { connection.getId(), -1 });
92                     lastModifiedMap.put(id, lastModified);
93                     break;
94
95                 case OPERATION_JOIN:
96                     // Se intenta unir dos conexiones con el código.
97                     Utils.readString(in);
98                     Utils.readString(in);
99                     int otherId = in.readInt();
100                     ids = matches.get(code);
101                     toGuest = connection.getDataOutputStream();
102                     toGuest.writeInt(RESPONSE_GUEST);
103                     if (ids != null && ids[1] == -1) {
104                         ServerConnection host = connectionsList.get(ids[0]);
105                         ids[1] = connection.getId();
```

```
105         Utils.writeString(toGuest, host.getName());
106         Utils.writeString(toGuest, host.getKingdom());
107
108         toHost = host.getDataOutputStream();
109         toHost.writeInt(RESPONSE_HOST);
110         Utils.writeString(toHost, connection.getName());
111         Utils.writeString(toHost, connection.getKingdom());
112         toHost.writeInt(otherId);
113
114         toHost.close();
115         lastModifiedMap.put(host.getId(), host.getLastModified());
116     }
117     toGuest.writeChar(0);
118     toGuest.close();
119     lastModifiedMap.put(id, connection.getLastModified());
120     break;
121
122     case OPERATION_START:
123         // Se inicia la conexión del invitado con el código.
124         ObjectInputStream inObj = connection.getObjectInputStream();
125         Board board = (Board) inObj.readObject();
126         board.invertBoard();
127         inObj.close();
128         connection.deleteObjConnection();
129
130         int idGuest = matches.get(code)[1];
131         ServerConnection guest = connectionsList.get(idGuest);
132
133         toGuest = guest.getDataOutputStream();
134         ObjectOutputStream toGuestObj = guest.getOutputStream();
135         toGuest.writeInt(RESPONSE_START);
136         toGuestObj.writeObject(board);
137         toGuest.close();
138         toGuestObj.close();
139
140         lastModifiedMap.put(idGuest, guest.getLastModified());
141         lastModifiedMap.put(id, connection.getLastModified());
142         break;
143
144     case OPERATION_CHAT:
145         String message = Utils.readString(in);
146         message.replaceAll("\n", "");
147         ids = matches.get(code);
148         idOther = id == ids[0] ? ids[1] : ids[0];
149         other = connectionsList.get(idOther);
150
151         toOther = other.getDataOutputStream();
152         toOther.writeInt(RESPONSE_CHAT);
153         Utils.writeString(toOther, message);
154         toOther.close();
155
156         lastModifiedMap.put(id, connection.getLastModified());
157         lastModifiedMap.put(idOther, other.getLastModified());
158         break;
159
160     case OPERATION_MOVE:
```

```

161         case OPERATION_ATTACK:
162             int sI = in.readInt();
163             int sJ = in.readInt();
164             int oI = in.readInt();
165             int oJ = in.readInt();
166
167             ids = matches.get(code);
168             idOther = id == ids[0] ? ids[1] : ids[0];
169             other = connectionsList.get(idOther);
170
171             toOther = other.getDataOutputStream();
172             toOther.writeInt(operation == OPERATION_MOVE ? RESPONSE_MOVE :
173                             RESPONSE_ATTACK);
174             Utils.writeIdxs(toOther, sI, sJ, oI, oJ);
175             toOther.close();
176
177             lastModifiedMap.put(id, connection.getLastModified());
178             lastModifiedMap.put(idOther, other.getLastModified());
179             break;
180         }
181         in.close();
182     } catch (Exception e) {
183         e.printStackTrace();
184     }
185 }
186 }
187
188 public static void main(String[] args) {
189     new DBConnector();
190     int tickRate = Integer.parseInt(JOptionPane.showInputDialog("Ingresa el tiempo entre
191                             ticks (en milisegundos):"));
192     MainServer server = new MainServer(tickRate);
193     server.start();
194     JOptionPane.showMessageDialog(null, "El servidor esta ejecutandose
195                             correctamente\nPresione ok para detenerlo");
196     server.end();
197 }

```

- Clase que se encarga de recibir las peticiones de los clientes y responderlas.
- Al momento de abrir una instancia del videojuego, se crea una conexión que se conecta con el servidor.
- Mediante el uso de hilos, se responde todas las peticiones de los clientes.

DBConnector.java

```

1 package Utils;
2
3 import java.io.*;
4 import java.sql.*;
5
6 public class DBConnector {
7     private final String url = "jdbc:mysql://localhost:3306/fp2_23b";

```

```
8 private final String user = "fp2_23b";
9 private final String password = "12345678";
10 private Connection connection;
11
12 public static void main(String[] args) {
13     new DBConnector();
14 }
15
16 public DBConnector() {
17     try {
18         Class.forName("com.mysql.cj.jdbc.Driver");
19
20         connection = DriverManager.getConnection(url, user, password);
21         if (!checkInitialized()) {
22             initDatabase();
23             writeInitialized();
24         }
25
26         System.out.println("Conexion exitosa");
27     } catch (Exception e) {
28         e.printStackTrace();
29     }
30 }
31
32 private boolean checkInitialized() {
33     File initFile = new File("data/dbinit.dat");
34     return initFile.exists();
35 }
36
37 private void writeInitialized() throws IOException {
38     File dir = new File("data/");
39     if (!dir.exists())
40         dir.mkdirs();
41     File initFile = new File("data/dbinit.dat");
42     initFile.createNewFile();
43 }
44
45 private void initDatabase() throws SQLException {
46     connection.prepareStatement(
47         "CREATE TABLE players_videogame (id int NOT NULL AUTO_INCREMENT, name
48         varchar(30) NOT NULL, password varchar(30) NOT NULL, PRIMARY KEY (id))")
49         .execute();
50     connection.prepareStatement(
51         "CREATE TABLE matches_videogame (id int NOT NULL AUTO_INCREMENT, winner_id int
52         NOT NULL, loser_id int NOT NULL, PRIMARY KEY (id), INDEX winner_id
53         (winner_id), INDEX loser_id (loser_id))")
54         .execute();
55     connection.prepareStatement(
56         "ALTER TABLE matches_videogame ADD CONSTRAINT winner_id FOREIGN KEY
57         (winner_id) REFERENCES players_videogame(id) ON DELETE CASCADE ON UPDATE
58         CASCADE")
59         .execute();
60     connection.prepareStatement(
61         "ALTER TABLE matches_videogame ADD CONSTRAINT loser_id FOREIGN KEY (loser_id)
62         REFERENCES players_videogame(id) ON DELETE CASCADE ON UPDATE CASCADE")
63         .execute();
64 }
```



```
58     }
59
60     public int loginPlayer(String name, String password) {
61         try {
62             String query = String.format("SELECT id FROM players_videogame WHERE name = '%s'
63                                     AND password = '%s'", name,
64                                     password);
65             ResultSet results = connection.prepareStatement(query).executeQuery();
66             if (results.next())
67                 return results.getInt(1);
68         } catch (Exception e) {
69             e.printStackTrace();
70         }
71         return -1;
72     }
73
74     public int[] getWinsLoses(int id) {
75         try {
76             int[] totals = new int[2];
77             String query = String.format("SELECT COUNT(*) from matches_videogame WHERE
78                                     winner_id = '%d'", id);
79             ResultSet results = connection.prepareStatement(query).executeQuery();
80             if (results.next())
81                 totals[0] = results.getInt(1);
82
83             query = String.format("SELECT COUNT(*) from matches_videogame WHERE loser_id =
84                                     '%d'", id);
85             results = connection.prepareStatement(query).executeQuery();
86             if (results.next())
87                 totals[1] = results.getInt(1);
88
89             return totals;
90         } catch (Exception e) {
91             e.printStackTrace();
92             return null;
93         }
94     }
95
96     public void registerPlayer(String name, String password) {
97         try {
98             String query = String.format("INSERT INTO players_videogame (name, password)
99                                     VALUES ('%s', '%s')", name, password);
100             connection.prepareStatement(query).execute();
101         } catch (Exception e) {
102             e.printStackTrace();
103         }
104     }
105
106     public void createMatch(int winner_id, int loser_id) {
107         try {
108             String query = String.format("INSERT INTO matches_videogame (winner_id, loser_id)
109                                     VALUES ('%d', '%d')", winner_id,
110                                     loser_id);
111             System.out.println(query);
112             connection.prepareStatement(query).execute();
113         } catch (Exception e) {
```

```
109         e.printStackTrace();
110     }
111 }
112 }
```

- Clase que se encarga de conectarse a la base de datos y realizar las operaciones.
- La primera vez que se inicia, crea la base de datos y guarda el usuario que puede acceder en un archivo.
- Posee todos los métodos que permiten interactuar con la base de datos a lo largo de todo el juego.

ServerConnection.java

```
1 package Utils;
2
3 import java.io.*;
4
5 public class ServerConnection {
6     private int id;
7     private String name;
8     private String kingdom;
9     private File connectionDataFile;
10    private File connectionObjFile;
11    private boolean initialized;
12
13    public ServerConnection(int id) {
14        this.id = id;
15        this.connectionDataFile = new File("connections/" + id + ".dat");
16        this.connectionObjFile = new File("connections/" + id + ".obj");
17    }
18
19    public void initialize() throws IOException {
20        DataInputStream in = getDataInputStream();
21        in.readInt();
22        Utils.readString(in);
23        this.name = Utils.readString(in);
24        this.kingdom = Utils.readString(in);
25        in.close();
26        initialized = true;
27    }
28
29    public int getId() {
30        return id;
31    }
32
33    public String getName() {
34        return name;
35    }
36
37    public String getKingdom() {
38        return kingdom;
39    }
40
41    public boolean isInitialized() {
```

