

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 29 Enero 2024

1. Tarea

■ Enunciado:

- Cree una versión del videojuego de estrategia usando componentes de GUI, bases de datos y archivos.

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.
```

493

494

To <https://github.com/cmestasz/fp2-23b.git>
fd709bc..68308c2 main -> main

Commit

.gitignore actualizado para solo publicar la carpeta src

main

cmestasz committed last week

1 parent 96d21f4 commit e8cc937

Showing 1 changed file with 3 additions and 0 deletions.

Whitespace Ignore whitespace Split Unified

Commit

Ejemplo dado en la documentacion de JavaFX

main

cmestasz committed last week

1 parent e8cc937 commit 118a77f

Showing 9 changed files with 155 additions and 0 deletions.

Whitespace Ignore whitespace Split Unified

Commit

Menú principal

main

cmestasz committed last week

1 parent 118a77f commit e9a3e21

Showing 9 changed files with 92 additions and 34 deletions.

Whitespace Ignore whitespace Split Unified

Commit

Controlador del menú principal

main

cmestasz committed last week

1 parent e9a3e21 commit 84ae869

Showing 5 changed files with 131 additions and 14 deletions.

Whitespace Ignore whitespace Split Unified

Commit

Modelo de servidor para manejar varias instancias locales del videojuego

main

cmestasz committed last week

1 parent 84ae869 commit 5e5d1a8

Showing 4 changed files with 133 additions and 56 deletions.

Whitespace Ignore whitespace Split Unified

Commit

Servidor completo

main

cmestasz committed last week

1 parent 5e5d1a8 commit 1d626a9

Showing 1 changed file with 33 additions and 11 deletions.

Whitespace Ignore whitespace Split Unified

Commit

Funcionalidad de servidor para el menú principal completa

main

cmestasz committed last week

1 parent 1d626a9 commit abc9d8d

Showing 5 changed files with 198 additions and 58 deletions.

Whitespace Ignore whitespace Split Unified

Commit

Inicio del juego vinculado entre ambas instancias

main

cmestasz committed last week

1 parent abc9d8d commit a9a8d44

Showing 9 changed files with 132 additions and 64 deletions.

Whitespace Ignore whitespace Split Unified

Commit

Cambio de estructura y vinculo entre ambos controladores
main
cmestas committed last week
1 parent: a7a8c44 commit: 0be0029

Showing 8 changed files with 77 additions and 35 deletions.

Whitespace Ignore whitespace Split Unified

Commit

Comentarios descriptivos en las partes confusas
main
cmestas committed last week
1 parent: 0be0029 commit: 38e2aaf

Showing 3 changed files with 21 additions and 12 deletions.

Whitespace Ignore whitespace Split Unified

Commit

Estructura del controlador del juego principal
main
cmestas committed last week
1 parent: 38e2aaf commit: 8c4d1bd

Showing 11 changed files with 86 additions and 82 deletions.

Whitespace Ignore whitespace Split Unified

Commit

Estructura visual del juego principal
main
cmestas committed last week
1 parent: 8c4d1bd commit: 49914f7

Showing 6 changed files with 124 additions and 35 deletions.

Whitespace Ignore whitespace Split Unified

Commit

Modelo de base de datos y cambios en el menu principal
main
cmestas committed last week
1 parent: 49914f7 commit: 5b878e5

Showing 6 changed files with 135 additions and 37 deletions.

Whitespace Ignore whitespace Split Unified

Commit

Clase que permite enviar y solicitar datos a la base de datos
main
cmestas committed 5 days ago
1 parent: 5b878e5 commit: e1375c5

Showing 3 changed files with 74 additions and 14 deletions.

Whitespace Ignore whitespace Split Unified

Commit

Implementacion de la clase DBConnector
main
cmestas committed 5 days ago
1 parent: e1375c5 commit: acca9f6

Showing 5 changed files with 101 additions and 70 deletions.

Whitespace Ignore whitespace Split Unified

Commit


Clases para el juego principal
main
cmestas committed 5 days ago
1 parent: acca9f6 commit: df6c1be


Showing 17 changed files with 186 additions and 23 deletions.

Whitespace Ignore whitespace Split Unified

Commit

Eleccion de reino y cambios en el menu principal

 main

 cimestasz committed 5 days ago

1 parent d9bc1be commit 790c088

Showing 5 changed files with 119 additions and 32 deletions.

Whitespace


Ignore whitespace


Split

Unified

Commit

El tablero ahora se conecta entre ambos jugadores

 main

 cimestasz committed 4 days ago

1 parent 790c088 commit 5f9c49f

Showing 13 changed files with 227 additions and 95 deletions.

Whitespace


Ignore whitespace


Split

Unified

Commit

Envio de mensajes y conexion en el juego principal

 main

 cimestasz committed 4 days ago

1 parent 5f9c49f commit 45de608

Showing 10 changed files with 270 additions and 75 deletions.

Whitespace


Ignore whitespace


Split

Unified

Commit

Nuevo sistema de mensajes

 main

 cimestasz committed 4 days ago

1 parent 45de608 commit 5207444

Showing 3 changed files with 35 additions and 16 deletions.

Whitespace


Ignore whitespace


Split

Unified

Commit

Chat de colores y mejor tratado

 main

 cimestasz committed 4 days ago

1 parent 5207444 commit 1aFcb77

Showing 3 changed files with 67 additions and 26 deletions.

Whitespace


Ignore whitespace


Split

Unified

Commit

Representacion visual de los soldados en ambos tableros

 main

 cimestasz committed 4 days ago

1 parent 1aFcb77 commit cc5eb65

Showing 13 changed files with 185 additions and 78 deletions.

Whitespace


Ignore whitespace


Split

Unified

Commit

Implementacion de los tipos de soldado y sus posibles acciones

 main

 cimestasz committed 2 days ago

1 parent cc5eb65 commit 78d26d2

Showing 23 changed files with 52 additions and 43 deletions.

Whitespace


Ignore whitespace

Split

Unified

Commit

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

 main

 cimestasz committed 2 days ago

1 parent 78d26d2 commit 4fc2a3f

Showing 26 changed files with 320 additions and 107 deletions.

Whitespace


Ignore whitespace


Split

Unified

Commit

Primera prueba de toda la funcionalidad completa

 main

 cmestas committed 2 days ago

1 parent: 4fc2a3f commit: 6520ee5

☐ Showing 6 changed files with 85 additions and 62 deletions.

Whitespace


Ignore whitespace


Split

Unified

Commit

Correccion de los ultimos errores

 main

 cmestas committed yesterday

1 parent: 6520ee5 commit: 9240a93

☐ Showing 12 changed files with 134 additions and 72 deletions.

Whitespace


Ignore whitespace


Split

Unified

Commit

Implementacion de más utilidades

 main

 cmestas committed yesterday

1 parent: 9240a93 commit: ea10ca9

☐ Showing 5 changed files with 27 additions and 13 deletions.

Whitespace


Ignore whitespace


Split

Unified

Commit

Ultimas correcciones de la logica y funciones

 main

 cmestas committed yesterday

1 parent: ea10ca9 commit: f9cf82b

☐ Showing 2 changed files with 4 additions and 1 deletion.

Whitespace


Ignore whitespace


Split

Unified

Commit

Correcciones de la base de datos

 main

 cmestas committed yesterday

