Tabla de Contenido

1. Introducción **1**
- 1.1. Motivación . . . 1
- 1.2. Estado del Arte . . . 3
  - 1.2.1. Herramientas integradas en motores . . . 3
  - 1.2.2. Herramientas autónomas . . . 4
- 1.3. Objetivos . . . 5
  - 1.3.1. Objetivo General . . . 5
  - 1.3.2. Objetivos Específicos . . . 5
- 1.4. Estructura de la memoria . . . 6

2. Antecedentes **7**
- 2.1. Proyección isométrica en videojuegos . . . 7
- 2.2. Desarrollo en Unity . . . 8
  - 2.2.1. GameObjects . . . 8
  - 2.2.2. Orientación a componentes . . . 9
  - 2.2.3. Prefabs . . . 10
  - 2.2.4. Espacios de coordenadas en Unity . . . 11

3. Análisis y Diseño **12**
- 3.1. Casos de uso . . . 12
- 3.2. Composición de los Niveles . . . 13
- 3.3. Arquitectura General . . . 13

4. Implementación **15**
- 4.1. Modelamiento de Niveles . . . 15
- 4.2. Posicionamiento de baldosas . . . 15
- 4.3. Observador de sprites . . . 16
- 4.4. Controlador de nivel . . . 17
- 4.5. Carga de recursos . . . 18
- 4.6. Cámara *Pixel Perfect* . . . 18
- 4.7. Serialización . . . 20
- 4.8. Rotación . . . 20
- 4.9. Posicionamiento de murallas . . . 21
- 4.10. Sprites de bordes de murallas . . . 22
- 4.11. Recorte de murallas . . . 24
- 4.12. Rotación de murallas . . . 25

4.13. Posicionamiento de ítems . . . 26
4.14. Observador de sprites de ítems . . . 26
4.15. Editor de niveles . . . 27
4.15.1. Modificaciones al modelo . . . 27
4.15.2. Modo de uso . . . 28
4.15.3. Baldosas fantasma . . . 28
4.15.4. Construcción de murallas . . . 30
4.16. Personajes . . . 30
4.17. Callbacks . . . 32

5. Prueba de concepto: Sokoban **33**
5.1. Murallas . . . 34
5.2. Movimiento del Jugador . . . 34
5.3. Movimiento de cajas . . . 34
5.4. Carga de Niveles . . . 35
5.5. Condiciones de Victoria y Derrota . . . 35
5.6. Interfaz del Juego . . . 35
5.7. Dificultades Específicas de Android . . . 36
5.8. Conclusiones . . . 36

6. Optimizaciones para Unity **38**
6.1. Serializadores como componentes . . . 38
6.1.1. Disminución de las variaciones de niveles . . . 38
6.1.2. Exposición de propiedades en el editor de Unity . . . 39
6.1.3. Solución implementada: Serializadores que heredan de MonoBehaviour 39
6.2. Transformaciones dependientes del nivel . . . 40
6.2.1. Existencia de un único nivel . . . 40
6.2.2. Uso de coordenadas locales . . . 40
6.2.3. Poca transparencia en el uso de rotaciones . . . 41
6.2.4. Solución implementada: transformador instanciable . . . 41
6.3. Componente de posicionamiento isométrico . . . 42
6.3.1. Funcionamiento independiente de otros componentes . . . 42
6.3.2. Mantener un registro de la posición isométrica del GameObject . . . 42
6.3.3. Serialización de posición y tipo . . . 42
6.3.4. Solución implementada: Componente de posicionamiento isométrico . 43
6.4. Componente de selección de sprites . . . 43
6.4.1. Agrupación de sprites pertenecientes al mismo elemento . . . 44
6.4.2. Opcionalidad de la funcionalidad . . . 44
6.4.3. Solución implementada: Componente de selección de sprites . . . 45
6.5. MonoBehaviour con callbacks . . . 46
6.5.1. Solución implementada: IsoMonoBehaviour . . . 46
6.6. Observadores de sprites como componentes . . . 47
6.6.1. Escenarios independientes del nivel . . . 47
6.6.2. Observadores personalizados . . . 48
6.6.3. Solución implementada: Observadores como componentes independientes 48
6.7. Selección de componentes basales . . . 49

6.7.1. Utilización de Prefabs como base para los elementos creados con las herramientas . . . 49
6.7.2. Versatilidad de la utilización de Prefabs Basales . . . 49
6.7.3. Solución Implementada: Carga de Prefabs en tiempo de ejecución . . . 50
6.8. Modificación de Prefabs individuales . . . 51
6.8.1. Encapsulación de diferencias entre sprites y Prefabs personalizados . . . 51
6.8.2. Incorporación de GameObjects de manera transversal a las herramientas 52
6.8.3. Solución implementada: Cargador de recursos con soporte para Prefabs 52
6.9. Almacenamiento de variaciones de murallas en un caché . . . 52
6.9.1. Impacto en el uso de memoria . . . 53
6.9.2. Solución implementada: Diccionario de variaciones de murallas . . . 54
6.10. Búsqueda de caminos . . . 54
6.10.1. Algoritmo de búsqueda A* . . . 54
6.10.2. Componente con información del terreno . . . 55
6.10.3. Solución implementada: Extensión de búsqueda de caminos . . . 55
6.10.4. Eliminación de callbacks desde el modelo . . . 56

7. Validación **58**
7.1. Juego de estrategia por turnos . . . 58
7.1.1. Diseño del juego de ajedrez . . . 59
7.1.2. Modelo del juego de ajedrez . . . 60
7.1.3. Visualización del juego de ajedrez . . . 61
7.1.4. Control de piezas de ajedrez . . . 62
7.2. Juego de Construcción de Niveles . . . 62
7.2.1. Prototipo a construir . . . 63
7.2.2. Modelo del prototipo de construcción de ciudades . . . 64
7.2.3. Prefabs de edificios . . . 65
7.2.4. Construcción del nivel . . . 66
7.3. Movimiento complejo de personajes . . . 67
7.3.1. Diseño del prototipo . . . 68
7.3.2. Creación de nivel . . . 68
7.3.3. Animación del personaje . . . 69
7.3.4. Personalización del personaje . . . 70
7.3.5. Movimiento del personaje . . . 71
7.4. Juego en línea . . . 73
7.4.1. Diseño del prototipo . . . 73
7.4.2. Persistencia del nivel . . . 74
7.4.3. Eventos en tiempo real . . . 76
7.4.4. Conversaciones en tiempo real . . . 77

Conclusión **78**

Bibliografía **79**