```
42     return initialized;
43 }
44
45 public File getConnectionDataFile() {
46     return connectionDataFile;
47 }
48
49 public File getConnectionObjFile() {
50     return connectionObjFile;
51 }
52
53 public long getLastModified() {
54     return connectionDataFile.lastModified();
55 }
56
57 public DataInputStream getDataInputStream() throws IOException {
58     return new DataInputStream(new FileInputStream(connectionDataFile));
59 }
60
61 public ObjectInputStream getObjectInputStream() throws IOException {
62     return new ObjectInputStream(new FileInputStream(connectionObjFile));
63 }
64
65 public DataOutputStream getDataOutputStream() throws IOException {
66     return new DataOutputStream(new FileOutputStream(connectionDataFile));
67 }
68
69 public ObjectOutputStream getObjectOutputStream() throws IOException {
70     return new ObjectOutputStream(new FileOutputStream(connectionObjFile));
71 }
72
73 public void deleteDataConnection() {
74     connectionDataFile.delete();
75 }
76
77 public void deleteObjConnection() {
78     connectionObjFile.delete();
79 }
80
81 public String toString() {
82     return id + ": " + name;
83 }
84
85 }
```

- Clase de utilidad para mantener una conexión con el servidor.
- Mantiene la id de la conexión, el nombre y el reino del jugador.
- Permite generar lectores y escritores de archivos para realizar el envío de datos.

5.2. Videojuego

Archer.java

```
1 package FX.MainGame.Classes;
2
3 public class Archer extends Soldier {
4     private static final int HEALTH = 5;
5     private static final int ATTACK = 7;
6     private static final int DEFENSE = 3;
7     private int arrows = 10;
8
9     public Archer(String name, int team, String type, String typeFile) {
10         super(name, team, HEALTH, ATTACK, DEFENSE, type, typeFile);
11     }
12
13     // Lamentablemente no llegare a implementar esto
14     public void shoot(Soldier other) {
15         attack(other);
16         arrows--;
17     }
18
19     public int getArrows() {
20         return arrows;
21     }
22 }
```

- Clase que almacena un arquero, y sus estadísticas.
- Los arqueros tiene un mayor rango de ataque.

Knight.java

```
1 package FX.MainGame.Classes;
2
3 public class Knight extends Soldier {
4     private static final int HEALTH = 3;
5     private static final int ATTACK = 10;
6     private static final int DEFENSE = 7;
7     private boolean mounted = false;
8
9     public Knight(String name, int team, String type, String typeFile) {
10         super(name, team, HEALTH, ATTACK, DEFENSE, type, typeFile);
11     }
12
13     // Lamentablemente no llegare a implementar esto
14     public void mount() {
15         mounted = true;
16         modifyAttack(1);
17         modifyDefense(-1);
18     }
19
20     // Lamentablemente no llegare a implementar esto
21     public void dismount() {
22         mounted = false;
23         modifyAttack(-1);
24         modifyDefense(1);
25     }
26
27     public boolean isMounted() {
```

```
28     return mounted;
29 }
30 }
```

- Clase que almacena un caballero, y sus estadísticas.
- Los arqueros tiene un mayor rango de movimiento.

Spearman.java

```
1 package FX.MainGame.Classes;
2
3 public class Spearman extends Soldier {
4     private static final int HEALTH = 8;
5     private static final int ATTACK = 5;
6     private static final int DEFENSE = 10;
7
8     public Spearman(String name, int team, String type, String typeFile) {
9         super(name, team, HEALTH, ATTACK, DEFENSE, type, typeFile);
10    }
11
12    // Lamentablemente no llegare a implementar esto
13    public void schiltrom() {
14        modifyDefense(1);
15    }
16 }
```

- Clase que almacena un lancero, y sus estadísticas.

Swordsman.java

```
1 package FX.MainGame.Classes;
2
3 public class Swordsman extends Soldier {
4     private static final int HEALTH = 10;
5     private static final int ATTACK = 10;
6     private static final int DEFENSE = 8;
7
8     public Swordsman(String name, int team, String type, String typeFile) {
9         super(name, team, HEALTH, ATTACK, DEFENSE, type, typeFile);
10    }
11
12    // Lamentablemente no llegare a implementar esto
13    public void swordDance() {
14        modifyAttack(1);
15    }
16 }
```

- Clase que almacena un espadachín, y sus estadísticas.

Soldier.java

```
1 package FX.MainGame.Classes;
```

```
2
3 import java.io.Serializable;
4
5 public abstract class Soldier implements Serializable {
6     private String name;
7     private int team;
8     private int initialHealth;
9     private int currentHealth;
10    private int attack;
11    private int defense;
12    private String type;
13    private String typeFile;
14
15    public Soldier(String name, int team, int initialHealth, int attack, int defense, String
        type, String typeFile) {
16        this.name = name;
17        this.team = team;
18        this.initialHealth = initialHealth;
19        this.currentHealth = initialHealth;
20        this.attack = attack;
21        this.defense = defense;
22        this.type = type;
23        this.typeFile = typeFile;
24    }
25
26    public String getName() {
27        return name;
28    }
29
30    public int getTeam() {
31        return team;
32    }
33
34    public int getInitialHealth() {
35        return initialHealth;
36    }
37
38    public int getCurrentHealth() {
39        return currentHealth;
40    }
41
42    public int getAttack() {
43        return attack;
44    }
45
46    public int getDefense() {
47        return defense;
48    }
49
50    public String getType() {
51        return type;
52    }
53
54    public String getTypeFile() {
55        return typeFile;
56    }
```

```
57
58     public void heal() {
59         currentHealth++;
60     }
61
62     public int attack(Soldier other) {
63         int damage = Math.max(1, attack - other.getDefense() / 2);
64         other.hurt(damage);
65         return damage;
66     }
67
68     public void hurt(int damage) {
69         currentHealth -= damage;
70     }
71
72     public void modifyAttack(int change) {
73         attack += change;
74     }
75
76     public void modifyDefense(int change) {
77         defense += change;
78     }
79
80     public String toString() {
81         return name;
82     }
83 }
```

- Clase que almacena un soldado.
- Superclase de todas las clases de soldados.
- Almacena nombre, equipo, vida, ataque, defensa, y tipo.
- Posee métodos para simular el comportamiento de un soldado.

Board.java

```
1 package FX.MainGame;
2
3 import FX.MainGame.Classes.*;
4 import Utils.*;
5 import java.io.Serializable;
6 import java.util.*;
7
8 public class Board implements Serializable, VideogameConstants {
9     private final Random RANDOM = new Random();
10
11     private String terrain;
12     private String terrainFile;
13     private BetterColor background;
14     private HashMap<String, Soldier> army1 = new HashMap<String, Soldier>();
15     private HashMap<String, Soldier> army2 = new HashMap<String, Soldier>();
16     private String kingdomPlayer;
17     private String kingdomEnemy;
18 }
```



```
19 public Board(String kingdom1, String kingdom2) {
20     int idxTerrain = RANDOM.nextInt(TERRAINS.length);
21     terrain = TERRAINS[idxTerrain];
22     terrainFile = TERRAIN_FILES[idxTerrain];
23
24     this.kingdomPlayer = kingdom1;
25     this.kingdomEnemy = kingdom2;
26
27     initSoldiers(army1, 1);
28     initSoldiers(army2, 2);
29 }
30
31 public void invertBoard() {
32     HashMap<String, Soldier> armyt = army1;
33     army1 = army2;
34     army2 = armyt;
35
36     String kingdomt = kingdomPlayer;
37     kingdomPlayer = kingdomEnemy;
38     kingdomEnemy = kingdomt;
39 }
40
41 public HashMap<String, Soldier> getArmy1() {
42     return army1;
43 }
44
45 public HashMap<String, Soldier> getArmy2() {
46     return army2;
47 }
48
49 public String getTerrain() {
50     return terrain;
51 }
52
53 public String getTerrainFile() {
54     return terrainFile;
55 }
56
57 public String getKingdomPlayer() {
58     return kingdomPlayer;
59 }
60
61 public String getKingdomEnemy() {
62     return kingdomEnemy;
63 }
64
65 public BetterColor getBackground() {
66     return background;
67 }
68
69 private void initSoldiers(HashMap<String, Soldier> map, int team) {
70     for (int i = 0; i < TOTAL_SOLDIERS; i++) {
71         int idx = RANDOM.nextInt(TYPES.length);
72         String type = TYPES[idx];
73         String fileType = TYPE_FILES[idx];
74         int row, col;
```

```
75     String key;
76     do {
77         row = RANDOM.nextInt(SIZE);
78         col = RANDOM.nextInt(SIZE);
79         key = generateKey(row, col);
80     } while (army1.containsKey(key) || army2.containsKey(key));
81     String name = type + i + "X" + team;
82     Soldier soldier = null;
83     switch (type) {
84         case "CABALLERO":
85             soldier = new Knight(name, team, type, fileType);
86             break;
87         case "ARQUERO":
88             soldier = new Archer(name, team, type, fileType);
89             break;
90         case "ESPADACHIN":
91             soldier = new Swordsman(name, team, type, fileType);
92             break;
93         case "LANCERO":
94             soldier = new Spearman(name, team, type, fileType);
95             break;
96     }
97     map.put(key, soldier);
98 }
99 }
100
101 private String generateKey(int i, int j) {
102     return i + "," + j;
103 }
104
105 public String toString() {
106     return "a board!";
107 }
108 }
```

- Clase que almacena un tablero.
- Almacena el terreno, los ejércitos y los reinos.
- Es serializable para permitir ser enviado entre ambos jugadores al momento de iniciar el juego.

MainGameController.java

```
1 package FX.MainGame;
2
3 import java.io.*;
4 import java.util.HashMap;
5 import FX.MainGame.Classes.Soldier;
6 import FX.MainMenu.MainMenuController;
7 import Utils.*;
8 import javafx.application.Platform;
9 import javafx.collections.ObservableList;
10 import javafx.fxml.FXML;
11 import javafx.scene.Node;
12 import javafx.scene.control.*;
13 import javafx.scene.image.*;
```

```
14 import javafx.scene.input.MouseEvent;
15 import javafx.scene.layout.*;
16 import javafx.scene.text.*;
17 import javafx.stage.Stage;
18
19 public class MainGameController implements MainGameOperation, VideogameConstants {
20     private Stage gameStage;
21     private Stage menuStage;
22     private Resolution resolution;
23     private int width;
24     private int height;
25     private MainMenuController menuController;
26     private Board board;
27     private String kingdomPlayer;
28     private String kingdomEnemy;
29     private File connectionFile;
30     private String path;
31     private int idConnection;
32     private int idPlayer;
33     private int idEnemy;
34     private DataReceiver dataReceiver;
35     private String matchCode;
36     private String pName;
37     private String eName;
38     private Tile[][] tiles = new Tile[SIZE][SIZE];
39     private String selectedAction = "MOVER";
40     private Tile selectedTile;
41     private HashMap<String, Soldier> army1;
42     private HashMap<String, Soldier> army2;
43     private DBConnector dbConnector;
44     private boolean gameEnded;
45     private boolean playerTurn = true;
46
47     @FXML
48     private GridPane uiBoard;
49     @FXML
50     private ImageView boardBackground;
51     @FXML
52     private ImageView dataBackground;
53     @FXML
54     private TextArea playerData;
55     @FXML
56     private TextArea enemyData;
57     @FXML
58     private ScrollPane chatOutputPane;
59     @FXML
60     private VBox chatOutput;
61     @FXML
62     private TextField chatInput;
63     @FXML
64     private TilePane actionsPane;
65     @FXML
66     private Pane messagePane;
67     @FXML
68     private TextArea messageOutput;
69     @FXML
```