1 parent: f9cf82b commit: 12becf8

☐ Showing 3 changed files with 15 additions and 7 deletions.

Whitespace


Ignore whitespace


Split

Unified

Commit

Implementacion de mejoras visuales y forma final

 main

 cmestas committed 10 hours ago

1 parent: 12becf8 commit: fd7890c

☐ Showing 47 changed files with 155 additions and 99 deletions.

Whitespace


Ignore whitespace


Split

Unified

Commit

Correcciones para el ejecutable y proyecto exportado

 main

 cmestas committed 1 hour ago

1 parent: fd7890c commit: 83ba99f

☐ Showing 9 changed files with 67 additions and 25 deletions.

Whitespace


Ignore whitespace


Split

Unified

Commit

Ejecutables del videojuego en una carpeta separada

 main

 cmestas committed 1 hour ago

1 parent: 83ba99f commit: 68388c2

☐ Showing 46 changed files with 569 additions and 1 deletion.

Whitespace

Ignore whitespace

Split

Unified

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 import javax.swing.JOptionPane;
7
```

```
8 public class DBConnector {
9     private String user;
10    private String password;
11    private Connection connection;
12
13    public static void main(String[] args) {
14        new DBConnector();
15    }
16
17    public DBConnector() {
18        try {
19            Class.forName("com.mysql.cj.jdbc.Driver");
20
21            Connection connection = readLogin();
22            if (connection == null) {
23                while (connection == null) {
24                    user = JOptionPane.showInputDialog(null, "Ingreso usuario", "Primera
25                        conexión",
26                        JOptionPane.QUESTION_MESSAGE);
27                    password = JOptionPane.showInputDialog(null, "Ingreso clave", "Primera
28                        conexión",
29                        JOptionPane.QUESTION_MESSAGE);
30                    // usuario = "query";
31                    // clave = "123456789";
32                    connection = login("");
33                }
34                createDatabase();
35                writeLogin();
36
37                // connection.prepareStatement("CREATE DATABASE test").execute();
38                // ResultSet resultados = conexion.prepareStatement("SELECT * FROM alumnos WHERE
39                // nombre='Juan'")
40                // .executeQuery();
41
42                System.out.println("Conexion exitosa");
43            } catch (Exception e) {
44                e.printStackTrace();
45            }
46        }
47
48        private Connection readLogin() throws IOException {
49            File dblogin = new File("data/dblogin.dat");
50            if (dblogin.exists()) {
51                DataInputStream in = new DataInputStream(new FileInputStream(dblogin));
52                user = Utils.readString(in);
53                password = Utils.readString(in);
54                return login("videogame");
55            }
56            return null;
57        }
58
59        private Connection login(String database) {
60            String url = "jdbc:mysql://localhost:3306/" + database;
61            try {
62                connection = DriverManager.getConnection(url, user, password);
63            }
64        }
65    }
66 }
```

```
62         return connection;
63     } catch (Exception e) {
64         JOptionPane.showMessageDialog(null, "Usuario o clave incorrectos", "Error",
65             JOptionPane.ERROR_MESSAGE);
66         return null;
67     }
68 }
69 private void writeLogin() throws IOException {
70     File directory = new File("data/");
71     if (!directory.exists())
72         directory.mkdirs();
73     DataOutputStream out = new DataOutputStream(new FileOutputStream("data/dblogin.dat"));
74     out.writeChars(user);
75     out.writeChar(0);
76     out.writeChars(password);
77     out.writeChar(0);
78     out.close();
79 }
80
81 private void createDatabase() throws SQLException {
82     connection.prepareStatement("CREATE DATABASE videogame;\n").execute();
83     connection = login("videogame");
84     connection.prepareStatement(
85         "CREATE TABLE players (id int NOT NULL AUTO_INCREMENT, name varchar(30) NOT
86             NULL, password varchar(30) NOT NULL, PRIMARY KEY (id))"
87         ).execute();
88     connection.prepareStatement(
89         "CREATE TABLE matches (id int NOT NULL AUTO_INCREMENT, winner_id int NOT NULL,
90             loser_id int NOT NULL, PRIMARY KEY (id), INDEX winner_id (winner_id),
91             INDEX loser_id (loser_id))"
92         ).execute();
93     connection.prepareStatement(
94         "ALTER TABLE matches ADD CONSTRAINT winner_id FOREIGN KEY (winner_id)
95             REFERENCES players(id) ON DELETE RESTRICT ON UPDATE RESTRICT")
96         .execute();
97     connection.prepareStatement(
98         "ALTER TABLE matches ADD CONSTRAINT loser_id FOREIGN KEY (loser_id) REFERENCES
99             players(id) ON DELETE RESTRICT ON UPDATE RESTRICT")
100         .execute();
101 }
102
103 public int loginPlayer(String name, String password) {
104     try {
105         String query = String.format("SELECT id FROM players WHERE name = '%s' AND
106             password = '%s'", name,
107             password);
108         ResultSet results = connection.prepareStatement(query).executeQuery();
109         if (results.next())
110             return results.getInt(1);
111     } catch (Exception e) {
112         e.printStackTrace();
113     }
114     return -1;
115 }
```

```
111 public int[] getWinsLoses(int id) {
112     try {
113         int[] totals = new int[2];
114         String query = String.format("SELECT COUNT(*) from matches WHERE winner_id =
115             '%d'", id);
116         ResultSet results = connection.prepareStatement(query).executeQuery();
117         if (results.next())
118             totals[0] = results.getInt(1);
119
120         query = String.format("SELECT COUNT(*) from matches WHERE loser_id = '%d'", id);
121         results = connection.prepareStatement(query).executeQuery();
122         if (results.next())
123             totals[1] = results.getInt(1);
124
125         return totals;
126     } catch (Exception e) {
127         e.printStackTrace();
128         return null;
129     }
130
131 public void registerPlayer(String name, String password) {
132     try {
133         String query = String.format("INSERT INTO players (name, password) VALUES ('%s',
134             '%s')", name, password);
135         connection.prepareStatement(query).execute();
136     } catch (Exception e) {
137         e.printStackTrace();
138     }
139
140 public void createMatch(int winner_id, int loser_id) {
141     try {
142         String query = String.format("INSERT INTO matches (winner_id, loser_id) VALUES
143             ('%d', '%d')", winner_id,
144             loser_id);
145         System.out.println(query);
146         connection.prepareStatement(query).execute();
147     } catch (Exception e) {
148         e.printStackTrace();
149     }
150 }
```

