

Contents

1	Introduction	1
1.1	Target Audience	1
1.1.1	New to Programming	2
1.1.2	New to Godot Engine	4
1.2	Why an Inventory System?	4
1.3	How to Use This Guide	5
1.3.1	The Inventory System Project files	6
1.3.2	Source Code Formatting	8
1.3.3	Diagrams	9
1.4	What We'll Cover	9
1.4.1	Data Model	9
1.4.2	Inventory Component	10
1.4.3	User Interfaces (Browse and Transfer)	10
1.4.4	Quality-of-Life Features	10
1.4.5	Game Integration with Examples	11
1.4.6	Granular Character-level Permissions	11
1.4.7	Networked Multiplayer Capabilities	11
1.4.8	Crafting System	11
1.4.9	Music Sequencing	11
1.5	Active Development and Future Releases	12
2	High-level Conceptual Overview	13
2.1	Item Management	13
2.2	High-level Components and Parts Breakdown	14
2.3	The Inventory Component	15
2.3.1	Player Character Integration	16
2.4	Inventory Items	16
2.4.1	Items in the Game World	17
2.5	Inventory User Interface	18
2.6	Configuration and Customization	19
2.7	Game Integration	20
2.7.1	Interactions	20
2.7.2	Permissions	21

2.7.3	Game UI	22
2.8	A Full Player/Object Interaction Flow Example	23
3	Modeling Inventory Items	27
3.1	Concepts	27
3.1.1	A Good Data Model	27
3.1.2	Data Modeling with Resources	28
3.1.3	Resource Classes vs Resource Objects	28
3.2	Inline Resources vs Separately Stored Resources	29
3.3	Modeling the Item	29
3.3.1	Modeling Relationships	30
3.3.2	Collections	31
3.3.3	Resources: Shared or Unique	32
3.4	Customizing the Data Model for Your Game	33
3.4.1	Creating Item Types	34
3.4.2	Creating Items	35
4	The Inventory Component	37
4.1	Concepts	38
4.1.1	Item Repository	38
4.1.2	API	38
4.1.3	Observability	38
4.2	Managing Contents	39
4.3	Limiting Inventory Size	40
4.4	CRUD Functionality	41
4.4.1	Typed vs Untyped Arrays	44
4.5	Arbitrary Item Placement	45
4.6	Composable Architecture	49
5	The Inventory UI	51
5.1	Concepts	51
5.1.1	Design	51
5.1.2	Containers and Slots	52
5.1.3	The Model-View-Controller Pattern	54
5.1.4	Inventory Controller	55
5.1.5	Managed Slots vs Assigned Slots	56
5.1.6	Fixed-column Grid View	57
5.1.7	Responsive Grid View	57
5.1.8	List View	58
5.1.9	Custom UIs	59
5.1.10	Toolbar	59

5.2	The Inventory UI Scene	59
5.2.1	The Inventory UI Shell	59
5.2.2	The Grid View Setup	59
5.2.3	The Controller Setup	61
5.2.3.1	The Abstract Controller	61
5.2.3.2	The Managed-Slots Controller	61
5.2.3.3	The Assigned-Slots Controller	62
5.2.3.4	Adding The Controller To The Inventory UI	62
5.2.4	Managing And Displaying Slots	63
5.2.4.1	The Slot Scene	63
5.2.4.2	Adding And Removing Slots	65
5.2.4.3	Retrieving Slots By ID	67
5.2.4.4	Reacting to Inventory and Item Changes	69
5.2.4.5	Switching the Slot Scene at Runtime	72
5.2.5	Adding List View and Responsive Grid View	72
5.2.5.1	Adding a List View	72
5.2.5.2	Assembling the List Item Scene	72
5.2.5.3	Switching Between Multiple Views	73
5.3	The Inventory UI Demo	76
5.3.1	Two Inventories, Four Dialogs	76
5.3.2	Test Adding Items	77
5.3.3	Test Adding and Removing Inventory Slots	79
5.3.4	Handling Overflow: Dropping Items	79
5.4	User Input Handling in the Controller	80
5.4.1	Listening to Slot Signals from the Controller	80
5.4.2	Exposing Slot Signals to the Inventory UI	83
5.5	Exposing Controller Functionality to the inventory UI	84
5.6	The Inventory UI Toolbar	85
5.6.1	Switching Item Presentation from the Toolbar	86
5.7	Custom Inventory UI Examples	88
5.7.1	A Simple Inventory UI	88
5.7.2	Creating a Tabbed Inventory UI	88
6	Basic Inventory Actions	91
6.1	Concepts	91
6.1.1	Using Items	91
6.1.2	Dropping Items	93
6.1.3	Action Strategy	93
6.2	Extending the Controller With an Action Strategy	94
6.2.1	The Action Strategy Interface	94