```
70     private VBox moveActionPane;
71     @FXML
72     private VBox attackActionPane;
73     @FXML
74     private TextField fileNameInput;
75
76     public void init(MainMenuController menuController, Resolution resolution, Stage
77         menuStage, Stage gameStage,
78         Board board,
79         int idConnection, String matchCode, String pName, String eName, int idPlayer, int
80         idEnemy) {
81         this.menuController = menuController;
82         this.resolution = resolution;
83         this.width = resolution.getWidth();
84         this.height = resolution.getHeight();
85         this.menuStage = menuStage;
86         this.gameStage = gameStage;
87         this.board = board;
88         army1 = board.getArmy1();
89         army2 = board.getArmy2();
90         this.idConnection = idConnection;
91         this.kingdomPlayer = board.getKingdomPlayer();
92         this.kingdomEnemy = board.getKingdomEnemy();
93         this.matchCode = matchCode;
94         this.pName = pName;
95         this.eName = eName;
96         this.idPlayer = idPlayer;
97         this.idEnemy = idEnemy;
98
99         initButtons();
100        initBackground();
101        initDataFields();
102        initChat();
103
104        actionsPane.setPrefWidth(width * 0.15);
105        actionsPane.setPrefHeight(width * 0.05);
106        setStyleColor(moveActionPane, SELECTED_COLOR);
107
108        dbConnector = new DBConnector();
109        setConnection();
110    }
111
112    public void initialize() {
113    }
114
115    public void sendMessage() {
116        String message = String.format("%s: %s\n", pName, chatInput.getText());
117        printMessage(message, PLAYER_COLOR);
118        try {
119            DataOutputStream out = new DataOutputStream(new FileOutputStream(connectionFile));
120            out.writeInt(OPERATION_CHAT);
121            Utils.writeStrings(out, new String[] { matchCode, message });
122            out.close();
123        } catch (Exception e) {
124            e.printStackTrace();
125        }
126    }
127}
```

```
124     }
125     chatInput.setText("");
126 }
127
128 public void setActionMove() {
129     setStyleColor(moveActionPane, SELECTED_COLOR);
130     setStyleColor(attackActionPane, null);
131     selectedAction = "MOVER";
132 }
133
134 public void setActionAttack() {
135     setStyleColor(attackActionPane, SELECTED_COLOR);
136     setStyleColor(moveActionPane, null);
137     selectedAction = "ATACAR";
138 }
139
140 public void closeMessage() {
141     messagePane.setVisible(false);
142     if (gameEnded) {
143         dataReceiver.endGame();
144         menuStage.show();
145         menuController.restartMenu();
146         gameStage.close();
147     }
148 }
149
150 private void initButtons() {
151     for (int i = 0; i < SIZE; i++) {
152         for (int j = 0; j < SIZE; j++) {
153             String key = generateKey(i, j);
154             double size = 1.0 * resolution.getHeight() / SIZE;
155             HashMap<String, Soldier> army1 = board.getArmy1();
156             HashMap<String, Soldier> army2 = board.getArmy2();
157             Tile tile;
158             Soldier soldier;
159
160             if (army1.containsKey(key)) {
161                 soldier = army1.get(key);
162                 tile = new Tile(soldier.getCurrentHealth(), soldier.getTypeFile(), size,
163                     i, j);
164                 setStyleColor(tile, PLAYER_COLOR_TRANS);
165             } else if (army2.containsKey(key)) {
166                 soldier = army2.get(key);
167                 tile = new Tile(soldier.getCurrentHealth(), soldier.getTypeFile(), size,
168                     i, j);
169                 setStyleColor(tile, ENEMY_COLOR_TRANS);
170             } else {
171                 tile = new Tile(0, "tile", size, i, j);
172             }
173             tiles[i][j] = tile;
174
175             tile.setOnMouseClicked(this::handleClick);
176             uiBoard.add(tile, i, j);
177         }
178     }
179 }
```

```
178 public void saveMatch() {
179     try {
180         String fileName = fileNameInput.getText();
181         ObjectOutputStream out = new ObjectOutputStream(new
182             FileOutputStream(String.format("data/%s.sav", fileName)));
183         out.writeObject(board);
184         out.close();
185
186         showMessage("Partida guardada correctamente");
187     } catch (Exception e) {
188         e.printStackTrace();
189     }
190 }
191
192 private void setStyleColor(Region pane, BetterColor color) {
193     if (color == null) {
194         pane.setStyle("-fx-background-color: none;");
195     } else {
196         pane.setStyle(String.format("-fx-background-color: %s;", color.getRGBA()));
197     }
198 }
199
200 private void initBackground() {
201     boardBackground.setFitWidth(width);
202     boardBackground.setFitHeight(height);
203     boardBackground.setImage(new Image(String.format("img/background_%s.png",
204         board.getTerrainFile())));
205
206     dataBackground.setFitWidth(width - height);
207     dataBackground.setFitHeight(height);
208     dataBackground.setImage(new Image("img/background_data.png"));
209 }
210
211 private void initDataFields() {
212     playerData.setText(String.format("%s: %s%n", pName, kingdomPlayer));
213     enemyData.setText(String.format("%s: %s%n", eName, kingdomEnemy));
214 }
215
216 private void initChat() {
217     chatOutput.setPrefHeight(height * 0.4);
218 }
219
220 private void handleClick(MouseEvent event) {
221     Tile tile = (Tile) event.getSource();
222
223     if (!tryDoAction(tile)) {
224         if (board.getArmy1().containsKey(tile.getKey())) {
225             selectedTile = tile;
226             showActionsMenu();
227         } else {
228             selectedTile = null;
229             removeActionsMenu();
230         }
231     }
232 }
```

```
232
233     private void showActionsMenu() {
234         actionsPane.setVisible(true);
235     }
236
237     private void removeActionsMenu() {
238         actionsPane.setVisible(false);
239     }
240
241     private boolean tryDoAction(Tile otherTile) {
242         if (selectedTile != null && playerTurn) {
243             String otherKey = otherTile.getKey();
244             try {
245                 DataOutputStream out;
246                 int sI = selectedTile.getI();
247                 int sJ = selectedTile.getJ();
248                 int oI = otherTile.getI();
249                 int oJ = otherTile.getJ();
250                 Soldier selectedSoldier = army1.get(generateKey(sI, sJ));
251                 int distance;
252                 switch (selectedAction) {
253                     case "MOVER":
254                         distance = selectedSoldier.getTypeFile().equals("knight") ? 2 : 1;
255                         if (selectedTile.isConnected(otherTile, distance) &&
256                             !army1.containsKey(otherKey)
257                             && !army2.containsKey(otherKey)) {
258                             moveSoldier(true, sI, sJ, oI, oJ);
259                             removeActionsMenu();
260
261                             out = new DataOutputStream(new FileOutputStream(connectionFile));
262                             out.writeInt(OPERATION_MOVE);
263                             Utils.writeString(out, matchCode);
264                             Utils.writeIdxs(out, sI, sJ, oI, oJ);
265                             out.close();
266
267                             playerTurn = false;
268                             selectedTile = null;
269                             out.close();
270                             return true;
271                         }
272                         showMessage("Movimiento no valido.");
273                         break;
274                     case "ATACAR":
275                         distance = selectedSoldier.getTypeFile().equals("archer") ? 2 : 1;
276                         if (selectedTile.isConnected(otherTile, distance) &&
277                             army2.containsKey(otherKey)) {
278                             attackSoldier(true, sI, sJ, oI, oJ);
279                             removeActionsMenu();
280
281                             out = new DataOutputStream(new FileOutputStream(connectionFile));
282                             out.writeInt(OPERATION_ATTACK);
283                             Utils.writeString(out, matchCode);
284                             Utils.writeIdxs(out, sI, sJ, oI, oJ);
285                             out.close();
286
287                             playerTurn = false;
```



```
286         selectedTile = null;
287         out.close();
288         return true;
289     }
290     showMessage("Ataque no valido.");
291     break;
292 }
293 } catch (Exception e) {
294     e.printStackTrace();
295 }
296 }
297 return false;
298 }
299
300 // Métodos que funcionan en ambos sentidos, host -> guest o guest -> host
301 private void moveSoldier(boolean isPlayer, int iSelected, int jSelected, int iOther, int
    jOther) {
302     Tile selectedTile = tiles[iSelected][jSelected];
303     Tile otherTile = tiles[iOther][jOther];
304     String selectedKey = selectedTile.getKey();
305     String otherKey = otherTile.getKey();
306
307     HashMap<String, Soldier> army = null;
308     BetterColor color = null;
309     if (isPlayer) {
310         army = army1;
311         color = PLAYER_COLOR_TRANS;
312     } else {
313         army = army2;
314         color = ENEMY_COLOR_TRANS;
315     }
316
317     Soldier soldier = army.remove(selectedKey);
318     army.put(otherKey, soldier);
319     selectedTile.setImageAndhealth("tile", 0);
320     setStyleColor(selectedTile, null);
321     otherTile.setImageAndhealth(soldier.getTypeFile(), soldier.getCurrentHealth());
322     setStyleColor(otherTile, color);
323
324     String message = soldier + " se mueve." + "\n";
325     if (isPlayer)
326         playerData.appendText(message);
327     else
328         enemyData.appendText(message);
329 }
330
331 private void attackSoldier(boolean isPlayer, int iSelected, int jSelected, int iOther,
    int jOther) {
332     Tile selectedTile = tiles[iSelected][jSelected];
333     Tile otherTile = tiles[iOther][jOther];
334     String selectedKey = selectedTile.getKey();
335     String otherKey = otherTile.getKey();
336
337     Soldier soldierAttacks = null;
338     Soldier soldierReceives = null;
339     if (isPlayer) {
```