- 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 }
```

```
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
74     public void init(MainMenuController menuController, Resolution resolution, Stage
       menuStage, Stage gameStage,
75         Board board,
76         int idConnection, String matchCode, String pName, String eName, int idPlayer, int
       idEnemy) {
77         this.menuController = menuController;
78         this.resolution = resolution;
79         this.width = resolution.getWidth();
80         this.height = resolution.getHeight();
81         this.menuStage = menuStage;
82         this.gameStage = gameStage;
83         this.board = board;
```

```
84     army1 = board.getArmy1();
85     army2 = board.getArmy2();
86     this.idConnection = idConnection;
87     this.kingdomPlayer = board.getKingdomPlayer();
88     this.kingdomEnemy = board.getKingdomEnemy();
89     this.matchCode = matchCode;
90     this.pName = pName;
91     this.eName = eName;
92     this.idPlayer = idPlayer;
93     this.idEnemy = idEnemy;
94
95     initButtons();
96     initBackground();
97     initDataFields();
98     initChat();
99
100     actionsPane.setPrefWidth(width * 0.15);
101     actionsPane.setPrefHeight(width * 0.05);
102     setStyleColor(moveActionPane, SELECTED_COLOR);
103
104     dbConnector = new DBConnector();
105     setConnection();
106 }
107
108 public void initialize() {
109
110 }
111
112 public void sendMessage() {
113     String message = String.format("%s: %s%n", pName, chatInput.getText());
114     printMessage(message, PLAYER_COLOR);
115     try {
116         DataOutputStream out = new DataOutputStream(new FileOutputStream(connectionFile));
117         out.writeInt(OPERATION_CHAT);
118         Utils.writeStrings(out, new String[] { matchCode, message });
119         out.close();
120     } catch (Exception e) {
121         e.printStackTrace();
122     }
123     chatInput.setText("");
124 }
125
126 public void setActionMove() {
127     setStyleColor(moveActionPane, SELECTED_COLOR);
128     setStyleColor(attackActionPane, null);
129     selectedAction = "MOVER";
130 }
131
132 public void setActionAttack() {
133     setStyleColor(attackActionPane, SELECTED_COLOR);
134     setStyleColor(moveActionPane, null);
135     selectedAction = "ATACAR";
136 }
137
138 public void closeMessage() {
139     messagePane.setVisible(false);
```

```
140     if (gameEnded) {
141         dataReceiver.endGame();
142         menuStage.show();
143         menuController.restartMenu();
144         gameStage.close();
145     }
146 }
147
148 private void initButtons() {
149     for (int i = 0; i < SIZE; i++) {
150         for (int j = 0; j < SIZE; j++) {
151             String key = generateKey(i, j);
152             double size = 1.0 * resolution.getHeight() / SIZE;
153             HashMap<String, Soldier> army1 = board.getArmy1();
154             HashMap<String, Soldier> army2 = board.getArmy2();
155             Tile tile;
156             Soldier soldier;
157
158             if (army1.containsKey(key)) {
159                 soldier = army1.get(key);
160                 tile = new Tile(soldier.getCurrentHealth(), soldier.getTypeFile(), size,
161                     i, j);
162                 setStyleColor(tile, PLAYER_COLOR_TRANS);
163             } else if (army2.containsKey(key)) {
164                 soldier = army2.get(key);
165                 tile = new Tile(soldier.getCurrentHealth(), soldier.getTypeFile(), size,
166                     i, j);
167                 setStyleColor(tile, ENEMY_COLOR_TRANS);
168             } else {
169                 tile = new Tile(0, "tile", size, i, j);
170             }
171             tiles[i][j] = tile;
172
173             tile.setOnMouseClicked(this::handleClick);
174             uiBoard.add(tile, i, j);
175         }
176     }
177
178 private void setStyleColor(Region pane, BetterColor color) {
179     if (color == null) {
180         pane.setStyle("-fx-background-color: none;");
181     } else {
182         pane.setStyle(String.format("-fx-background-color: %s;", color.getRGBA()));
183     }
184 }
185
186 private void initBackground() {
187     boardBackground.setFitWidth(width);
188     boardBackground.setFitHeight(height);
189     boardBackground.setImage(new Image(String.format("img/background_%s.png",
190         board.getTerrainFile())));
191
192     dataBackground.setFitWidth(width - height);
193     dataBackground.setFitHeight(height);
194     dataBackground.setImage(new Image("img/background_data.png"));
```

```
193     }
194
195     private void initDataFields() {
196         playerData.setText(String.format("%s: %s%n", pName, kingdomPlayer));
197         enemyData.setText(String.format("%s: %s%n", eName, kingdomEnemy));
198     }
199
200     private void initChat() {
201         chatOutput.setPrefHeight(height * 0.4);
202     }
203
204     private void handleClick(MouseEvent event) {
205         Tile tile = (Tile) event.getSource();
206
207         if (!tryDoAction(tile)) {
208             if (board.getArmy1().containsKey(tile.getKey())) {
209                 selectedTile = tile;
210                 showActionsMenu();
211             } else {
212                 selectedTile = null;
213                 removeActionsMenu();
214             }
215         }
216     }
217
218     private void showActionsMenu() {
219         actionsPane.setVisible(true);
220     }
221
222     private void removeActionsMenu() {
223         actionsPane.setVisible(false);
224     }
225
226     private boolean tryDoAction(Tile otherTile) {
227         if (selectedTile != null && playerTurn) {
228             String otherKey = otherTile.getKey();
229             try {
230                 DataOutputStream out;
231                 int sI = selectedTile.getI();
232                 int sJ = selectedTile.getJ();
233                 int oI = otherTile.getI();
234                 int oJ = otherTile.getJ();
235                 Soldier selectedSoldier = army1.get(generateKey(sI, sJ));
236                 int distance;
237                 switch (selectedAction) {
238                     case "MOVER":
239                         distance = selectedSoldier.getTypeFile().equals("knight") ? 2 : 1;
240                         if (selectedTile.isConnected(otherTile, distance) &&
241                             !army1.containsKey(otherKey)
242                             && !army2.containsKey(otherKey)) {
243                             moveSoldier(true, sI, sJ, oI, oJ);
244                             removeActionsMenu();
245
246                             out = new DataOutputStream(new FileOutputStream(connectionFile));
247                             out.writeInt(OPERATION_MOVE);
248                             Utils.writeString(out, matchCode);
249                         }
250                     }
251             } catch (IOException e) {
252                 e.printStackTrace();
253             }
254         }
255     }
256 }
```

