

n8 Game Framework Design Document

Nate Ebel

February 25, 2014

Contents

1	Design and Use Patterns	3
1.1	Command	3
1.2	Observer	3
1.3	Subclass Sandbox	3
1.4	Singleton	3
1.5	Service Locator	3
1.6	Components	3
2	Client Use/Interaction	3
3	Resource Management	3
4	Game Loop	3
5	Window Management	3
6	Game States	3
7	Entities	3
8	Components	3
9	Entity States	3
10	Input	3
11	Events	3

12	Rendering	3
12.1	Rendering operations	3
12.2	Static sprite	4
12.3	Animated sprite	4
13	Camera	4
14	Operations on or using entities	4
15	Game levels	4
16	Logging	4
17	System wide values and enums	4
18	Configuration data/files	4
19	Use Cases	4
19.1	Input changes game state	4
19.2	Input changes player state	4
19.3	Input moves player	5
19.4	Entity destroyed and animation is played	5
19.5	Timed event activated	5
19.6	Timed event destroyed	5
19.7	Entity collision	5
19.8	AI player must make a decision	5

1 Design and Use Patterns

1.1 Command

Object oriented replacement for callbacks that encapsulates a request within an object.

Benefits and uses include:

- Allows easy redo/undo functionality if do and undo methods are implemented and some type of history is kept
- Can be made general enough to operate on different entities such as a “jump” command that works for both user controlled and computer controlled entities
- Can become a sandbox if many common operations are included with the base class
- If command is stateless, a single instance of the command may be used for the entire program
- Can be easily mapped to new inputs

1.2	Observer
1.3	Subclass Sandbox
1.4	Singleton
1.5	Service Locator
1.6	Components
2	Client Use/Interaction
3	Resource Management
4	Game Loop
5	Window Management
6	Game States
7	Entities
8	Components
9	Entity States
10	Input
11	Events
12	Rendering
12.1	Rendering operations
	<ul style="list-style-type: none">• Render single entity• Render single entity with a camera object (Camera: see Section 13)

12.2 Static sprite

12.3 Animated sprite

13 Camera

14 Operations on or using entities

AI physics

15 Game levels

16 Logging

17 System wide values and enums

18 Configuration data/files

19 Use Cases

19.1 Input changes game state

19.2 Input changes player state

walking → flying

- 19.3 Input moves player
- 19.4 Entity destroyed and animation is played
- 19.5 Timed event activated
- 19.6 Timed event destroyed
- 19.7 Entity collision
- 19.8 AI player must make a decision
- 19.9 Notify an achievement system when 5 enemies are destroyed
- 19.10 Remap an action's key