```
340         soldierAttacks = army1.get(selectedKey);
341         soldierReceives = army2.get(otherKey);
342     } else {
343         soldierAttacks = army2.get(selectedKey);
344         soldierReceives = army1.get(otherKey);
345     }
346
347     int damage = soldierAttacks.attack(soldierReceives);
348     otherTile.setImageAndhealth(soldierReceives.getTypeFile(),
349         soldierReceives.getCurrentHealth());
350     String message = String.format("%s ataca a %s con %d de daño\n", soldierAttacks,
351         soldierReceives, damage);
352     if (isPlayer)
353         playerData.appendText(message);
354     else
355         enemyData.appendText(message);
356
357     if (soldierReceives.getCurrentHealth() <= 0) {
358         soldierAttacks.heal();
359         selectedTile.setImageAndhealth(soldierAttacks.getTypeFile(),
360             soldierAttacks.getCurrentHealth());
361         otherTile.setImageAndhealth("tile", 0);
362         setStyleColor(otherTile, null);
363
364         message = soldierReceives + " ha muerto!\n";
365         if (isPlayer) {
366             playerData.appendText(message);
367             army2.remove(otherKey);
368             if (army2.size() == 0) {
369                 if (idEnemy != 0)
370                     dbConnector.createMatch(idPlayer, idEnemy);
371                 endGame(pName, kingdomPlayer);
372             }
373         } else {
374             enemyData.appendText(message);
375             army1.remove(otherKey);
376             if (army1.size() == 0) {
377                 if (idEnemy != 0)
378                     dbConnector.createMatch(idEnemy, idPlayer);
379                 endGame(eName, kingdomEnemy);
380             }
381         }
382     }
383
384     private void endGame(String name, String kingdom) {
385         showMessage(String.format("%s ha ganado con el reino %s!", name, kingdom));
386         gameEnded = true;
387     }
388
389     private void printMessage(String message, BetterColor color) {
390         Text messageText = new Text(message);
391         messageText.setFont(Font.font("Book Antiqua"));
392         messageText.setFill(color.getColor());
393         messageText.setWrappingWidth(width - height);
```

```
393
394     ObservableList<Node> children = chatOutput.getChildren();
395     children.add(children.size() - 1, messageText);
396     chatOutputPane.setVvalue(1);
397 }
398
399 private void setConnection() {
400     path = "connections/" + idConnection + ".dat";
401     connectionFile = new File(path);
402     try {
403         connectionFile.createNewFile();
404         dataReceiver = new DataReceiver();
405         dataReceiver.start();
406     } catch (Exception e) {
407         e.printStackTrace();
408     }
409 }
410
411 private String generateKey(int i, int j) {
412     return i + "," + j;
413 }
414
415 private void showMessage(String message) {
416     messagePane.setVisible(true);
417     messageOutput.setText(message);
418 }
419
420 private class DataReceiver extends Thread {
421     private File matchFile = new File(path);
422     private long lastModified = matchFile.lastModified();
423     private boolean gameEnded;
424
425     public void run() {
426         try {
427             while (!gameEnded) {
428                 // Comprueba si el archivo de la partida ha sido modificado
429                 if (matchFile.lastModified() != lastModified) {
430                     DataInputStream in = new DataInputStream(new
431                         FileInputStream(matchFile));
432                     int response = in.readInt();
433                     switch (response) {
434                         // Mensaje de chat
435                         case RESPONSE_CHAT:
436                             String message = Utils.readString(in);
437                             Platform.runLater(() -> {
438                                 printMessage(message, ENEMY_COLOR);
439                             });
440                             break;
441
442                         // Movimientos y ataques
443                         case RESPONSE_MOVE:
444                         case RESPONSE_ATTACK:
445                             int sI = in.readInt();
446                             int sJ = in.readInt();
447                             int oI = in.readInt();
448                             int oJ = in.readInt();
```

```

448
449         Platform.runLater(() -> {
450             showActionsMenu();
451             playerTurn = true;
452             if (response == RESPONSE_MOVE)
453                 moveSoldier(false, sI, sJ, oI, oJ);
454             else
455                 attackSoldier(false, sI, sJ, oI, oJ);
456         });
457         break;
458     }
459     lastModified = matchFile.lastModified();
460     in.close();
461 }
462
463     sleep(500);
464 }
465 } catch (Exception e) {
466     e.printStackTrace();
467 }
468 }
469
470 public void endGame() {
471     gameEnded = true;
472 }
473 }
474 }

```

- Clase que controla el videojuego principal, se encarga de la manipulación de todos los elementos FX.
- Realiza la inicialización de todos los elementos del escenario.
- Posee métodos que permiten enviar al servidor los datos tanto de chats como de movimientos.
- Al concluir la partida envía los resultados a la base de datos para ser recuperados más tarde.
- Posee la clase interna DataReceiver que permite recibir las respuestas del servidor y ejecutarlas.

MainGame.fxml

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <?import javafx.geometry.Insets?>
4  <?import javafx.scene.control.Button?>
5  <?import javafx.scene.control.Label?>
6  <?import javafx.scene.control.ScrollPane?>
7  <?import javafx.scene.control.TextArea?>
8  <?import javafx.scene.control.TextField?>
9  <?import javafx.scene.control.TitledPane?>
10 <?import javafx.scene.image.Image?>
11 <?import javafx.scene.image.ImageView?>
12 <?import javafx.scene.layout.AnchorPane?>
13 <?import javafx.scene.layout.ColumnConstraints?>
14 <?import javafx.scene.layout.GridPane?>
15 <?import javafx.scene.layout.HBox?>

```

```
16 <?import javafx.scene.layout.Pane?>
17 <?import javafx.scene.layout.RowConstraints?>
18 <?import javafx.scene.layout.StackPane?>
19 <?import javafx.scene.layout.TilePane?>
20 <?import javafx.scene.layout.VBox?>
21 <?import javafx.scene.text.Font?>
22 <?import javafx.scene.text.Text?>
23
24 <StackPane xmlns="http://javafx.com/javafx/21" xmlns:fx="http://javafx.com/fxml/1"
    fx:controller="FX.MainGame.MainGameController">
25     <children>
26         <ImageView fx:id="boardBackground" fitHeight="150.0" fitWidth="200.0"
            pickOnBounds="true" preserveRatio="true" />
27         <HBox>
28             <children>
29                 <GridPane fx:id="uiBoard" maxHeight="-Infinity" maxWidth="-Infinity"
                    minHeight="-Infinity" minWidth="-Infinity">
30                     <columnConstraints>
31                         <ColumnConstraints hgrow="SOMETIMES" />
32                         <ColumnConstraints hgrow="SOMETIMES" />
33                         <ColumnConstraints hgrow="SOMETIMES" />
34                         <ColumnConstraints hgrow="SOMETIMES" />
35                         <ColumnConstraints hgrow="SOMETIMES" />
36                         <ColumnConstraints hgrow="SOMETIMES" />
37                         <ColumnConstraints hgrow="SOMETIMES" />
38                         <ColumnConstraints hgrow="SOMETIMES" />
39                     <ColumnConstraints hgrow="SOMETIMES" />
40                     <ColumnConstraints hgrow="SOMETIMES" />
41                 </columnConstraints>
42                 <rowConstraints>
43                     <RowConstraints vgrow="SOMETIMES" />
44                     <RowConstraints vgrow="SOMETIMES" />
45                     <RowConstraints vgrow="SOMETIMES" />
46                     <RowConstraints vgrow="SOMETIMES" />
47                     <RowConstraints vgrow="SOMETIMES" />
48                     <RowConstraints vgrow="SOMETIMES" />
49                     <RowConstraints vgrow="SOMETIMES" />
50                     <RowConstraints vgrow="SOMETIMES" />
51                     <RowConstraints vgrow="SOMETIMES" />
52                     <RowConstraints vgrow="SOMETIMES" />
53                 </rowConstraints>
54             </GridPane>
55             <StackPane>
56                 <children>
57                     <ImageView fx:id="dataBackground" fitHeight="150.0" fitWidth="200.0"
                        pickOnBounds="true" preserveRatio="true" />
58                     <VBox>
59                         <children>
60                             <HBox>
61                                 <children>
62                                     <TextArea fx:id="playerData" blendMode="MULTIPLY">
63                                         <font>
64                                             <Font name="Book Antiqua" size="10.0" />
65                                         </font></TextArea>
66                                     <TextArea fx:id="enemyData" blendMode="MULTIPLY">
67                                         <font>
```

```

68         <Font name="Book Antiqua" size="10.0" />
69     </font></TextArea>
70 </children>
71 </HBox>
72 <VBox alignment="CENTER">
73     <children>
74         <ScrollPane fx:id="chatOutputPane" blendMode="MULTIPLY"
75             hbarPolicy="NEVER" vbarPolicy="ALWAYS">
76             <content>
77                 <VBox fx:id="chatOutput" style="-fx-background-color:
78                     white;">
79                     <children>
80                         <Text strokeType="OUTSIDE" strokeWidth="0.0">
81                             <font>
82                                 <Font size="18.0" />
83                             </font>
84                         </Text>
85                     </children>
86                 </VBox>
87             </content>
88         </ScrollPane>
89         <TextField fx:id="chatInput" blendMode="MULTIPLY"
90             onAction="#sendMessage" promptText="Envia un mensaje!">
91             <VBox.margin>
92                 <Insets bottom="15.0" />
93             </VBox.margin>
94             <font>
95                 <Font name="Book Antiqua" size="12.0" />
96             </font></TextField>
97         <TilePane fx:id="actionsPane" alignment="CENTER" hgap="30.0">
98             <children>
99                 <VBox fx:id="moveActionPane" onMouseClicked="#setActionMove"
100                     TilePane.alignment="CENTER">
101                     <children>
102                         <Label alignment="CENTER" prefWidth="50.0" text="Mover">
103                             <font>
104                                 <Font name="Book Antiqua" size="12.0" />
105                             </font></Label>
106                         <ImageView fitHeight="50.0" fitWidth="50.0"
107                             pickOnBounds="true" preserveRatio="true">
108                             <image>
109                                 <Image url="@../img/action_move.png" />
110                             </image>
111                         </ImageView>
112                     </children>
113                 </VBox>
114                 <VBox fx:id="attackActionPane"
115                     onMouseClicked="#setActionAttack"
116                     TilePane.alignment="CENTER">
117                     <children>
118                         <Label alignment="CENTER"
119                             maxWidth="1.7976931348623157E308" prefWidth="50.0"
120                             text="Atacar">
121                             <font>
122                                 <Font name="Book Antiqua" size="12.0" />
123                             </font></Label>