```
248         Utils.writeIdxs(out, sI, sJ, oI, oJ);
249         out.close();
250
251         playerTurn = false;
252         selectedTile = null;
253         out.close();
254         return true;
255     }
256     showMessage("Movimiento no valido.");
257     break;
258     case "ATACAR":
259         distance = selectedSoldier.getTypeFile().equals("archer") ? 2 : 1;
260         if (selectedTile.isConnected(otherTile, distance) &&
261             army2.containsKey(otherKey)) {
262             attackSoldier(true, sI, sJ, oI, oJ);
263             removeActionsMenu();
264
265             out = new DataOutputStream(new FileOutputStream(connectionFile));
266             out.writeInt(OPERATION_ATTACK);
267             Utils.writeString(out, matchCode);
268             Utils.writeIdxs(out, sI, sJ, oI, oJ);
269             out.close();
270
271             playerTurn = false;
272             selectedTile = null;
273             out.close();
274             return true;
275         }
276         showMessage("Ataque no valido.");
277         break;
278     }
279     catch (Exception e) {
280         e.printStackTrace();
281     }
282     return false;
283 }
284
285 // Métodos que funcionan en ambos sentidos, host -> guest o guest -> host
286 private void moveSoldier(boolean isPlayer, int iSelected, int jSelected, int iOther, int
    jOther) {
287     Tile selectedTile = tiles[iSelected][jSelected];
288     Tile otherTile = tiles[iOther][jOther];
289     String selectedKey = selectedTile.getKey();
290     String otherKey = otherTile.getKey();
291
292     HashMap<String, Soldier> army = null;
293     BetterColor color = null;
294     if (isPlayer) {
295         army = army1;
296         color = PLAYER_COLOR_TRANS;
297     } else {
298         army = army2;
299         color = ENEMY_COLOR_TRANS;
300     }
301 }
```

```
302     Soldier soldier = army.remove(selectedKey);
303     army.put(otherKey, soldier);
304     selectedTile.setImageAndhealth("tile", 0);
305     setStyleColor(selectedTile, null);
306     otherTile.setImageAndhealth(soldier.getTypeFile(), soldier.getCurrentHealth());
307     setStyleColor(otherTile, color);
308
309     String message = soldier + " se mueve." + "\n";
310     if (isPlayer)
311         playerData.appendText(message);
312     else
313         enemyData.appendText(message);
314 }
315
316 private void attackSoldier(boolean isPlayer, int iSelected, int jSelected, int iOther,
317     int jOther) {
318     Tile selectedTile = tiles[iSelected][jSelected];
319     Tile otherTile = tiles[iOther][jOther];
320     String selectedKey = selectedTile.getKey();
321     String otherKey = otherTile.getKey();
322
323     Soldier soldierAttacks = null;
324     Soldier soldierReceives = null;
325     if (isPlayer) {
326         soldierAttacks = army1.get(selectedKey);
327         soldierReceives = army2.get(otherKey);
328     } else {
329         soldierAttacks = army2.get(selectedKey);
330         soldierReceives = army1.get(otherKey);
331     }
332
333     int damage = soldierAttacks.attack(soldierReceives);
334     otherTile.setImageAndhealth(soldierReceives.getTypeFile(),
335         soldierReceives.getCurrentHealth());
336     String message = String.format("%s ataca a %s con %d de daño\n", soldierAttacks,
337         soldierReceives, damage);
338     if (isPlayer)
339         playerData.appendText(message);
340     else
341         enemyData.appendText(message);
342
343     if (soldierReceives.getCurrentHealth() <= 0) {
344         soldierAttacks.heal();
345         selectedTile.setImageAndhealth(soldierAttacks.getTypeFile(),
346             soldierAttacks.getCurrentHealth());
347         otherTile.setImageAndhealth("tile", 0);
348         setStyleColor(otherTile, null);
349
350         message = soldierReceives + " ha muerto!\n";
351         if (isPlayer) {
352             playerData.appendText(message);
353             army2.remove(otherKey);
354             if (army2.size() == 0) {
355                 if (idEnemy != 0)
356                     dbConnector.createMatch(idPlayer, idEnemy);
357                 endGame(pName, kingdomPlayer);
358             }
359         }
360     }
```



```
354     }
355     } else {
356         enemyData.appendText(message);
357         army1.remove(otherKey);
358         if (army1.size() == 0) {
359             if (idEnemy != 0)
360                 dbConnector.createMatch(idEnemy, idPlayer);
361             endGame(eName, kingdomEnemy);
362         }
363     }
364 }
365
366 }
367
368 private void endGame(String name, String kingdom) {
369     showMessage(String.format("%s ha ganado con el reino %s!", name, kingdom));
370     gameEnded = true;
371 }
372
373 private void printMessage(String message, BetterColor color) {
374     Text messageText = new Text(message);
375     messageText.setFont(Font.font("Book Antiqua"));
376     messageText.setFill(color.getColor());
377     messageText.setWrappingWidth(width - height);
378
379     ObservableList<Node> children = chatOutput.getChildren();
380     children.add(children.size() - 1, messageText);
381     chatOutputPane.setVvalue(1);
382 }
383
384 private void setConnection() {
385     path = "connections/" + idConnection + ".dat";
386     connectionFile = new File(path);
387     try {
388         connectionFile.createNewFile();
389         dataReceiver = new DataReceiver();
390         dataReceiver.start();
391     } catch (Exception e) {
392         e.printStackTrace();
393     }
394 }
395
396 private String generateKey(int i, int j) {
397     return i + "," + j;
398 }
399
400 private void showMessage(String message) {
401     messagePane.setVisible(true);
402     messageOutput.setText(message);
403 }
404
405 private class DataReceiver extends Thread {
406     private File matchFile = new File(path);
407     private long lastModified = matchFile.lastModified();
408     private boolean gameEnded;
409 }
```