6.2.2	Delegating Input Events to the Strategy	95
6.2.3	Delegating Mouse Signals to the Strategy	95
6.2.4	Delegating Focus Signals to the Strategy	96
6.3	Implementing an Action Strategy	96
6.3.1	The Default Action Strategy	97
6.3.2	Using Items	97
6.3.3	Dropping Items	100
6.3.4	Handling Item Focus	102
6.3.5	Slot Focus Follows the Mouse	102
6.4	Action Strategy Configuration from the Inventory UI	103
6.5	Testing Using and Dropping Items	104
7	Inventory Saving and Loading	105
7.1	Concepts	105
7.1.1	Persistence	106
7.1.2	Serialization and Deserialization	106
7.2	Serialization Strategy	106
7.3	Extending the Serialization Strategy	109
7.4	Handling Backward Compatibility in Save Games	110
7.5	Testing Saving and Loading	110
7.6	Dynamic Serialization Strategy	113
8	Inventory Theming	117
8.1	Concepts	117
8.1.1	Theming Visuals	117
8.1.2	Theming Audio	117
8.2	Implementing Godot Themes	118
8.2.1	The Unstyled Inventory UI	118
8.2.2	Variants and Naming Conventions	119
8.2.3	Creating a Theme	119
8.3	Implementing Audio Themes	120
8.3.1	Defining an Audio Theme Resource Class	120
8.3.2	Defining an Audio Theme for Inventories	120
8.3.3	The Audio Theme Handler	121
8.3.4	Creating the Audio Theme Binding Class	122
8.3.5	Creating Audio Bindings for the Inventory UI	123
8.3.6	Adding the Audio Theme Handler to the Inventory UI	125
9	Item Expirations	127
9.1	Concepts	128
9.1.1	Item Expiration Support	128

9.1.2	Inventory Support	128
9.1.3	Scalability	128
9.2	Implementing Item Expiration	129
9.2.1	Adding Expiration to the Data Model	129
9.2.2	Adding Expiration to the Inventory Component	130
9.3	Creating an Expiration Extension	130
10	Advanced Inventory Actions	135
10.1	Concepts	135
10.1.1	Sorting	135
10.1.2	Stacking	136
10.1.3	Searching / Filtering	136
10.2	Sorting Inventory Items	137
10.2.1	The Sorting Strategy	137
10.2.2	Implementing Sorting in the Inventory	138
10.2.3	Custom Sorting Strategy	138
10.2.4	Sorting Through the Inventory Toolbar	139
10.3	Stacking and Splitting Inventory Items	141
10.3.1	Theory	142
10.3.2	The Stacking Strategy Interface	142
10.3.3	The Default Stacking Strategy	144
10.3.4	Adding Stacking to the Inventory Class	147
10.3.5	Adding Stacking to the Toolbar Button	147
10.3.6	Testing Stacking With the Inventory Demo	148
10.3.7	Automatic Stacking when Inventory Items are Added	148
10.4	Searching Items	150
10.4.1	Implementing the Filter Strategy	150
10.4.2	Adding Filtering to the Controller	151
10.4.3	Updating the Inventory UI Component	152
10.4.4	Localization Considerations	153
11	Inventory Transfers and Drag-and-Drop	155
11.1	Concepts	156
11.1.1	Transferring items	156
11.1.2	Drag-and-Drop	157
11.1.3	The Click-and-Drag Approach	157
11.1.4	The Click-and-Release Approach	157
11.1.5	Picking up Arbitrary Quantities With Modifier Keys	158
11.1.6	Drag-and-Drop Theory	158
11.1.7	Use Cases and Player Expectations	159