```

```

115         <ImageView fitHeight="50.0" fitWidth="50.0"
116             pickOnBounds="true" preserveRatio="true">
117             <image>
118                 <Image url="@../img/action_attack.png" />
119             </image>
120         </ImageView>
121     </children>
122 </VBox>
123 </children>
124 <padding>
125     <Insets bottom="15.0" />
126 </padding>
127 </TilePane>
128 <TextField fx:id="fileNameInput" blendMode="MULTIPLY"
129     promptText="Nombre del archivo">
130     <font>
131         <Font name="Book Antiqua" size="12.0" />
132     </font>
133 </TextField>
134 <Button mnemonicParsing="false" onAction="#saveMatch"
135     text="Guardar partida">
136     <font>
137         <Font name="Book Antiqua" size="12.0" />
138     </font>
139 </Button>
140 </children>
141 </VBox>
142 </children>
143 </VBox>
144 </children>
145 </StackPane>
146 </children>
147 </HBox>
148 <Pane fx:id="messagePane" visible="false">
149     <children>
150         <TitledPane animated="false" collapsible="false" layoutX="274.0" layoutY="74.0"
151             prefHeight="139.0" prefWidth="279.0" text="Mensaje">
152             <content>
153                 <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="180.0"
154                     prefWidth="200.0">
155                     <children>
156                         <TextArea fx:id="messageOutput" layoutX="-1.0" prefHeight="81.0"
157                             prefWidth="279.0" wrapText="true">
158                             <font>
159                                 <Font name="Book Antiqua" size="14.0" />
160                             </font>
161                         </TextArea>
162                         <Button layoutX="122.0" layoutY="84.0" mnemonicParsing="false"
163                             onAction="#closeMessage" text="OK">
164                             <font>
165                                 <Font name="Book Antiqua" size="12.0" />
166                             </font>
167                         </Button>
168                     </children>
169                 </AnchorPane>
170             </content>

```

```
164         <font>
165             <Font name="Book Antiqua" size="12.0" />
166         </font>
167     </TitledPane>
168 </children>
169 </Pane>
170 </children>
171 </StackPane>
```

- Clase FXML de JavaFX que posee el juego principal.
- Posee las maquetas que luego son rellenadas con el controlador para el juego principal.

MainMenuController.java

```
1 package FX.MainMenu;
2
3 import Utils.*;
4 import java.io.*;
5 import FX.MainGame.Board;
6 import FX.MainGame.MainGameController;
7 import javafx.application.*;
8 import javafx.collections.*;
9 import javafx.fxml.*;
10 import javafx.scene.*;
11 import javafx.scene.control.*;
12 import javafx.scene.layout.Pane;
13 import javafx.stage.Stage;
14
15 public class MainMenuController implements MainMenuOperation {
16     private final ObservableList<Resolution> RESOLUTIONS =
17         FXCollections.observableArrayList();
18     private final ObservableList<String> KINGDOMS = FXCollections.observableArrayList();
19     private final int CODE_LENGTH = 6;
20
21     private String pName;
22     private String eName;
23     private String pKingdom;
24     private String eKingdom;
25     private Resolution resolution;
26     private int idConnection;
27     private int idPlayer;
28     private int idEnemy;
29     private String path;
30     private File connectionFile;
31     private DataReceiver dataReceiver;
32     private Stage stage;
33     private DBConnector dbConnector;
34     private Board board;
35     private String matchCode;
36
37     @FXML
38     private TextField nameInput;
39     @FXML
40     private TextField passwordInput;
```



```
41 private Pane settingsPane;
42 @FXML
43 private ComboBox<Resolution> resolutionInput;
44 @FXML
45 private ComboBox<String> kingdomInput;
46 @FXML
47 private TextField createMatchCode;
48 @FXML
49 private TextField joinMatchCode;
50 @FXML
51 private Label playerName;
52 @FXML
53 private Label enemyName;
54 @FXML
55 private Label playerKingdom;
56 @FXML
57 private Label enemyKingdom;
58 @FXML
59 private Button startButton;
60 @FXML
61 private TitledPane messagePane;
62 @FXML
63 private TextArea messageOutput;
64 @FXML
65 private TextField fileNameInput;
66
67 public void setStage(Stage stage) {
68     this.stage = stage;
69 }
70
71 public void initialize() throws IOException {
72     try {
73         RESOLUTIONS.addAll(new Resolution(850, 480), new Resolution(1280, 720), new
74             Resolution(1366, 768),
75             new Resolution(1920, 1080));
76         resolutionInput.setItems(RESOLUTIONS);
77         resolutionInput.setValue(RESOLUTIONS.get(0));
78         resolution = resolutionInput.getValue();
79
80         KINGDOMS.addAll("INGLATERRA", "FRANCIA", "CASTILLA-ARAGÓN", "MOROS", "SACRO
81             IMPERIO");
82         kingdomInput.setItems(KINGDOMS);
83
84         dbConnector = new DBConnector();
85         setConnection();
86     } catch (Exception e) {
87         FileWriter writer = new FileWriter("error.log");
88         writer.write(e.getMessage());
89         writer.close();
90     }
91
92 public void setKingdom() {
93     pKingdom = kingdomInput.getValue();
94     playerKingdom.setText(pKingdom);
95 }
```

```
95
96 public void toggleSettings() {
97     settingsPane.setVisible(!settingsPane.isVisible());
98 }
99
100 public void setResolution() {
101     resolution = resolutionInput.getValue();
102 }
103
104 public void createMatch() {
105     if (checkName() && checkKingdom()) {
106         matchCode = "";
107         for (int i = 0; i < CODE_LENGTH; i++)
108             matchCode += (char) ('A' + (int) (Math.random() * 26));
109
110         createMatchCode.setText(matchCode);
111         try {
112             DataOutputStream out = new DataOutputStream(new
113                 FileOutputStream(connectionFile));
114             out.writeInt(OPERATION_CREATE);
115             Utils.writeStrings(out, new String[] { matchCode, pName, pKingdom });
116             out.close();
117         } catch (Exception e) {
118             e.printStackTrace();
119         }
120     }
121
122 public void joinMatch() {
123     matchCode = joinMatchCode.getText();
124     if (checkName() && checkKingdom() && matchCode.length() == CODE_LENGTH) {
125         try {
126             DataOutputStream out = new DataOutputStream(new
127                 FileOutputStream(connectionFile));
128             out.writeInt(OPERATION_JOIN);
129             Utils.writeStrings(out, new String[] { matchCode, pName, pKingdom });
130             out.writeInt(idPlayer);
131         } catch (Exception e) {
132             e.printStackTrace();
133         }
134     }
135
136 public void loadMatch() {
137     try {
138         String fileName = fileNameInput.getText();
139         ObjectInputStream out = new ObjectInputStream(new
140             FileInputStream(String.format("data/%s.sav", fileName)));
141         board = (Board) out.readObject();
142         out.close();
143
144         pKingdom = board.getKingdomPlayer();
145         eKingdom = board.getKingdomEnemy();
146         playerKingdom.setText(pKingdom);
147         enemyKingdom.setText(eKingdom);
```

```
148         showMessage("Partida cargada correctamente");
149     } catch (Exception e) {
150         e.printStackTrace();
151     }
152 }
153
154 public void startMatch() {
155     if (checkName() && checkEnemy() && checkKingdom()) {
156         try {
157             DataOutputStream out = new DataOutputStream(new
158                 FileOutputStream(connectionFile));
159             out.writeInt(OPERATION_START);
160             Utils.writeString(out, matchCode);
161             out.close();
162
163             if (board == null)
164                 board = new Board(pKingdom, eKingdom);
165
166             ObjectOutputStream outObj = new ObjectOutputStream(
167                 new FileOutputStream("connections/" + idConnection + ".obj"));
168             outObj.writeObject(board);
169             outObj.close();
170         } catch (Exception e) {
171             e.printStackTrace();
172         }
173
174         createGameStage();
175     }
176 }
177
178 public void login() {
179     String name = nameInput.getText();
180     idPlayer = dbConnector.loginPlayer(name, passwordInput.getText());
181     if (idPlayer == -1) {
182         showMessage("Usuario no encontrado.");
183     } else {
184         showMessage("Acceso correcto.");
185         pName = name;
186         playerName.setText(pName);
187         nameInput.setText("");
188         passwordInput.setText("");
189     }
190 }
191
192 public void register() {
193     pName = nameInput.getText();
194     dbConnector.registerPlayer(pName, passwordInput.getText());
195     showMessage("Usuario creado correctamente.");
196     login();
197 }
198
199 public void getStatistics() {
200     if (checkName()) {
201         int[] status = dbConnector.getWinsLoses(idPlayer);
202         showMessage(String.format("W: %d | L: %d", status[0], status[1]));
203     }
```

```
203     }
204
205     public void closeMessage() {
206         messagePane.setVisible(false);
207     }
208
209     public void restartMenu() {
210         createMatchCode.setText("");
211         joinMatchCode.setText("");
212         enemyName.setText("");
213         enemyKingdom.setText("");
214         startButton.setDisable(false);
215         dataReceiver = new DataReceiver();
216         dataReceiver.start();
217     }
218
219     private void setConnection() {
220         if (connectionFile == null) {
221             path = "connections/" + idConnection + ".dat";
222             connectionFile = new File(path);
223             while (connectionFile.exists()) {
224                 idConnection++;
225                 path = "connections/" + idConnection + ".dat";
226                 connectionFile = new File(path);
227             }
228         }
229         try {
230             connectionFile.createNewFile();
231             dataReceiver = new DataReceiver();
232             dataReceiver.start();
233         } catch (Exception e) {
234             e.printStackTrace();
235         }
236     }
237
238     private boolean checkName() {
239         boolean nameSet = pName != null;
240         if (!nameSet)
241             showMessage("Crea o accede a tu cuenta!");
242         return nameSet;
243     }
244
245     private boolean checkEnemy() {
246         boolean enemySet = eName != null;
247         if (!enemySet)
248             showMessage("Crea o únete a una partida!");
249         return enemySet;
250     }
251
252     private boolean checkKingdom() {
253         boolean kingdomSet = pKingdom != null;
254         if (!kingdomSet)
255             showMessage("Escoge un reino!");
256         return kingdomSet;
257     }
258
```

```
259 private void createGameStage() {
260     dataReceiver.startGame();
261     stage.hide();
262     new MainGame(this);
263 }
264
265 private void showMessage(String message) {
266     messagePane.setVisible(true);
267     messageOutput.setText(message);
268 }
269
270 // Clase interna para el receptor de datos en un hilo separado
271 private class DataReceiver extends Thread {
272     private File matchFile = new File(path);
273     private long lastModified = matchFile.lastModified();
274     private boolean gameStarted;
275
276     public void run() {
277         try {
278             while (!gameStarted) {
279                 // Comprueba si el archivo de la partida ha sido modificado
280                 if (matchFile.lastModified() != lastModified) {
281                     DataInputStream in = new DataInputStream(new
282                         FileInputStream(matchFile));
283                     int response = in.readInt();
284                     String name, kingdom;
285                     switch (response) {
286                         // Respuesta del anfitrión
287                         case RESPONSE_HOST:
288                             name = Utils.readString(in);
289                             kingdom = Utils.readString(in);
290                             int idOther = in.readInt();
291                             // Actualiza el nombre del oponente en la interfaz de usuario
292                             Platform.runLater(() -> {
293                                 setEnemy(name, kingdom);
294                                 idEnemy = idOther;
295                             });
296                             break;
297                         // Respuesta del invitado
298                         case RESPONSE_GUEST:
299                             name = Utils.readString(in);
300                             kingdom = Utils.readString(in);
301                             if (name.equals("")) {
302                                 showMessage("La partida no existe.");
303                             } else {
304                                 // Actualiza el nombre del oponente en la interfaz de
305                                 // usuario y desactiva el
306                                 // botón de inicio
307                                 Platform.runLater(() -> {
308                                     setEnemy(name, kingdom);
309                                     startButton.setDisable(true);
310                                 });
311                             }
312                             break;
313                         // Respuesta de inicio de la partida
314                         case RESPONSE_START:
```

```
313         File objFile = new File("connections/" + idConnection + ".obj");
314         ObjectInputStream inObj = new ObjectInputStream(new
            FileInputStream(objFile));
315         board = (Board) inObj.readObject();
316         inObj.close();
317         objFile.delete();
318         // Inicia el juego principal
319         Platform.runLater(() -> {
320             createGameStage();
321         });
322     }
323     lastModified = matchFile.lastModified();
324     in.close();
325 }
326
327     sleep(1000);
328 }
329 } catch (Exception e) {
330     e.printStackTrace();
331 }
332 }
333
334 public void startGame() {
335     gameStarted = true;
336 }
337
338 private void setEnemy(String name, String kingdom) {
339     eName = name;
340     eKingdom = kingdom;
341     enemyName.setText(eName);
342     enemyKingdom.setText(eKingdom);
343 }
344 }
345
346 private class MainGame {
347     public MainGame(MainMenuController mainMenuController) {
348         try {
349             // Carga el archivo FXML del juego principal y configura la escena
350             FXMLLoader loader = new
                FXMLLoader(getClass().getResource("/FX/MainGame/MainGame.fxml"));
351             Parent root = loader.load();
352
353             Stage mainGame = new Stage();
354             mainGame.setTitle("Main Game");
355             mainGame.setScene(new Scene(root, resolution.getWidth(),
                resolution.getHeight()));
356             mainGame.setResizable(false);
357             mainGame.show();
358
359             MainGameController controller = loader.getController();
360             controller.init(mainMenuController, resolution, stage, mainGame, board,
                idConnection, matchCode, pName,
361                 eName,
362                 idPlayer, idEnemy);
363         } catch (Exception e) {
364             e.printStackTrace();
365         }
366     }
367 }
```

```
365     }
366   }
367 }
368 }
```