```

410     public void run() {
411         try {
412             while (!gameEnded) {
413                 // Comprueba si el archivo de la partida ha sido modificado
414                 if (matchFile.lastModified() != lastModified) {
415                     DataInputStream in = new DataInputStream(new
416                         FileInputStream(matchFile));
417                     int response = in.readInt();
418                     switch (response) {
419                         // Mensaje de chat
420                         case RESPONSE_CHAT:
421                             String message = Utils.readString(in);
422                             Platform.runLater(() -> {
423                                 printMessage(message, ENEMY_COLOR);
424                             });
425                             break;
426
427                         // Movimientos y ataques
428                         case RESPONSE_MOVE:
429                         case RESPONSE_ATTACK:
430                             int sI = in.readInt();
431                             int sJ = in.readInt();
432                             int oI = in.readInt();
433                             int oJ = in.readInt();
434
435                             Platform.runLater(() -> {
436                                 showActionsMenu();
437                                 playerTurn = true;
438                                 if (response == RESPONSE_MOVE)
439                                     moveSoldier(false, sI, sJ, oI, oJ);
440                                 else
441                                     attackSoldier(false, sI, sJ, oI, oJ);
442                             });
443                             break;
444                         }
445                     lastModified = matchFile.lastModified();
446                     in.close();
447                 }
448                 sleep(500);
449             }
450         } catch (Exception e) {
451             e.printStackTrace();
452         }
453     }
454
455     public void endGame() {
456         gameEnded = true;
457     }
458 }
459 }

```

- 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.
- 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"
25     fx:controller="FX.MainGame.MainGameController">
26     <children>
27         <ImageView fx:id="boardBackground" fitHeight="150.0" fitWidth="200.0"
28             pickOnBounds="true" preserveRatio="true" />
29         <HBox>
30             <children>
31                 <GridPane fx:id="uiBoard" maxHeight="-Infinity" maxWidth="-Infinity"
32                     minHeight="-Infinity" minWidth="-Infinity">
33                     <columnConstraints>
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 hgrow="SOMETIMES" />
42                     </columnConstraints>
43                     <rowConstraints>
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"
58             pickOnBounds="true" preserveRatio="true" />
59         <VBox>
60             <children>
61                 <HBox>
62                     <children>
63                         <TextArea fx:id="playerData" blendMode="MULTIPLY">
64                             <font>
65                                 <Font name="Book Antiqua" size="10.0" />
66                             </font></TextArea>
67                         <TextArea fx:id="enemyData" blendMode="MULTIPLY">
68                             <font>
69                                 <Font name="Book Antiqua" size="10.0" />
70                             </font></TextArea>
71                     </children>
72                 </HBox>
73                 <VBox alignment="CENTER">
74                     <children>
75                         <ScrollPane fx:id="chatOutputPane" blendMode="MULTIPLY"
76                             hbarPolicy="NEVER" vbarPolicy="ALWAYS">
77                             <content>
78                                 <VBox fx:id="chatOutput" style="-fx-background-color:
79                                     white;">
80                                     <children>
81                                         <Text strokeType="OUTSIDE" strokeWidth="0.0">
82                                             <font>
83                                                 <Font size="18.0" />
84                                             </font>
85                                         </Text>
86                                     </children>
87                                 </VBox>
88                             </content>
89                         </ScrollPane>
90                         <TextField fx:id="chatInput" blendMode="MULTIPLY"
91                             onAction="#sendMessage" promptText="Envia un mensaje!">
92                             <VBox.margin>
93                                 <Insets bottom="15.0" />
94                             </VBox.margin>
95                             <font>
96                                 <Font name="Book Antiqua" size="12.0" />
97                             </font></TextField>
98                         <TilePane fx:id="actionsPane" alignment="CENTER" hgap="30.0">
99                             <children>
100                                 <VBox fx:id="moveActionPane" onMouseClicked="#setActionMove"
101                                     TilePane.alignment="CENTER">
102                                     <children>
103                                         <Label alignment="CENTER" prefWidth="50.0" text="Mover">
104                                             <font>

```

```

100         <Font name="Book Antiqua" size="12.0" />
101     </font></Label>
102     <ImageView fitHeight="50.0" fitWidth="50.0"
103         pickOnBounds="true" preserveRatio="true">
104         <image>
105             <Image url="@../img/action_move.png" />
106         </image>
107     </ImageView>
108 </children>
109 </VBox>
110 <VBox fx:id="attackActionPane"
111     onMouseClicked="#setActionAttack"
112     TilePane.alignment="CENTER">
113     <children>
114         <Label alignment="CENTER"
115             maxWidth="1.7976931348623157E308" prefWidth="50.0"
116             text="Atacar">
117             <font>
118                 <Font name="Book Antiqua" size="12.0" />
119             </font></Label>
120             <ImageView fitHeight="50.0" fitWidth="50.0"
121                 pickOnBounds="true" preserveRatio="true">
122                 <image>
123                     <Image url="@../img/action_attack.png" />
124                 </image>
125             </ImageView>
126         </children>
127     </VBox>
128 </children>
129 </TilePane>
130 </children>
131 </VBox>
132 </children>
133 </StackPane>
134 </children>
135 </HBox>
136 <Pane fx:id="messagePane" visible="false">
137     <children>
138         <TitledPane animated="false" collapsible="false" layoutX="274.0" layoutY="74.0"
139             prefHeight="139.0" prefWidth="279.0" text="Mensaje">
140             <content>
141                 <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="180.0"
142                     prefWidth="200.0">
143                     <children>
144                         <TextArea fx:id="messageOutput" layoutX="-1.0" prefHeight="81.0"
145                             prefWidth="279.0" wrapText="true">
146                             <font>
147                                 <Font name="Book Antiqua" size="14.0" />
148                             </font>
149                         </TextArea>
150                         <Button layoutX="122.0" layoutY="84.0" mnemonicParsing="false"
151                             onAction="#closeMessage" text="OK">
152                             <font>
153                                 <Font name="Book Antiqua" size="12.0" />