11.1.8	Leveraging the Dragon Drop Addon	160
11.2	Implementing Inventory Transfers	162
11.2.1	The Inventory Transfer UI	163
11.2.2	Adding Item Transfer to the Action Strategy	164
11.2.3	Adding Transfer Logic to the Controller	166
11.2.4	Adding Remote Inventories to the Inventory UI	166
11.2.5	Adding Transfer Capabilities to the Inventory Component	168
11.2.6	The Transfer Strategy Interface	169
11.2.7	A Simple Transfer Strategy	170
11.2.8	The Stacking Transfer Strategy	171
11.3	Implementing Drag-and-Drop Support	172
11.3.1	Godot's Drag-and-Drop Building Blocks	172
11.3.1.1	Initiating Drag	173
11.3.1.2	Drag Preview	173
11.3.1.3	Indicating Whether Dropping is Allowed	174
11.3.1.4	Ending Drag-and-Drop: The Drop	175
11.3.1.5	Forcing Drag-and-Drop	175
11.3.1.6	Canceling Drag-and-Drop	176
11.3.1.7	Handling Failed Drag-and-Drop	176
11.3.1.8	Using composition with Drag-and-Drop forwarding	176
11.3.2	The Dragon Drop Addon	177
11.3.2.1	The Dragon Drop Component	177
11.3.2.2	The Dragon Drop Helper (singleton)	180
11.3.2.3	The Dragon Drop Data	181
11.3.2.4	The Dragon Drop Strategy	182
11.3.3	Integrating and Extending Dragon Drop	189
11.3.3.1	Extending the Dragon Drop Drag Data	189
11.3.3.2	Integrating Dragon Drop into the Inventory Controller	190
11.3.3.3	Extending the Dragon Drop Preview Scene	191
11.3.3.4	Extending the Dragon Drop Strategy	192
11.3.3.5	Setting the Drag-and-Drop Strategy From the Inventory UI	196
11.3.3.6	Creating Drag-and-Drop Configuration Presets	197
12	Inventory UX Refinements	199
12.1	Inventory Slot Popovers	199
12.1.1	Concepts	199
12.1.1.1	The Popover UI	200
12.1.1.2	The Popover Component	200
12.1.1.3	The Popover Settings	201

12.1.2	The Inventory Popover UI Scene	201
12.1.2.1	Ignoring Popover Mouse Events	202
12.1.2.2	Handling Varying Sizes of Popovers	203
12.1.3	The Inventory Popover Scene	203
12.1.3.1	Listening to Mouse Signals	204
12.1.3.2	Configuring the Popover UI Scene	204
12.1.3.3	Following the Mouse	206
12.1.3.4	Reacting to Hover Events	206
12.1.3.5	Inventory Popover Integration	209
12.1.4	The Inventory Popover Settings Resource	209
12.1.5	Custom Popover UIs	211
12.1.5.1	The Custom MyInventoryPopoverUI Scene	211
12.1.5.2	The Custom MyAdvancedPopoverUI Scene	213
12.2	Refining Controls	214
12.2.1	Focus Stuck in the Search Box	214
12.2.2	Responsive Design: Toolbar Button Labels	216
13	Character Interactions	219
13.1	Concepts	219
13.1.1	Interactor and Interactable	220
13.1.2	Stateless Interactions	220
13.1.3	Stateful Interactions	220
13.2	The Interaction Demo	221
13.2.1	The Player Character	221
13.2.2	The Lamp Scene	222
13.3	Interaction Components: Interactor and Interactable	223
13.3.1	The Interactor Component	224
13.3.1.1	Initiating an Interaction	225
13.3.2	The Interactable Component	226
13.3.2.1	Handling Complexity	228
13.3.2.2	The Interactable Strategy	229
13.3.2.3	The Interactable Toggle Strategy	230
13.4	The Inventory Interaction Demo	230
13.4.1	The Box Scene	231
13.4.2	The Inventory Screen Scene	231
13.4.3	The Interactable Inventory Screen Strategy	233
13.4.4	Stateful Inventory Screen Interactions	233