- Clase que controla el menú principal, se encarga de la manipulación de todos los elementos FX.
- Posee métodos que permiten enviar al servidor los datos para crear partidas, unirse a partidas y iniciar la partida.
- Es el encargado de realizar la conexión con la base de datos usando la clase DBConnector y con el servidor usando un archivo.
- Permite crear usuarios y obtener las victorias y derrotas asociadas a ese usuario.
- Posee la clase interna DataReceiver que permite recibir las respuestas del servidor y ejecutarlas.

MainMenu.fxml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <?import javafx.scene.control.Button?>
4 <?import javafx.scene.control.ComboBox?>
5 <?import javafx.scene.control.Label?>
6 <?import javafx.scene.control.PasswordField?>
7 <?import javafx.scene.control.TextArea?>
8 <?import javafx.scene.control.TextField?>
9 <?import javafx.scene.control.TitledPane?>
10 <?import javafx.scene.image.Image?>
11 <?import javafx.scene.image.ImageView?>
12 <?import javafx.scene.layout.AnchorPane?>
13 <?import javafx.scene.layout.Pane?>
14 <?import javafx.scene.text.Font?>
15
16 <AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
  minWidth="-Infinity" prefHeight="600.0" prefWidth="325.0"
  xmlns="http://javafx.com/javafx/21" xmlns:fx="http://javafx.com/fxml/1"
  fx:controller="FX.MainMenu.MainMenuController">
17   <children>
18     <Label layoutX="99.0" layoutY="9.0" text="VIDEOJUEGO">
19       <font>
20         <Font name="Book Antiqua" size="20.0" />
21       </font>
22     </Label>
23     <Button layoutX="26.0" layoutY="307.0" mnemonicParsing="false" onAction="#joinMatch"
      text="Unirse a partida">
24       <font>
25         <Font name="Book Antiqua" size="12.0" />
26       </font></Button>
27     <Button layoutX="26.0" layoutY="276.0" mnemonicParsing="false" onAction="#createMatch"
      text="Crear partida">
28       <font>
29         <Font name="Book Antiqua" size="12.0" />
30       </font>
31     </Button>
```

```

32 <TextField fx:id="createMatchCode" editable="false" layoutX="149.0" layoutY="276.0"
    prefHeight="25.0" prefWidth="150.0" promptText="Código">
33 <font>
34 <Font name="Book Antiqua" size="12.0" />
35 </font>
36 </TextField>
37 <Button layoutX="26.0" layoutY="244.0" mnemonicParsing="false" onAction="#loadMatch"
    text="Cargar partida">
38 <font>
39 <Font name="Book Antiqua" size="12.0" />
40 </font></Button>
41 <TextField fx:id="joinMatchCode" layoutX="149.0" layoutY="307.0" prefHeight="25.0"
    prefWidth="150.0" promptText="Código">
42 <font>
43 <Font name="Book Antiqua" size="12.0" />
44 </font></TextField>
45 <TextField fx:id="fileNameInput" layoutX="149.0" layoutY="244.0" prefHeight="25.0"
    prefWidth="150.0" promptText="Nombre del archivo">
46 <font>
47 <Font name="Book Antiqua" size="12.0" />
48 </font></TextField>
49 <Label layoutX="113.0" layoutY="223.0" text="SALA DE ESPERA">
50 <font>
51 <Font name="Book Antiqua" size="14.0" />
52 </font></Label>
53 <ImageView fitHeight="211.0" fitWidth="317.0" layoutX="61.0" layoutY="343.0"
    pickOnBounds="true" preserveRatio="true">
54 <image>
55 <Image url="@../img/waiting.png" />
56 </image>
57 </ImageView>
58 <Button fx:id="startButton" layoutX="141.0" layoutY="564.0" mnemonicParsing="false"
    onAction="#startMatch" text="Iniciar">
59 <font>
60 <Font name="Book Antiqua" size="12.0" />
61 </font></Button>
62 <TextField fx:id="nameInput" layoutX="60.0" layoutY="66.0" prefHeight="25.0"
    prefWidth="212.0" promptText="Nombre">
63 <font>
64 <Font name="Book Antiqua" size="12.0" />
65 </font></TextField>
66 <Label fx:id="playerName" layoutX="74.0" layoutY="357.0" prefHeight="17.0"
    prefWidth="150.0">
67 <font>
68 <Font name="Book Antiqua" size="16.0" />
69 </font></Label>
70 <Label layoutX="146.0" layoutY="426.0" text="VS">
71 <font>
72 <Font name="Gill Sans MT" size="40.0" />
73 </font>
74 </Label>
75 <Label fx:id="enemyName" alignment="CENTER_RIGHT" layoutX="110.0" layoutY="519.0"
    prefHeight="17.0" prefWidth="150.0">
76 <font>
77 <Font name="Book Antiqua" size="16.0" />
78 </font></Label>

```



```

79     <Label layoutX="131.0" layoutY="39.0" text="USUARIO">
80         <font>
81             <Font name="Book Antiqua" size="14.0" />
82         </font>
83     </Label>
84     <PasswordField fx:id="passwordInput" layoutX="61.0" layoutY="100.0" prefHeight="25.0"
85         prefWidth="212.0" promptText="Contraseña">
86         <font>
87             <Font name="Book Antiqua" size="12.0" />
88         </font></PasswordField>
89     <Button layoutX="10.0" layoutY="135.0" mnemonicParsing="false" onAction="#register"
90         text="Crear cuenta">
91         <font>
92             <Font name="Book Antiqua" size="12.0" />
93         </font></Button>
94     <Button layoutX="102.0" layoutY="135.0" mnemonicParsing="false" onAction="#login"
95         text="Ingresar a cuenta">
96         <font>
97             <Font name="Book Antiqua" size="12.0" />
98         </font></Button>
99     <Button layoutX="218.0" layoutY="135.0" mnemonicParsing="false"
100         onAction="#getStatistics" text="Ver estadísticas">
101         <font>
102             <Font name="Book Antiqua" size="12.0" />
103         </font></Button>
104     <Label fx:id="playerKingdom" layoutX="74.0" layoutY="377.0" prefHeight="17.0"
105         prefWidth="150.0">
106         <font>
107             <Font name="Book Antiqua" size="12.0" />
108         </font></Label>
109     <Label fx:id="enemyKingdom" alignment="CENTER_RIGHT" layoutX="110.0" layoutY="502.0"
110         prefHeight="17.0" prefWidth="150.0">
111         <font>
112             <Font name="Book Antiqua" size="12.0" />
113         </font></Label>
114     <ComboBox fx:id="kingdomInput" layoutX="90.0" layoutY="189.0" onAction="#setKingdom"
115         prefWidth="150.0" promptText="Reino" />
116     <Label layoutX="143.0" layoutY="165.0" text="REINO">
117         <font>
118             <Font name="Book Antiqua" size="14.0" />
119         </font>
120     </Label>
121     <Pane fx:id="settingsPane" prefHeight="116.0" prefWidth="325.0"
122         style="-fx-background-color: white;" visible="false">
123     <children>
124         <Label layoutX="93.0" layoutY="14.0" text="CONFIGURACIONES">
125             <font>
126                 <Font name="Book Antiqua" size="14.0" />
127             </font>
128         </Label>
129         <Label layoutX="133.0" layoutY="41.0" text="Resolución">
130             <font>
131                 <Font name="Book Antiqua" size="12.0" />
132             </font></Label>
133         <ComboBox fx:id="resolutionInput" layoutX="88.0" layoutY="63.0"
134             onAction="#setResolution" prefWidth="150.0" promptText="Resolución" />

```