```

```
146         </font>
147     </Button>
148 </children>
149 </AnchorPane>
150 </content>
151 <font>
152     <Font name="Book Antiqua" size="12.0" />
153 </font>
154 </TitledPane>
155 </children>
156 </Pane>
157 </children>
158 </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  @FXML
37  private TextField nameInput;
38  @FXML
39  private TextField passwordInput;
40  @FXML
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
65  public void setStage(Stage stage) {
66      this.stage = stage;
67  }
68
69  public void initialize() throws IOException {
70      try {
71          RESOLUTIONS.addAll(new Resolution(850, 480), new Resolution(1280, 720), new
              Resolution(1366, 768),
72              new Resolution(1920, 1080));
73          resolutionInput.setItems(RESOLUTIONS);
74          resolutionInput.setValue(RESOLUTIONS.get(0));
75          resolution = resolutionInput.getValue();
76
77          KINGDOMS.addAll("INGLATERRA", "FRANCIA", "CASTILLA-ARAGÓN", "MOROS", "SACRO
              IMPERIO");
78          kingdomInput.setItems(KINGDOMS);
79
80          dbConnector = new DBConnector();
81          setConnection();
82      } catch (Exception e) {
83          FileWriter writer = new FileWriter("error.log");
84          writer.write(e.getMessage());
85          writer.close();
86      }
87  }
88
89  public void setKingdom() {
```

```
90     pKingdom = kingdomInput.getValue();
91     playerKingdom.setText(pKingdom);
92 }
93
94 public void toggleSettings() {
95     settingsPane.setVisible(!settingsPane.isVisible());
96 }
97
98 public void setResolution() {
99     resolution = resolutionInput.getValue();
100 }
101
102 public void createMatch() {
103     if (checkName() && checkKingdom()) {
104         matchCode = "";
105         for (int i = 0; i < CODE_LENGTH; i++)
106             matchCode += (char) ('A' + (int) (Math.random() * 26));
107
108         createMatchCode.setText(matchCode);
109         try {
110             DataOutputStream out = new DataOutputStream(new
111                 FileOutputStream(connectionFile));
112             out.writeInt(OPERATION_CREATE);
113             Utils.writeStrings(out, new String[] { matchCode, pName, pKingdom });
114             out.close();
115         } catch (Exception e) {
116             e.printStackTrace();
117         }
118     }
119
120     public void joinMatch() {
121         matchCode = joinMatchCode.getText();
122         if (checkName() && checkKingdom() && matchCode.length() == CODE_LENGTH) {
123             try {
124                 DataOutputStream out = new DataOutputStream(new
125                     FileOutputStream(connectionFile));
126                 out.writeInt(OPERATION_JOIN);
127                 Utils.writeStrings(out, new String[] { matchCode, pName, pKingdom });
128                 out.writeInt(idPlayer);
129             } catch (Exception e) {
130                 e.printStackTrace();
131             }
132         }
133
134         public void startMatch() {
135             if (checkName() && checkEnemy() && checkKingdom()) {
136                 try {
137                     DataOutputStream out = new DataOutputStream(new
138                         FileOutputStream(connectionFile));
139                     out.writeInt(OPERATION_START);
140                     Utils.writeString(out, matchCode);
141                     out.close();
142
143                     board = new Board(pKingdom, eKingdom);
```



```
143         ObjectOutputStream outObj = new ObjectOutputStream(  
144             new FileOutputStream("connections/" + idConnection + ".obj"));  
145         outObj.writeObject(board);  
146         outObj.close();  
147     } catch (Exception e) {  
148         e.printStackTrace();  
149     }  
150  
151     createGameStage();  
152 }  
153 }  
154  
155 public void login() {  
156     String name = nameInput.getText();  
157     idPlayer = dbConnector.loginPlayer(name, passwordInput.getText());  
158     if (idPlayer == -1) {  
159         showMessage("Usuario no encontrado.");  
160     } else {  
161         showMessage("Acceso correcto.");  
162         pName = name;  
163         playerName.setText(pName);  
164         nameInput.setText("");  
165         passwordInput.setText("");  
166     }  
167 }  
168  
169 public void register() {  
170     pName = nameInput.getText();  
171     dbConnector.registerPlayer(pName, passwordInput.getText());  
172     showMessage("Usuario creado correctamente.");  
173     login();  
174 }  
175  
176 public void getStatistics() {  
177     if (checkName()) {  
178         int[] status = dbConnector.getWinsLoses(idPlayer);  
179         showMessage(String.format("W: %d | L: %d", status[0], status[1]));  
180     }  
181 }  
182  
183 public void closeMessage() {  
184     messagePane.setVisible(false);  
185 }  
186  
187 public void restartMenu() {  
188     createMatchCode.setText("");  
189     joinMatchCode.setText("");  
190     enemyName.setText("");  
191     enemyKingdom.setText("");  
192     startButton.setDisable(false);  
193     dataReceiver = new DataReceiver();  
194     dataReceiver.start();  
195 }  
196  
197 private void setConnection() {  
198     if (connectionFile == null) {
```

```
199     path = "connections/" + idConnection + ".dat";
200     connectionFile = new File(path);
201     while (connectionFile.exists()) {
202         idConnection++;
203         path = "connections/" + idConnection + ".dat";
204         connectionFile = new File(path);
205     }
206 }
207 try {
208     connectionFile.createNewFile();
209     dataReceiver = new DataReceiver();
210     dataReceiver.start();
211 } catch (Exception e) {
212     e.printStackTrace();
213 }
214 }
215
216 private boolean checkName() {
217     boolean nameSet = pName != null;
218     if (!nameSet)
219         showMessage("Crea o accede a tu cuenta!");
220     return nameSet;
221 }
222
223 private boolean checkEnemy() {
224     boolean enemySet = eName != null;
225     if (!enemySet)
226         showMessage("Crea o únete a una partida!");
227     return enemySet;
228 }
229
230 private boolean checkKingdom() {
231     boolean kingdomSet = pKingdom != null;
232     if (!kingdomSet)
233         showMessage("Escoge un reino!");
234     return kingdomSet;
235 }
236
237 private void createGameStage() {
238     dataReceiver.startGame();
239     stage.hide();
240     new MainGame(this);
241 }
242
243 private void showMessage(String message) {
244     messagePane.setVisible(true);
245     messageOutput.setText(message);
246 }
247
248 // Clase interna para el receptor de datos en un hilo separado
249 private class DataReceiver extends Thread {
250     private File matchFile = new File(path);
251     private long lastModified = matchFile.lastModified();
252     private boolean gameStarted;
253
254     public void run() {
```

```
255     try {
256         while (!gameStarted) {
257             // Comprueba si el archivo de la partida ha sido modificado
258             if (matchFile.lastModified() != lastModified) {
259                 DataInputStream in = new DataInputStream(new
260                     FileInputStream(matchFile));
261                 int response = in.readInt();
262                 String name, kingdom;
263                 switch (response) {
264                     // Respuesta del anfitrión
265                     case RESPONSE_HOST:
266                         name = Utils.readString(in);
267                         kingdom = Utils.readString(in);
268                         int idOther = in.readInt();
269                         // Actualiza el nombre del oponente en la interfaz de usuario
270                         Platform.runLater(() -> {
271                             setEnemy(name, kingdom);
272                             idEnemy = idOther;
273                         });
274                         break;
275                     // Respuesta del invitado
276                     case RESPONSE_GUEST:
277                         name = Utils.readString(in);
278                         kingdom = Utils.readString(in);
279                         if (name.equals("")) {
280                             showMessage("La partida no existe.");
281                         } else {
282                             // Actualiza el nombre del oponente en la interfaz de
283                             // usuario y desactiva el
284                             // botón de inicio
285                             Platform.runLater(() -> {
286                                 setEnemy(name, kingdom);
287                                 startButton.setDisable(true);
288                             });
289                         }
290                         break;
291                     // Respuesta de inicio de la partida
292                     case RESPONSE_START:
293                         File objFile = new File("connections/" + idConnection + ".obj");
294                         ObjectInputStream inObj = new ObjectInputStream(new
295                             FileInputStream(objFile));
296                         board = (Board) inObj.readObject();
297                         inObj.close();
298                         objFile.delete();
299                         // Inicia el juego principal
300                         Platform.runLater(() -> {
301                             createGameStage();
302                         });
303                     }
304                     lastModified = matchFile.lastModified();
305                     in.close();
306                 }
307                 sleep(1000);
308             }
309         } catch (Exception e) {
```

```

308         e.printStackTrace();
309     }
310 }
311
312 public void startGame() {
313     gameStarted = true;
314 }
315
316 private void setEnemy(String name, String kingdom) {
317     eName = name;
318     eKingdom = kingdom;
319     enemyName.setText(eName);
320     enemyKingdom.setText(eKingdom);
321 }
322 }
323
324 private class MainGame {
325     public MainGame(MainMenuController mainMenuController) {
326         try {
327             // Carga el archivo FXML del juego principal y configura la escena
328             FXMLLoader loader = new
329                 FXMLLoader(getClass().getResource("/FX/MainGame/MainGame.fxml"));
330             Parent root = loader.load();
331
332             Stage mainGame = new Stage();
333             mainGame.setTitle("Main Game");
334             mainGame.setScene(new Scene(root, resolution.getWidth(),
335                 resolution.getHeight()));
336             mainGame.setResizable(false);
337             mainGame.show();
338
339             MainGameController controller = loader.getController();
340             controller.init(mainMenuController, resolution, stage, mainGame, board,
341                 idConnection, matchCode, pName,
342                 eName,
343                 idPlayer, idEnemy);
344         } catch (Exception e) {
345             e.printStackTrace();
346         }
347     }
348 }