14 Inventory Permissions	237
14.1 Concepts	237
14.1.1 Authentication vs Authorization	238
14.1.2 The Character: Player Actions by Proxy	239
14.1.3 The Access Manager	240
14.1.4 The Access Policy	240
14.1.5 Access Contexts	240
14.1.6 The Inventory Access Policy	241
14.1.7 Custom Component Policies	241
14.1.8 Accessing Inventories	242
14.2 The Access Manager Component	242
14.3 The Access Policy	245
14.3.1 Default Access Policy	245
14.3.2 Open Access Policy	245
14.4 Context-aware Inventories	245
14.4.1 The Inventory Access Policy	246
14.4.2 Simple Inventory Access Policy	248
14.4.3 Using Inventory Items	249
14.4.4 Dropping Inventory Items	250
14.5 Interaction Demo With Permissions	251
14.5.1 The World Scene	252
14.5.2 Team-aware Character	252
14.5.3 Team-selector Object	253
14.5.3.1 Team-selector Interaction Strategy	254
14.5.4 Team-aware Storage Box Scene	255
14.5.4.1 Storage Box Team Access Policy	255
14.5.4.2 The Interactable Inventory Strategy	256
14.5.5 Handling Interaction Authorization	258
14.5.6 The Inventory Screen With Permissions Scene	259
14.5.7 Context IDs and Drag-and-drop Access Inventories	260
15 Game UI Integration	261
15.1 Concepts	261
15.1.1 Game UI	261
15.1.2 Game and UI Separation	262
15.1.3 Different types of Game UIs	262
15.1.4 Single-dialog Interface	264
15.1.5 Multi-dialog Interface	265
15.2 The GameUI as a Singleton	265
15.2.1 Via Singleton Pattern	265

15.2.2	The Downside of Singletons	266
15.2.3	Node Groups as Alternatives to Autoload Singletons	266
15.3	The Dialog Context Helper	267
15.4	Assembling the Single-Dialog Interface Game UI	273
15.4.1	The Inventory Browse Screen	274
15.4.2	The In-Game Menu Screen	276
15.4.3	The Inventory Transfer Screen	277
15.4.4	The SDIGameUI Scene	278
15.5	Multi-dialog Research	280
15.5.1	The Resizable Inventory Demo	280
15.5.1.1	Resizable Inventory Scene	281
15.5.1.2	Moving and Resizing with GGResizableControl	282
15.5.1.3	The ResizableInventory Demo Scene	284
15.5.1.4	The Inventory Button	287
15.5.1.5	Running the Dynamic Inventory Demo	290
15.5.1.6	Shortcomings	290
15.5.2	The Resizable Container Demo	292
15.5.2.1	The Challenge With Nesting Scenes	292
15.5.2.2	Relocating Nested Content With GGSceneRelocator	294
15.5.2.3	Adding Additional Resizable Scenes	295
15.5.2.4	Focus Handling	296
15.5.2.5	Remembering and Restoring Focus per Window	297
15.5.2.6	Focus Switching/cycling With a Controller	300
15.5.3	The Window Manager Demo	302
15.5.3.1	The UI Manager	303
15.5.3.2	The UI Container	305
15.5.3.3	The Inventory Window Manager	307
15.6	Assembling the Multi-Dialog Interface Game UI	308
15.7	The Game UI Demo	311
15.7.1	The GameUI Selector	312
15.8	The GameUI Interaction Demo	313
15.9	Notifications	314
16	Multiplayer Inventory	315
16.1	Prerequisites	315
16.2	Concepts	316
16.2.1	Godot High-level Multiplayer	316
16.2.2	Node Class Multiplayer Integration	316
16.2.3	Replicating State Across the Network	319
16.2.4	Remote Procedure Calls (RPCs)	319