```

126     </children>
127 </Pane>
128 <Pane layoutX="295.0" layoutY="7.0" onMouseClicked="#toggleSettings" prefHeight="25.0"
    prefWidth="25.0">
129     <children>
130         <ImageView fitHeight="25.0" fitWidth="25.0" layoutX="-1.0" pickOnBounds="true"
            preserveRatio="true">
131             <image>
132                 <Image url="@../img/settings.png" />
133             </image>
134         </ImageView>
135     </children>
136 </Pane>
137 <TitledPane fx:id="messagePane" animated="false" collapsible="false" layoutX="25.0"
    layoutY="231.0" prefHeight="139.0" prefWidth="279.0" text="Mensaje" visible="false">
138     <content>
139         <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="180.0" prefWidth="200.0">
140             <children>
141                 <TextArea fx:id="messageOutput" layoutX="-1.0" prefHeight="81.0"
                    prefWidth="279.0" wrapText="true">
142                     <font>
143                         <Font name="Book Antiqua" size="14.0" />
144                     </font></TextArea>
145                 <Button layoutX="122.0" layoutY="84.0" mnemonicParsing="false"
                    onAction="#closeMessage" text="OK">
146                     <font>
147                         <Font name="Book Antiqua" size="12.0" />
148                     </font></Button>
149             </children>
150         </AnchorPane>
151     </content>
152     <font>
153         <Font name="Book Antiqua" size="12.0" />
154     </font>
155 </TitledPane>
156 </children>
157 </AnchorPane>

```

- Clase FXML de JavaFX que posee el juego principal.
- Posee todo el menú principal ya ordenado, puesto que el menú principal no es reescalable.

BetterColor.java

```

1 package Utils;
2
3 import java.io.Serializable;
4 import javafx.scene.paint.Color;
5
6 public class BetterColor implements Serializable {
7     private double redF;
8     private double greenF;
9     private double blueF;
10    private double alphaF;
11    private int redD;
12    private int greenD;

```

```
13 private int blueD;
14 private int alphaD;
15
16 public BetterColor(double red, double green, double blue, double alpha) {
17     this.redF = red;
18     this.greenF = green;
19     this.blueF = blue;
20     this.alphaF = alpha;
21
22     this.redD = (int) (redF * 255);
23     this.greenD = (int) (greenF * 255);
24     this.blueD = (int) (blueF * 255);
25     this.alphaD = (int) (alphaF * 255);
26 }
27
28 public Color getColor() {
29     return new Color(redF, greenF, blueF, alphaF);
30 }
31
32 public String getRGBA() {
33     return String.format("rgba(%d, %d, %d, %d)", redD, blueD, greenD, alphaD);
34 }
35 }
```

- Clase de apoyo que contiene un color serializable.
- Permite generar su representación en color FX y como rgba para los estilos.

MainGameOperation.java

```
1 package Utils;
2
3 public interface MainGameOperation {
4     int OPERATION_CHAT = 300;
5     int OPERATION_MOVE = 301;
6     int OPERATION_ATTACK = 302;
7     int RESPONSE_CHAT = 400;
8     int RESPONSE_MOVE = 401;
9     int RESPONSE_ATTACK = 402;
10 }
```

- Interfaz que mantiene los códigos de operación y respuesta para el servidor (Del menú principal).

MainMenuOperation.java

```
1 package Utils;
2
3 public interface MainMenuOperation {
4     int OPERATION_CREATE = 100;
5     int OPERATION_JOIN = 101;
6     int OPERATION_START = 102;
7     int RESPONSE_HOST = 200;
8     int RESPONSE_GUEST = 201;
9     int RESPONSE_START = 202;
10 }
```

- Interfaz que mantiene los códigos de operación y respuesta para el servidor (Del juego principal).

Resolution.java

```
1 package Utils;
2
3 public class Resolution {
4     private int width;
5     private int height;
6
7     public Resolution(int width, int height) {
8         this.width = width;
9         this.height = height;
10    }
11
12    public int getWidth() {
13        return width;
14    }
15
16    public int getHeight() {
17        return height;
18    }
19
20    public String toString() {
21        return width + " x " + height;
22    }
23 }
```

- Clase de apoyo que contiene una resolución (ancho x alto).

Tile.java

```
1 package Utils;
2
3 import javafx.scene.control.Label;
4 import javafx.scene.image.*;
5 import javafx.scene.layout.Pane;
6
7 import java.util.HashMap;
8
9 public class Tile extends Pane implements VideogameConstants {
10     private HashMap<String, ImageView> images = new HashMap<String, ImageView>();
11     private Label health;
12     private ImageView image;
13     private String type;
14     private int i;
15     private int j;
16
17     public Tile(int health, String type, double size, int i, int j) {
18         this.i = i;
19         this.j = j;
20         this.type = type;
21         for (int n = 0; n < TYPE_FILES.length; n++)
22             images.put(TYPE_FILES[n], generateImageView(size, TYPE_FILES[n]));
23         this.health = generateHealthLabel(size);
24     }
25 }
```

```
24     setImageAndhealth(type, health);
25 }
26
27 public void setImageAndhealth(String type, int hp) {
28     while (getChildren().size() > 0)
29         getChildren().remove(0);
30
31     image = images.get(type);
32     getChildren().add(image);
33
34     if (type.equals("tile"))
35         health.setText("");
36     else
37         health.setText(hp + "");
38     getChildren().add(health);
39
40 }
41
42 public int getI() {
43     return i;
44 }
45
46 public int getJ() {
47     return j;
48 }
49
50 public String getType() {
51     return type;
52 }
53
54 public boolean isConnected(Tile other, int distance) {
55     return Math.abs(other.getI() - i) <= distance && Math.abs(other.getJ() - j) <=
56         distance;
57 }
58
59 private ImageView generateImageView(double size, String type) {
60     Image image = new Image(String.format("img/tile_%s.png", type));
61     ImageView imageView = new ImageView(image);
62     imageView.setFitWidth(size);
63     imageView.setFitHeight(size);
64     images.put(type, imageView);
65     return imageView;
66 }
67
68 private Label generateHealthLabel(double size) {
69     Label label = new Label();
70     label.setLayoutX(size / 18);
71     label.setLayoutY(size / 18);
72     label.setTextFill(BACKGROUND_COLOR.getColor());
73     return label;
74 }
75
76 public String getKey() {
77     return i + "," + j;
78 }
```

```
79 public String toString() {  
80     return "a " + type + "!: " + i + ", " + j;  
81 }  
82 }
```

- Clase que mantiene una celda del tablero.
- Permite cambiar la imagen y la vida de cada celda.
- Posee otros métodos de utilidad como retornar posición en el tablero y hallar distancia entre casillas.

Utils.java

```
1 package Utils;  
2  
3 import java.io.*;  
4  
5 public class Utils {  
6     public static String readString(DataInputStream in) throws IOException {  
7         char c;  
8         String str = "";  
9         while ((c = in.readChar()) != 0)  
10             str += c;  
11         return str;  
12     }  
13  
14     public static void writeStrings(DataOutputStream out, String[] strings) throws  
15         IOException {  
16         for (String str : strings) {  
17             out.writeChars(str);  
18             out.writeChar(0);  
19         }  
20  
21     public static void writeString(DataOutputStream out, String str) throws IOException {  
22         out.writeChars(str);  
23         out.writeChar(0);  
24     }  
25  
26     public static void writeIdxs(DataOutputStream out, int sI, int sJ, int oI, int oJ) throws  
27         IOException {  
28         out.writeInt(sI);  
29         out.writeInt(sJ);  
30         out.writeInt(oI);  
31         out.writeInt(oJ);  
32     }  
33 }
```

- Clase de utilidad que posee diferentes atajos de lectura y escritura de archivos.

VideogameConstants.java

```
1 package Utils;
```

```
2
3 public interface VideogameConstants {
4     String[] TERRAINS = { "BOSQUE", "CAMPO ABIERTO", "MONTAA", "DESIERTO", "PLAYA" };
5     String[] TERRAIN_FILES = { "forest", "meadow", "mountain", "desert", "beach" };
6     String[] TYPES = { "CABALLERO", "ARQUERO", "ESPADACHIN", "LANCERO" };
7     String[] TYPE_FILES = { "knight", "archer", "swordsman", "spearman", "tile" };
8
9     BetterColor PLAYER_COLOR = new BetterColor(0.27, 0.51, 1, 1); // #4580ff
10    BetterColor ENEMY_COLOR = new BetterColor(1, 0.27, 0.27, 1); // #ff4545
11    BetterColor PLAYER_COLOR_TRANS = new BetterColor(0.27, 0.51, 1, 0.1); // #4580ff, op 10%
12    BetterColor ENEMY_COLOR_TRANS = new BetterColor(1, 0.27, 0.27, 0.1); // #ff4545, op 10%
13    BetterColor BACKGROUND_COLOR = new BetterColor(0.1, 0.1, 0.1, 1); // #1a1a1a
14    BetterColor SELECTED_COLOR = new BetterColor(0.8, 0.8, 0.8, 0.05); // #cccccc, op 5%
15    int TOTAL_SOLDIERS = 5;
16    int SIZE = 10;
17 }
```

- Interfaz que mantiene valores predeterminados por el juego.

Videogame.java

