
Contents

Preface	1
Who this book is for	1
About the author	2
1 Design a large application	3
1.1 Objectives	3
1.2 Software design: a considerable time savings	4
1.2.1 A simple example	4
1.2.2 Divide and conquer	9
1.2.3 An example of improvement	11
1.2.4 Model-View-Controller Approach (MVC)	17
1.3 Preparation	18
1.3.1 Choice of the game	18
1.3.2 Development environment	22
1.3.3 Example projects	27
2 Represent the state of the game	29
2.1 Data representation	29
2.1.1 Initial design	29
2.1.2 Types of information	30
2.1.3 Example with the game Pacman	30
2.1.4 Video Game Development: Specifications	32
2.2 Basic information	32
2.2.1 Classes	32
2.2.2 Hierarchy of classes (Polymorphism)	39
2.2.3 Combination of properties (Composition)	45

2.2.4	Exercices	49
2.2.5	Video Game Development: Element classes	50
2.3	Containers	50
2.3.1	Lists and associative arrays	50
2.3.2	Store sparse elements (Decorator Pattern)	57
2.3.3	Graphs of elements	60
2.3.4	Multidimensional arrays	61
2.3.5	Iterating containers (Iterator Pattern)	64
2.3.6	Main container	68
2.3.7	Exercices	69
2.3.8	Video Game Development: Containers	70
2.4	Unit tests	71
2.4.1	Implement unit tests	71
2.4.2	Exercices	78
2.4.3	Video game development: state test	80
2.5	Exercices solutions	81
2.5.1	Exercise 2.4.1: Role-playing game	81
2.5.2	Exercise 2.4.2: Roleplay with multiple classes	82
2.5.3	Exercise 3.6.1: Civilization	83
2.5.4	Exercise 3.6.2: Stellaris	85
2.5.5	Exercise 4.2.1: Civilization Test	86
2.5.6	Exercise 4.2.2: Stellaris Test	87
3	User interface	89
3.1	2D User interface with AWT	89
3.1.1	Synchronous display with double buffering	89
3.1.2	Display with tiles	99
3.1.3	Controls	113
3.1.4	Optimizing the display	120
3.2	3D User Interface with LWJGL	125
3.2.1	Creating a window	125
3.2.2	Show a triangle	129
3.2.3	Color and index	137
3.2.4	Perspective and controls	142
3.2.5	Transformation and animations	149
3.2.6	Textures	151
3.3	Software design	155
3.3.1	Abstraction of the user interface	155
3.3.2	Game modes	173
3.3.3	Exercices	191
3.4	Video Game Development: User Interface	194
3.5	Exercise Solutions	195
3.5.1	Exercise 3.3.1: Display text with the facade	195
3.5.2	Exercise 3.3.2: Display an image with the facade	197
3.5.3	Exercise 3.3.3: Detecting key sequences	199

4	Rules engine	203
4.1	General approach	203
4.1.1	Presentation	203
4.1.2	Motivation	204
4.2	Synchronization between state and user interface	206
4.2.1	Render a view of the game state	209
4.2.2	Displacement and Animations	218
4.2.3	Separate state and rendering	228
4.2.4	Exercices	237
4.2.5	Video game development: state rendering	238
4.3	Game rules	239
4.3.1	Define the rules of the game	239
4.3.2	Example of rules definitions	240
4.3.3	Apply the rules (Command Pattern)	242
4.3.4	Do and undo	253
4.3.5	Test the rules engine	265
4.3.6	Exercises	273
4.3.7	Video game development: rules of the game	273
4.4	Additional features	275
4.4.1	Game settings (Builder Pattern)	275
4.4.2	Load a level (Visitor Pattern)	278
4.4.3	Load/Save Game Status (Proxy Pattern)	283
4.5	Exercise Solutions	291
4.5.1	Exercise 2.4.1: Build a galaxy	291
4.5.2	Exercise 2.4.2: Move an army	294
4.5.3	Exercise 3.6.1: Several players on the same keyboard	296
4.5.4	Exercise 3.6.2: Checkers game with the command pattern	296
5	Artificial Intelligence	299
5.1	Preparation	299
5.1.1	Immutable game state	299
5.1.2	Artificial Intelligence Interface	304
5.1.3	Exercises	309
5.1.4	Video Game Development: AI Preparation	309
5.2	Artificial Intelligence without planning	311
5.2.1	Simple heuristics	311
5.2.2	Distance maps	313
5.2.3	Management of distance maps computations	323
5.2.4	Behavior AI	329
5.2.5	Exercises	333
5.2.6	Video game development: AI without planning	333
5.3	Artificial Intelligence with planning	335
5.3.1	Browse future states	335
5.3.2	Minimax	343
5.3.3	Other usual approaches in Artificial Intelligence	362

5.3.4	Exercises	363
5.3.5	Video game development: AI with planning	364
5.4	Exercises solutions	365
5.4.1	Exercise 1.3.1: Immutable galaxy	365
5.4.2	Exercise 1.3.2: Connect 4 with a simple AI	367
5.4.3	Exercise 2.5.1: Shortest path in the galaxy	369
5.4.4	Exercise 2.5.2: Connect 4 with a heuristic AI	370
5.4.5	Exercise 3.4.1: Traversal with the Node interface	372
5.4.6	Exercise 3.4.2: Connect 4 with an AI with planning	375
6	Concurrent execution and networking	377
6.1	Concurrent execution	377
6.1.1	Separate rules engine and user interface	377
6.1.2	Parallelize processing	388
6.1.3	Exercises	406
6.1.4	Video game development: concurrent execution	408
6.2	Network communication	411
6.2.1	Essentials	411
6.2.2	Implement Web Services	427
6.2.3	Exercises	440
6.3	Network Gaming	443
6.3.1	Principles	443
6.3.2	Serialize commands	444
6.3.3	Gather the players	448
6.3.4	Multiplayer game	461
6.3.5	Exercises	471
6.3.6	Video game development: Network gaming	475
6.4	Exercise solutions	478
6.4.1	Exercise 1.3.1: Parallelize the exhaustive search for collisions	478
6.4.2	Exercise 1.3.2: Parallelize the indexed search for collisions	480
6.4.3	Exercise 2.3.1: Creating a chat with sockets	482
6.4.4	Exercise 2.3.2: Implementing a nano HTTP server	484
6.4.5	Exercise 3.5.1: Multiplayer with UDP	493
6.4.6	Exercise 3.5.2: Local Stateless Multiplayer	498