Project Ngeen

Glossary

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 3/20/2016 | 1.0 | Initial commit | Daian Dragos |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

2. Glossary 4

Glossary

# Introduction

Here is a list of all the terms that you might encounter while using the product or reading the manual.

# Glossary

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Term | **Definition and Information** | **Format** | **Validation Rules** |
| System | Class that processess entities with specific components. | SystemName | - |
| Entity | Class containing components. | Entity | - |
| Component | Class containing user defined data or logic. | ComponentName | - |
| Asset | Data used in components(fonts, images). | Asset<T> | T has to be a valid type |
| Script | Logic contained in a ComponentScript. Usually operates on the containing entity. | ScriptName | - |
| Scene | Class defining current game phase and folder defining current game phase in use assets. | SceneName.java and data/scenes/SceneName | Assets used have to exist(in folder) |
| Command | Action done by user. Can be undo-ed or redo-ed. | - | - |
| Factory | Place that creates objects. | - | - |
| Function | Class that describes lambda type of command. | - | - |
| Relative Path | Path relative to project classpath. | ./ | - |
| Texture | Class imported from libgdx defining an image stored on the GPU | - | - |
| TextureAtlas | Class imported from libgdx defining an image with multiple smaller images(improves efficiency) | - | - |
| BitmapFont | Class imported from libgdx defining a loaded font. | - | - |
| Sound | Class imported from libgdx defining a small(in size) sound file. | - | - |
| Music | Class imported from libgdx defining a large(in size) sound file. This one is streamed. | - | - |
| ShaderProgram | Class imported from libgdx that contains a compiled OpenGL program. | - | - |
| OpenGL | Open Graphics Library. | - | - |
| OpenGL program | Compiled program that runs on the GPU. Contains vertex and fragment shader | - | - |
| GPU | Graphics Processing Unit | - | - |
| Vertex Shader | Specific shader that describes how the vertices are rendered. | - | - |
| Shader Program | Program that describes certain stages of the graphics pipeline. | - | - |
| Graphics Pipeline | Process of rendering drawable objects using the GPU. | - | - |
| Vertices | Describes object's boundaries. | - | - |
| Mesh | An array of vertices, normals and other. | - | - |
| Fragment Shader | Specific shader that describes how different fragments(color shapes) are rendered | - | - |
| Camera | Class that translated objects from world space to screen space. | - | - |
| World space | Objects that we work with are here(in 3d, at certain x,y,z location) | - | - |
| Screen space | Objects are drawn here(in 2d, on screen). They no longer have depth. | - | - |
| Base Component | Every possible component extends this. | - | - |
| Camera | Contains a camera. | - | - |
| Material | Contains information about how the mesh is drawn. | - | - |
| Mesh | Contains a mesh. | - | - |
| Point | Contains information about the object location, scale and rotation. | - | - |
| Rigid | Simulates close to real physics properties for the object(mass, size, shape, etc.) | - | - |
| Box2D | Library that solves 2d physics in game | - | - |
| Bullet | Library that solves 3d physics in game | - | - |
| LibGDX | Library that wraps OpenGL and other system calls and makes the project cross platform | - | - |
| Dynamic code replacement | Realtime code replacement, without the need of recompiling | - | - |
| Component Variable | Hold and search for any data type. | - | - |
| Component Sprite | Component made to ease up the drawing of 2d textures without knowledge of shaders. | - | - |
| Component Script | Contains a single script which can be relative to the project or not. | - | - |
| Xml load/save | Saving or loading the state of the game into an xml file. | - | - |
| Component UI Base | This extends the component base and is the base class for every UI component. | - | - |
| UI | User Interface | - | - |
| Button | A button with different callbacks and texture. | - | - |
| Button Group | Used to simulate radio groups. | - | - |
| CheckBox | Checkbox implementation for the UI. | - | - |
| Container | A simple class that can hold any UI element. Useful to bypass some limitations of some elements. | - | - |
| Dialog | A popout dialog asking the user for click input. | - | - |
| Horizontal Group | A horizontal group. Like a table but with one row only. | - | - |
| Image | Draws an image in screen space. | - | - |
| Image Button | Specific implementation of button. | - | - |
| Label | Text only UI element. | - | - |
| Layout | Base class for every UI element that can accept a widget. | - | - |
| UI List | A list (aka list box) displays textual items and highlights the currently selected item. | - | - |
| Progress Bar | A progress bar is a widget that visually displays the progress of some activity or a value within given range | - | - |
| Scroll Pane | A group that scrolls a child widget using scrollbars and/or mouse or touch dragging. | - | - |
| SelectBox | A select box (aka a drop-down list) allows a user to choose one of a number of values from a list. | - | - |
| Slider | A slider is a horizontal indicator that allows a user to set a value. | - | - |
| Split Pane | A container that contains two widgets and is divided either horizontally or vertically. | - | - |
| Stack | A stack is a container that sizes its children to its size and positions them at 0,0 on top of each other. | - | - |
| Stage | Other actors are placed here. | - | - |
| Actor | Actual objects held by all UI elements. | - | - |
| Table | A group that sizes and positions children using table constraints | - | - |
| Text Area | Text only UI element with multiple lines. | - | - |
| Text Button | Extended button that allows text. | - | - |
| TextField | Text UI element that accepts user input. | - | - |
| Touchpad | An on-screen joystick. | - | - |
| Tree | A tree widget where each node has an icon, actor, and child nodes. | - | - |
| Vertical Group | Group of actors with a single collumn. | - | - |
| Widget | Cannot contain other actors but itself. | - | - |
| Window | A table that can be dragged and act as a modal window. | - | - |
| Ngeen | Name given to this Game Engine(N - g - ee - n -> En - g - i - ne) | - | - |
| Overlay | White shapes apearing while selecting entities. | - | - |
| System Configuration | Describes what kind of entities it accepts. | - | - |
| Draw System | Handles the drawing of sprites and calling of shaders using a sprite batch. | - | - |
| Sprite Batch | Stores all immediate draw calls and does them only when necessary. | - | - |
| Overlay System | Does entities overlay. | - | - |
| Physics System | Calls the Box2d implementation every set interval. | - | - |
| Scene System | Processes the current scene(includes loading). | - | - |
| System Stage | Processes stages and other UI related things. | - | - |
| Gradle | Open source build automation system. | - | - |
| Android | Device operating system made by Google. | - | - |
| IOS | Device operating system made by Apple. | - | - |
| Windows Phone | Device operating system made by Microsoft. | - | - |