16.2.5	MultiplayerSpawner and MultiplayerSynchronizer	320
16.2.6	When to Use Spawners/Synchronizer vs RPCs	320
16.2.7	RPC Patterns	321
16.2.7.1	Client to Server	321
16.2.7.2	Server to Client	322
16.2.7.3	Unified RPC Invocation	322
16.2.8	Retrofitting Single-player Games With RPCs	323
16.2.9	Hiding Inventory Contents from Clients by Default	324
16.2.10	Synchronizing Inventory Contents	324
16.3	Multiplayer Support For the Access Manager	324
16.3.1	Opening a Context from the Client	325
16.3.2	AccessManager Client/Server Sequence Diagram	329
16.3.3	Closing a Context from the Client	330
16.4	Multiplayer Capabilities for the Inventory	333
16.4.1	Listening to Access Manager Context Signals	333
16.4.2	Sending Inventory Contents to Clients	334
16.4.3	Propagating Inventory Signals to Clients	335
16.4.4	The Multiplayer Interface for Using Items	337
16.4.5	The Multiplayer Interface for Dropping Items	338
16.4.6	The Multiplayer Interface for Sorting, Stacking/splitting, and Transferring Items	339
16.5	The Inventory Multiplayer Demo	340
16.5.1	The WebSockets Demo Scene	341
16.5.2	The Main Menu	342
16.5.3	Abstracting Network Handling: The NetworkManager	342
16.5.4	The Multiplayer Client	345
16.5.5	Spawning Clients when a Player Connects	346
16.5.6	Managing Dialogs and Contexts	347
16.5.7	Opening an Inventory (Browse View)	348
16.5.8	Opening Inventories for Transfer	351
16.5.9	Toggling Inventory Between Public/Private	352
16.5.10	Closing All Inventories	356
16.5.11	Adjusting the Number of Item Slots	356
16.5.12	Inventory Access Policy	356
17	Multiplayer Interaction Demo	359
17.1	Concepts	360
17.1.1	Multiplayer Authority, and When to Trust the Client	360
17.1.2	Handling Player Input and Movement	361
17.1.2.1	Giving the Client Authority	361
17.1.2.2	Only Sending Inputs	361

17.1.2.3	Client-side Prediction and Server Reconciliation	362
17.1.3	Client-side Authority and Inventories	362
17.1.4	Multiplayer Visibility	363
17.2	The Multiplayer Interaction Demo Scene	364
17.3	The Multiplayer World Scene	365
17.3.1	Player Spawn Points for the TileMapLayer	366
17.3.2	The MultiplayerSpawner Setup	366
17.4	The Multiplayer Character Scene	367
17.4.1	The Simple Version	368
17.4.1.1	Character Sync and Data Flow	369
17.4.1.2	Sending Client Inputs to the Server	370
17.4.1.3	Trusting the Client for Movement Only	370
17.4.2	The Netfox Version	372
17.4.2.1	Gathering Input	372
17.4.2.2	Applying Movement	373
17.5	The Multiplayer Storage Box	373
17.5.1	Open/closed Storage Box	373
17.5.2	Close Contexts when Leaving Proximity	375
17.6	The Multiplayer Loot Bag	375
17.6.1	Expiration	376
17.6.2	Preventing Players from Adding Items to Loot Bag	377
17.6.3	Automatically Destroying Empty Loot Bags	377
17.7	Spawning Scenes	377
17.7.1	Spawning Bags for Dropped Items	379
17.7.2	Using the Auto Spawn List	380
17.7.3	Using Custom Spawn Functions	381
17.7.4	Keep Spawning Loot	382
17.8	Saving and Loading World State	383
17.8.1	Save and Load Mechanism	384
17.8.2	Saving at Regular Intervals	385
17.8.3	Loading when the Game Starts	385
17.8.4	Saving on Exit	386
17.9	Limiting Multiplayer Visibility	386
17.9.1	Network Client Scope	387
17.9.2	Network Objects	389
17.9.3	Network Object Strategies	389
17.9.4	The Visibility Synchronizer	391
17.9.5	Broadcasting Character Actions to Visible Peers	393