```
1
2 import java.io.*;
3 import FX.MainMenu.MainMenuController;
4 import javafx.application.Application;
5 import javafx.fxml.FXMLLoader;
6 import javafx.scene.*;
7 import javafx.stage.*;
8
9 public class Videogame extends Application {
10
11     @Override
12     public void start(Stage primaryStage) throws IOException {
13         try {
14             // Carga el archivo FXML del menú principal y configura la escena
15             FXMLLoader loader = new
16                 FXMLLoader(getClass().getResource("FX/MainMenu/MainMenu.fxml"));
17             Parent root = loader.load();
18
19             MainMenuController controller = loader.getController();
20             controller.setStage(primaryStage);
21
22             primaryStage.setTitle("Main Menu");
23             primaryStage.setScene(new Scene(root, 325, 600));
24             primaryStage.setResizable(false);
25             primaryStage.show();
26         } catch (Exception e) {
27             FileWriter writer = new FileWriter("error.log");
28             writer.write(e.getMessage());
29             writer.close();
30         }
31     }
32
33     public static void main(String[] args) {
34         launch(args);
35     }
36 }
```

```
34 }  
35 }
```

- Clase principal, que instancia la ventana e inicia el juego.

Main.java

```
1 public class Main {  
2     public static void main(String[] args) {  
3         Videogame.main(args);  
4     }  
5 }
```

- Clase que llama al main de la clase principal (necesario para la exportación a jar).

6. Ejecución del código

6.1. Video de ejecución

<https://drive.google.com/file/d/1EVE3Fd08z9WhAqnuwiKGPKAzVU614Aol/view?usp=sharing>

7. Diagrama UML

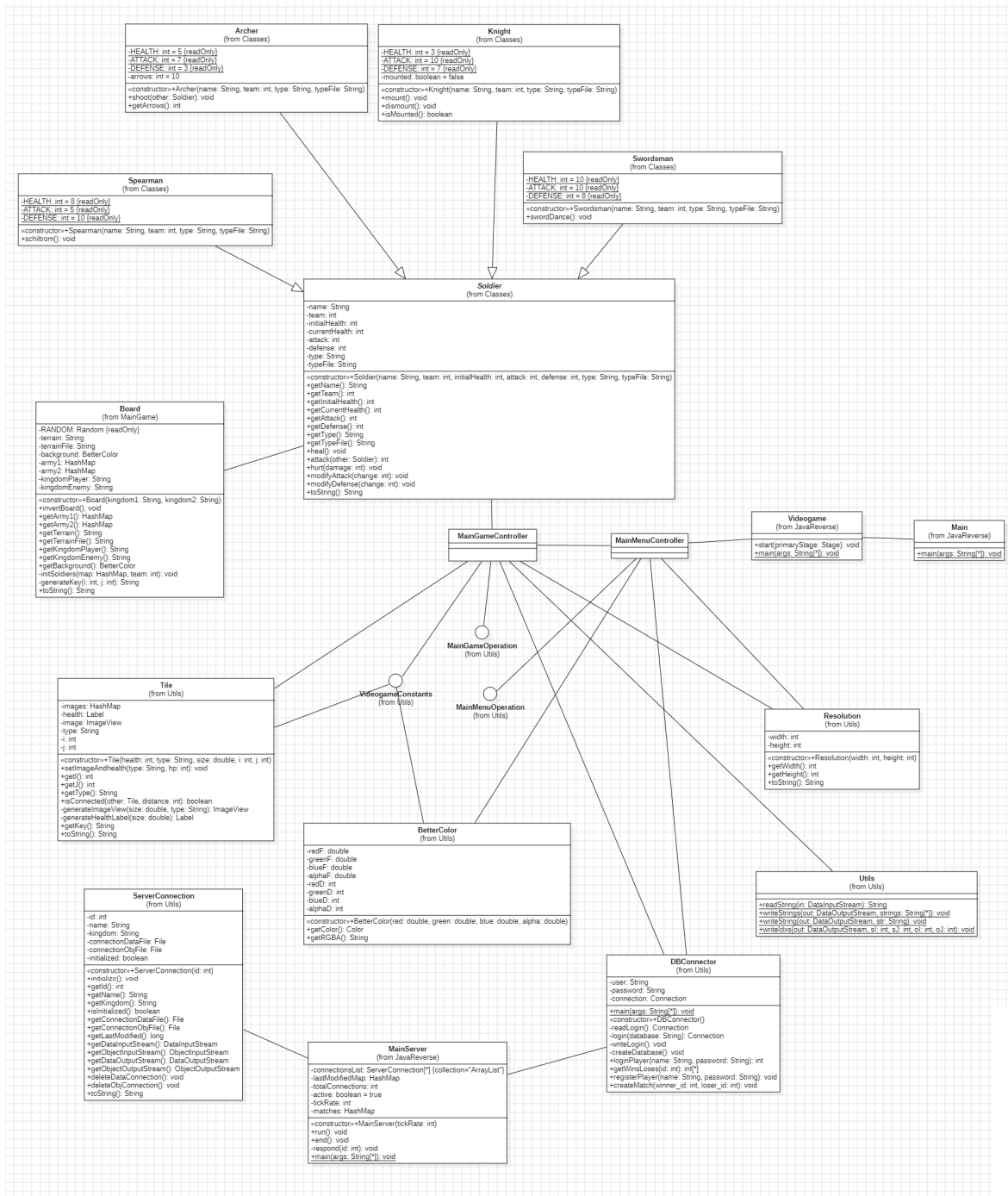


Diagrama UML.

8. Estructura de laboratorio Proyecto Final

- El contenido que se entrega en este laboratorio es el siguiente:

```
proyecto_final/  
|--- EJECUTABLES  
|   |--- SERVER.jar  
|   |--- VIDEOGAME.jar  
|--- INFORME  
|   |--- img  
|       |--- commit_01.png  
|       |--- commit_02.png  
|       |--- commit_03.png  
|       |--- commit_04.png  
|       |--- commit_05.png  
|       |--- commit_06.png  
|       |--- commit_07.png  
|       |--- commit_08.png  
|       |--- commit_09.png  
|       |--- commit_10.png  
|       |--- commit_11.png  
|       |--- commit_12.png  
|       |--- commit_13.png  
|       |--- commit_14.png  
|       |--- commit_15.png  
|       |--- commit_16.png  
|       |--- commit_17.png  
|       |--- commit_18.png  
|       |--- commit_19.png  
|       |--- commit_20.png  
|       |--- commit_21.png  
|       |--- commit_22.png  
|       |--- commit_23.png  
|       |--- commit_24.png  
|       |--- commit_25.png  
|       |--- commit_26.png  
|       |--- commit_27.png  
|       |--- commit_28.png  
|       |--- commit_29.png  
|       |--- commit_30.png  
|       |--- commit_31.png  
|       |--- commit_32.png  
|       |--- commit_33.png  
|       |--- logo_abet.png  
|       |--- logo_unsa.jpg  
|       |--- logo_episunsa.png  
|       |--- uml.png  
|   |--- commits.bash  
|   |--- Informe.pdf  
|   |--- Informe.tex  
|--- SERVER  
|   |--- .vscode  
|   |--- bin  
|   |--- lib  
|   |--- src  
|   |--- FX
```

```
|--- MainGame
|--- Classes
|--- Archer.java
|--- Knight.java
|--- Soldier.java
|--- Spearman.java
|--- Swordsman.java
|--- Board.java
|--- Utils
|--- BetterColor.java
|--- DBConnector.java
|--- MainGameOperation.java
|--- MainMenuOperation.java
|--- ServerConnection.java
|--- Utils.java
|--- VideogameConstants.java
|--- MainServer.java
|--- SERVER.jar
|--- VIDEOGAME
|--- .vscode
|--- bin
|--- JavaFX
|--- lib
|--- src
|--- FX
|--- MainGame
|--- Classes
|--- Archer.java
|--- Knight.java
|--- Soldier.java
|--- Spearman.java
|--- Swordsman.java
|--- Board.java
|--- MainGame.fxml
|--- MainGameController.java
|--- MainMenu
|--- MainMenu.fxml
|--- MainMenuController.java
|--- img
|--- action_attack.png
|--- action_move.png
|--- background_beach.png
|--- background_data.png
|--- background_desert.png
|--- background_forest.png
|--- background_meadow.png
|--- background_mountain.png
|--- settings.png
|--- tile_archer.png
|--- tile_knight.png
|--- tile_spearman.png
|--- tile_swordsman.png
|--- tile_tile.png
|--- waiting.png
|--- Utils
|--- BetterColor.java
```

```
|--- DBConnector.java
|--- MainGameOperation.java
|--- MainMenuOperation.java
|--- Resolution.java
|--- Tile.java
|--- Utils.java
|--- VideogameConstants.java
|--- Videogame.java
|--- VIDEOGAME.jar
```

9. Rúbricas

9.1. Entregable Informe

Tipo de Informe

Informe	
Latex	El informe está en formato PDF desde Latex, con un formato limpio (buena presentación) y facil de leer.

9.2. Rúbrica para el contenido del Informe y demostración

- El alumno debe marcar o dejar en blanco en celdas de la columna Checklist si cumplio con el ítem correspondiente.
- Si un alumno supera la fecha de entrega, su calificación será sobre la nota mínima aprobatoria, siempre y cuando cumpla con todos los items.
- El alumno debe autocalificarse en la columna Estudiante de acuerdo a la siguiente tabla:

Niveles de desempeño

Puntos	Nivel			
	Insatisfactorio 25 %	En Proceso 50 %	Satisfactorio 75 %	Sobresaliente 100 %
2.0	0.5	1.0	1.5	2.0
4.0	1.0	2.0	3.0	4.0

Rúbrica para contenido del Informe y demostración

Contenido y demostración		Puntos	Checklist	Estudiante	Profesor
1. GitHub	Hay enlace URL activo del directorio para el laboratorio hacia su repositorio GitHub con código fuente terminado y fácil de revisar.	2	X	2	
2. Commits	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X	3	
3. Código fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	1.5	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	1.5	
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	2	
6. Fechas	Las fechas de modificación del código fuente estan dentro de los plazos de fecha de entrega establecidos.	2	X	2	
7. Ortografía	El documento no muestra errores ortográficos.	2	X	1.5	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente, explicaciones puntuales pero precisas y un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4	X	4	
Total		20		17.5	

10. Referencias

- Aedo, M. y Castro, E. (2021). **FUNDAMENTOS DE PROGRAMACIÓN 2 - Tópicos de Programación Orientada a Objetos**. Editorial UNSA.

- **JavaFX (2023). Getting Started with JavaFX.** <https://openjfx.io/openjfx-docs/>