```

- 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 tanto de chats como de movimientos.
- Es el encargado de realizar la conexión con la base de datos usando la clase DBConnector y con el servidor usando un archivo.
- 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="27.0" layoutY="248.0" mnemonicParsing="false" onAction="#createMatch"
24       text="Crear partida">
25       <font>
26         <Font name="Book Antiqua" size="12.0" />
27       </font></Button>
28     <Button layoutX="27.0" layoutY="286.0" mnemonicParsing="false" onAction="#joinMatch"
29       text="Unirse a partida">
30       <font>
31         <Font name="Book Antiqua" size="12.0" />
32       </font></Button>
33     <TextField fx:id="createMatchCode" editable="false" layoutX="150.0" layoutY="248.0"
34       prefHeight="25.0" prefWidth="150.0" promptText="Código">
35       <font>
36         <Font name="Book Antiqua" size="12.0" />
37       </font></TextField>
38     <TextField fx:id="joinMatchCode" layoutX="150.0" layoutY="286.0" prefHeight="25.0"
39       prefWidth="150.0" promptText="Código">
40       <font>
41         <Font name="Book Antiqua" size="12.0" />
42       </font></TextField>
43     <Label layoutX="113.0" layoutY="223.0" text="SALA DE ESPERA">
44       <font>
45         <Font name="Book Antiqua" size="14.0" />
46       </font></Label>
47     <ImageView fitHeight="225.0" fitWidth="225.0" layoutX="52.0" layoutY="326.0"
48       pickOnBounds="true" preserveRatio="true">
49       <image>
50         <Image url="@../img/waiting.png" />
51       </image>
52     </ImageView>
53     <Button fx:id="startButton" layoutX="141.0" layoutY="564.0" mnemonicParsing="false"
54       onAction="#startMatch" text="Iniciar">
55       <font>
```

```
50     <Font name="Book Antiqua" size="12.0" />
51     </font></Button>
52     <TextField fx:id="nameInput" layoutX="60.0" layoutY="66.0" prefHeight="25.0"
53         prefWidth="212.0" promptText="Nombre">
54         <font>
55             <Font name="Book Antiqua" size="12.0" />
56         </font></TextField>
57     <Label fx:id="playerName" layoutX="71.0" layoutY="343.0" prefHeight="17.0"
58         prefWidth="150.0">
59         <font>
60             <Font name="Book Antiqua" size="16.0" />
61         </font></Label>
62     <Label layoutX="141.0" layoutY="416.0" text="VS">
63         <font>
64             <Font name="Gill Sans MT" size="40.0" />
65         </font>
66     </Label>
67     <Label fx:id="enemyName" alignment="CENTER_RIGHT" layoutX="110.0" layoutY="512.0"
68         prefHeight="17.0" prefWidth="150.0">
69         <font>
70             <Font name="Book Antiqua" size="16.0" />
71         </font></Label>
72     <Label layoutX="131.0" layoutY="39.0" text="USUARIO">
73         <font>
74             <Font name="Book Antiqua" size="14.0" />
75         </font>
76     </Label>
77     <PasswordField fx:id="passwordInput" layoutX="61.0" layoutY="100.0" prefHeight="25.0"
78         prefWidth="212.0" promptText="Contraseña">
79         <font>
80             <Font name="Book Antiqua" size="12.0" />
81         </font></PasswordField>
82     <Button layoutX="10.0" layoutY="135.0" mnemonicParsing="false" onAction="#register"
83         text="Crear cuenta">
84         <font>
85             <Font name="Book Antiqua" size="12.0" />
86         </font></Button>
87     <Button layoutX="102.0" layoutY="135.0" mnemonicParsing="false" onAction="#login"
88         text="Ingresar a cuenta">
89         <font>
90             <Font name="Book Antiqua" size="12.0" />
91         </font></Button>
92     <Button layoutX="218.0" layoutY="135.0" mnemonicParsing="false"
93         onAction="#getStatistics" text="Ver estadísticas">
94         <font>
95             <Font name="Book Antiqua" size="12.0" />
96         </font></Button>
97     <Label fx:id="playerKingdom" layoutX="71.0" layoutY="363.0" prefHeight="17.0"
98         prefWidth="150.0">
99         <font>
100             <Font name="Book Antiqua" size="12.0" />
101         </font></Label>
102     <Label fx:id="enemyKingdom" alignment="CENTER_RIGHT" layoutX="110.0" layoutY="492.0"
103         prefHeight="17.0" prefWidth="150.0">
104         <font>
105             <Font name="Book Antiqua" size="12.0" />
```