18 Multiplayer Matchmaking and Miscellaneous	397
18.1 There's No Place Like 127.0.0.1	397
18.2 Hosting a Server	397
18.3 Firewalls and NAT	397
18.4 NAT Piercing / Hole Punching	398
18.5 Dedicated Servers	398
18.6 Self-hosted STUN/TURN Solutions	398
18.7 Matchmaking: How to Find a Server to Connect To	399
18.7.1 Steamworks Integration	399
18.7.2 W4 Cloud Integration	399
18.8 Selecting the Right Network Protocol	399
18.9 Connecting Multiple Servers and Clients	400
18.10 Latency	401
18.10.1 Emulating latency (on Windows, MacOS, and Linux)	401
18.11 Additional Resources	401
18.11.1 Network-related Talks (Videos)	401
18.11.2 Quake, QuakeWorld, Quake 3 Networking articles	402
18.11.3 Starsiege: Tribes Networking Model	402
18.11.4 Terraria's Networking Protocol	402
18.11.5 Misc Networking Architecture	402
19 Equipping Items	403
19.1 Concepts	403
19.1.1 Equipment Items	403
19.1.2 Item Use Handler	403
19.1.3 The Item Equipper Component	404
19.1.4 The Equipment Holder	404
19.1.5 Storing and Restoring Equipment State	404
19.2 Implementing Equipment for a Character	404
19.2.1 Adding Equipment Items	405
19.2.2 Implementing the MyItemUseHandler Component	405
19.2.3 Implementing the MyItemEquipper Component	406
19.2.4 Implementing the MyEquipmentHolder Component	411
19.2.4.1 Connecting Equipment Controls	413
19.2.4.2 Equipment Multiplayer and Visibility Sync	415
20 Crafting Items	417
20.1 Concepts	419
20.2 The Crafting Demo	419
20.3 The Crafting Recipe Data Model	420

20.4	The Core Crafting Component	421
20.4.1	Basic Crafting	421
20.4.2	Consuming Inventory Items Efficiently	424
20.4.2.1	Fast Item Search by Type; Creating a Lookup Table	425
20.4.2.2	Implementing Consuming Items from the Inventory	432
20.4.2.3	Using the New consume_items() Method	433
20.4.2.4	Unlocking New Feature Possibilities with the Lookup Table	434
20.4.3	Continuing with Basic Crafting	434
20.4.4	Pause Crafting when Inventory Is Full	435
20.4.5	Automatic Crafting	437
20.4.6	Optimizing by Disabling the Process Function	440
20.4.7	Multiplayer Handling	441
20.4.8	Crafting Smelters, Campfires, and Crafting Tables	446
20.4.9	Future Enhancements	447

21 Inventory Weight Limits 449

21.1	Concepts	449
21.2	Adding Weight to Items	449
21.3	Adding Weight Awareness to the Inventory	450
21.4	Preventing Overloading the Inventory	451
21.4.1	The Inventory Item Filter	451
21.4.1.1	The Inventory Item Filter Strategy	451
21.4.1.2	Enhancing the Inventory Component	451
21.4.1.3	Updating Drag-and-Drop and Transfer Logic	452
21.4.1.4	The Item Filter Weight Strategy	454
21.4.1.5	The Item Filter Type Strategy	455
21.4.1.6	The Item Filter All Strategy	455

22 Building Structures 457

22.1	Concepts	458
22.1.1	Craft First, Then Place	459
22.1.2	Ad-hoc Crafting While Placing	459
22.1.3	The Placement Process	460
22.2	Defining Items to Represent Structures	461
22.2.1	The Structure Item Type	461
22.2.2	The Structure Item Data	461
22.3	Enhancing GGInventory with the Use Strategy	462
22.3.1	The MyInventoryUseBuildStrategy	463
22.3.2	The GGInventoryUseSetStrategy	465

22.4	The MyStructurePlacerUI	465
22.4.1	Start Placement Preview	466
22.4.2	Display Structure Preview	467
22.4.3	Confirming and Canceling Placement Preview	470
22.5	The MyStructurePlacer	472
22.5.1	Starting and Ending Placement	472
22.5.2	Placing the Structure	473
22.5.2.1	Finding the Ground Node	475
22.5.2.2	Snapping Structure Placement	476
22.5.2.3	Determining Whether Placement is Valid	477
22.6	Configuring the Character Scene for Structure Placing	480
22.7	Structure Placement UI in the Multiplayer Interaction Demo	480
23	Music Sequencer (Prototype)	483
23.1	Concepts	483
23.1.1	Sound Waves	483
23.1.2	Hertz (Hz)	484
23.1.3	Notes	484
23.1.4	Octaves	484
23.1.5	Semitones	484
23.1.6	Music Scales	485
23.1.7	Time Signatures	485
23.1.8	Pitching Audio Stream Players	485
23.1.9	Patterns	485
23.2	Mapping Music Concepts onto the Inventory	486
23.2.1	Notes and Instruments	486
23.2.2	Calculating the Pitch	487
23.2.3	Representing Tracks With Inventories	489
23.2.4	BPM Provider - Gimme a Beat!	491
23.2.5	The Sequencer User Interface	492
24	Architecture & Advice	493
24.1	Proof of Concept	494
24.2	Prototypes	494
24.3	Abstractions	494
24.4	Composition and Decomposition	495
24.5	Product-focused Mindset	495
24.6	Best Practices	495
24.7	Principles and Guidelines	496
24.7.1	DRY	496

24.7.2	KISS	497
24.7.3	YAGNI	497
24.7.4	SOLID	497
24.7.4.1	Single Responsibility	497
24.7.4.2	Open/Closed	498
24.7.4.3	Liskov Substitution	498
24.7.4.4	Interface Segregation	498
24.7.4.5	Dependency Inversion	498
24.7.5	GRASP	499
24.7.5.1	Information Expert	499
24.7.5.2	Creator	499
24.7.5.3	Controller	499
24.7.5.4	Low Coupling	499
24.7.5.5	High Cohesion	500
24.7.5.6	Polymorphism	500
24.7.5.7	Pure Fabrication	500
24.7.5.8	Indirection	500
24.7.5.9	Protected Variations	501
24.7.6	XY Problem	501
24.7.7	Yak Shaving	501
24.7.8	Technical Debt	502
24.7.9	Bike shedding	502
24.7.10	Sharpening the Axe	503
24.7.11	The Takeaway	503
24.8	Components	504
24.8.1	The Root Node is a Scene's API	504
24.8.2	Program to Interfaces, not Implementations.	504
24.8.3	Reusable Scenes vs Reusable Scripts	504
24.8.4	Registering Named Classes	505
24.8.5	Composition and Working With Nodes	505
24.8.6	Creating Semantic Components	506
24.9	Diegesis: Player vs Character	507
25	Appendix	509
25.1	Object-Oriented Programming Concepts	509
25.1.1	Classes and Objects	509
25.1.2	Encapsulation	509
25.1.3	Abstraction	509
25.1.4	Inheritance	510
25.1.5	Polymorphism	510

25.1.6	Interfaces vs Duck Typing	511
25.1.7	Composition	512
25.2	List of Design Patterns	513
25.2.1	Creational Patterns	514
25.2.1.1	The Singleton Pattern	514
25.2.1.2	The Factory Pattern	515
25.2.1.3	The Abstract Factory Pattern	516
25.2.1.4	The Builder Pattern	518
25.2.1.5	The Prototype Pattern	519
25.2.2	Structural Patterns	520
25.2.2.1	The Adapter Pattern	520
25.2.2.2	The Bridge Pattern	521
25.2.2.3	The Composite Pattern	523
25.2.2.4	The Decorator Pattern	524
25.2.2.5	The Façade Pattern	525
25.2.2.6	The Flyweight Pattern	526
25.2.2.7	The Proxy Pattern	528
25.2.2.8	Structural Semantics	528
25.2.3	Behavioral Patterns	529
25.2.3.1	The Chain of Responsibility Pattern	529
25.2.3.2	The Command Pattern	530
25.2.3.3	The Interpreter Pattern	531
25.2.3.4	The Iterator Pattern	532
25.2.3.5	The Mediator Pattern	533
25.2.3.6	The Memento Pattern	535
25.2.3.7	The Observer Pattern	536
25.2.3.8	The State Pattern	537
25.2.3.9	The Strategy Pattern	538
25.2.3.10	The Template Method Pattern	539
25.2.3.11	The Visitor Pattern	540