```

97     </font></Label>
98     <ComboBox fx:id="kingdomInput" layoutX="90.0" layoutY="189.0" onAction="#setKingdom"
99         prefWidth="150.0" promptText="Reino" />
100     <Label layoutX="143.0" layoutY="165.0" text="REINO">
101         <font>
102             <Font name="Book Antiqua" size="14.0" />
103         </font>
104     </Label>
105     <Pane fx:id="settingsPane" prefHeight="116.0" prefWidth="325.0"
106         style="-fx-background-color: white;" visible="false">
107         <children>
108             <Label layoutX="93.0" layoutY="14.0" text="CONFIGURACIONES">
109                 <font>
110                     <Font name="Book Antiqua" size="14.0" />
111                 </font>
112             </Label>
113             <Label layoutX="133.0" layoutY="41.0" text="Resolución">
114                 <font>
115                     <Font name="Book Antiqua" size="12.0" />
116                 </font></Label>
117             <ComboBox fx:id="resolutionInput" layoutX="88.0" layoutY="63.0"
118                 onAction="#setResolution" prefWidth="150.0" promptText="Resolución" />
119         </children>
120     </Pane>
121     <Pane layoutX="295.0" layoutY="7.0" onMouseClicked="#toggleSettings" prefHeight="25.0"
122         prefWidth="25.0">
123         <children>
124             <ImageView fitHeight="25.0" fitWidth="25.0" layoutX="-1.0" pickOnBounds="true"
125                 preserveRatio="true">
126                 <image>
127                     <Image url="@../img/settings.png" />
128                 </image>
129             </ImageView>
130         </children>
131     </Pane>
132     <TitledPane fx:id="messagePane" animated="false" collapsible="false" layoutX="25.0"
133         layoutY="231.0" prefHeight="139.0" prefWidth="279.0" text="Mensaje" visible="false">
134         <content>
135             <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="180.0" prefWidth="200.0">
136                 <children>
137                     <TextArea fx:id="messageOutput" layoutX="-1.0" prefHeight="81.0"
138                         prefWidth="279.0" wrapText="true">
139                         <font>
140                             <Font name="Book Antiqua" size="14.0" />
141                         </font></TextArea>
142                     <Button layoutX="122.0" layoutY="84.0" mnemonicParsing="false"
143                         onAction="#closeMessage" text="OK">
144                         <font>
145                             <Font name="Book Antiqua" size="12.0" />
146                         </font></Button>
147                 </children>
148             </AnchorPane>
149         </content>
150         <font>
151             <Font name="Book Antiqua" size="12.0" />
152         </font>

```

```
145 </TitledPane>
146 </children>
147 </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         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     public int getI() {
42         return i;
43     }
44
45     public int getJ() {
46         return j;
47     }
48
49     public String getType() {
50         return type;
51     }
52
53     public boolean isConnected(Tile other, int distance) {
54         return Math.abs(other.getI() - i) <= distance && Math.abs(other.getJ() - j) <=
```

```
        distance;
56     }
57
58     private ImageView generateImageView(double size, String type) {
59         Image image = new Image(String.format("img/tile_%s.png", type));
60         ImageView imageView = new ImageView(image);
61         imageView.setFitWidth(size);
62         imageView.setFitHeight(size);
63         images.put(type, imageView);
64         return imageView;
65     }
66
67     private Label generateHealthLabel(double size) {
68         Label label = new Label();
69         label.setLayoutX(size / 18);
70         label.setLayoutY(size / 18);
71         label.setTextFill(BACKGROUND_COLOR.getColor());
72         return label;
73     }
74
75     public String getKey() {
76         return i + "," + j;
77     }
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 }
```

```

18     }
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 }
```

- 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/1ZT1aK9tdZdqz731l37N5dQ56h89uFjGI/view?usp=sharing>

7. Diagrama UML

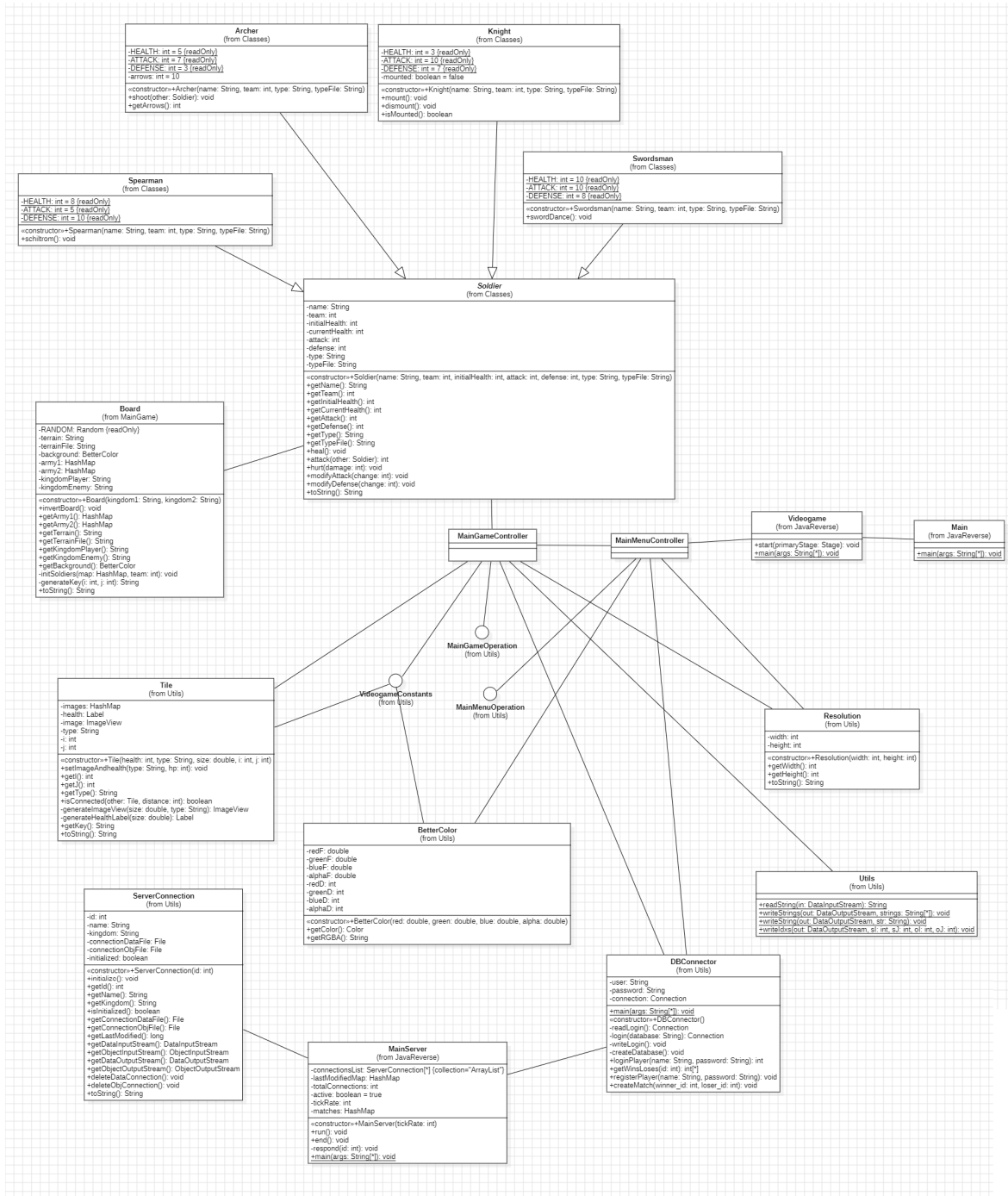


Diagrama UML.

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  
|       |--- 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 fácil 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 cumple 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 ítems.
- 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	4	
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		18.5	

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/>