COMP20050 Software Engineering Project 2

11. Intro to LibGDX

Assoc. Prof. Pavel Gladyshev

UCD School of Computer Science. **Scoil na Ríomheolaíochta UCD.**

LibGDX

Created by Mario Zechner as a java library for Android game development (GD) and effects (X). Support for desktop environment was added to simplify development process, HTML5 and iOS support was added later.

Features:

- High-level 2D APIs (Textures, Sprites, Bitmap fonts, Tile maps...)
- Audio support
- Gesture recognition
- Utility functions to handle JSON, XML, cross-platform File I/O
- High-level 3D API
- Cross-platform networking support

LibGDX is cross platform

LibGDX provides platform independent APIs, but relies on platform-specific libraries to realise it

- **Desktop** target uses LWJGL (Lightweight Java Game Library)
- **Android** target uses Android SDK
- **HTML5** target uses Google Web Toolkit (gwt) that compiles Java to JavaScript
- **iOS** target uses RoboVM that compiles Java into Objective-C

LWJGL

LWJGL is a Java “wrapper” for OpenGL functionality

- MineCraft is built using LWJGL

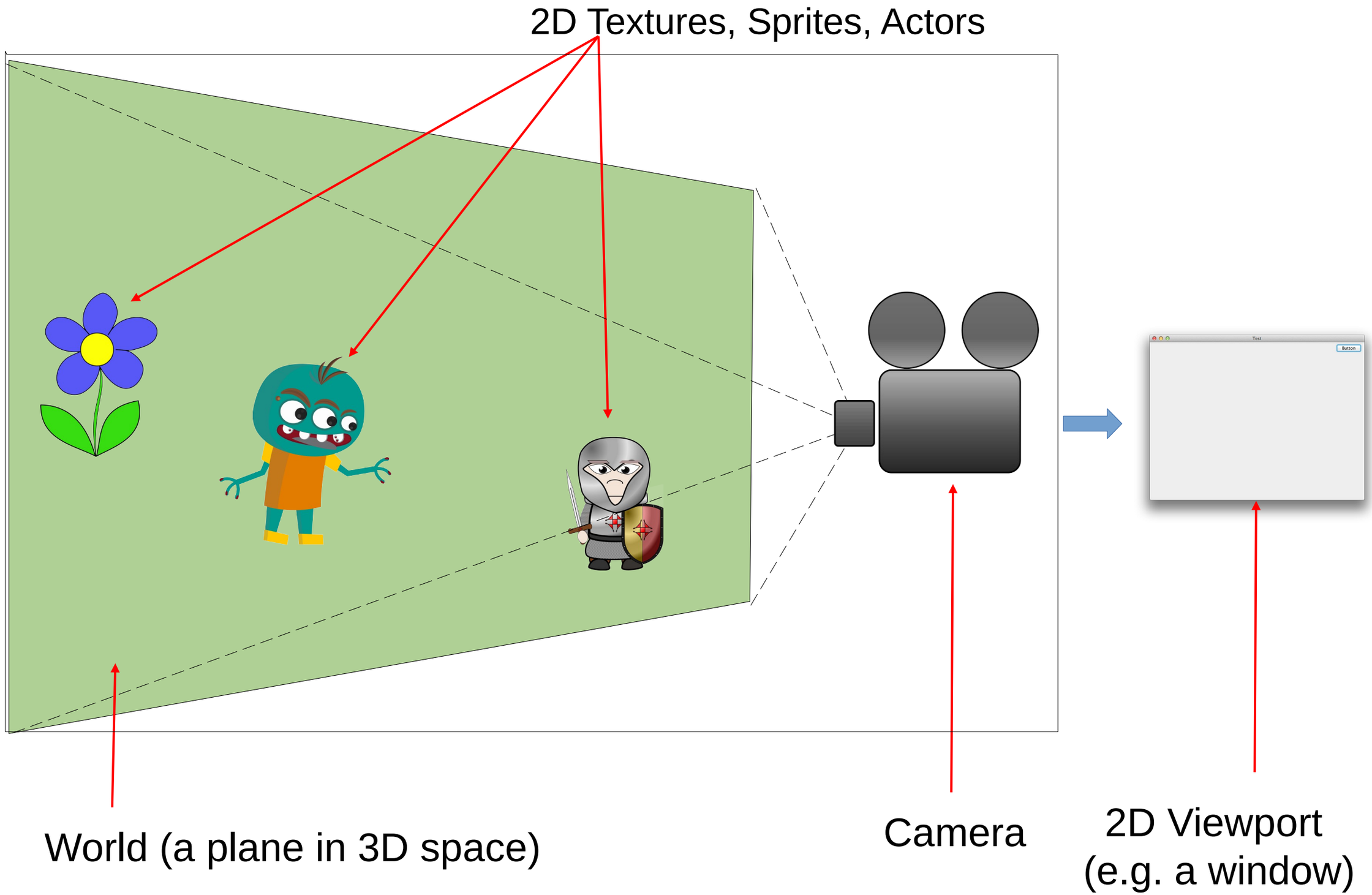
OpenGL stands for “Open Graphics Library”

OpenGL is an open standard that defines common APIs for specialised graphcis hardware.

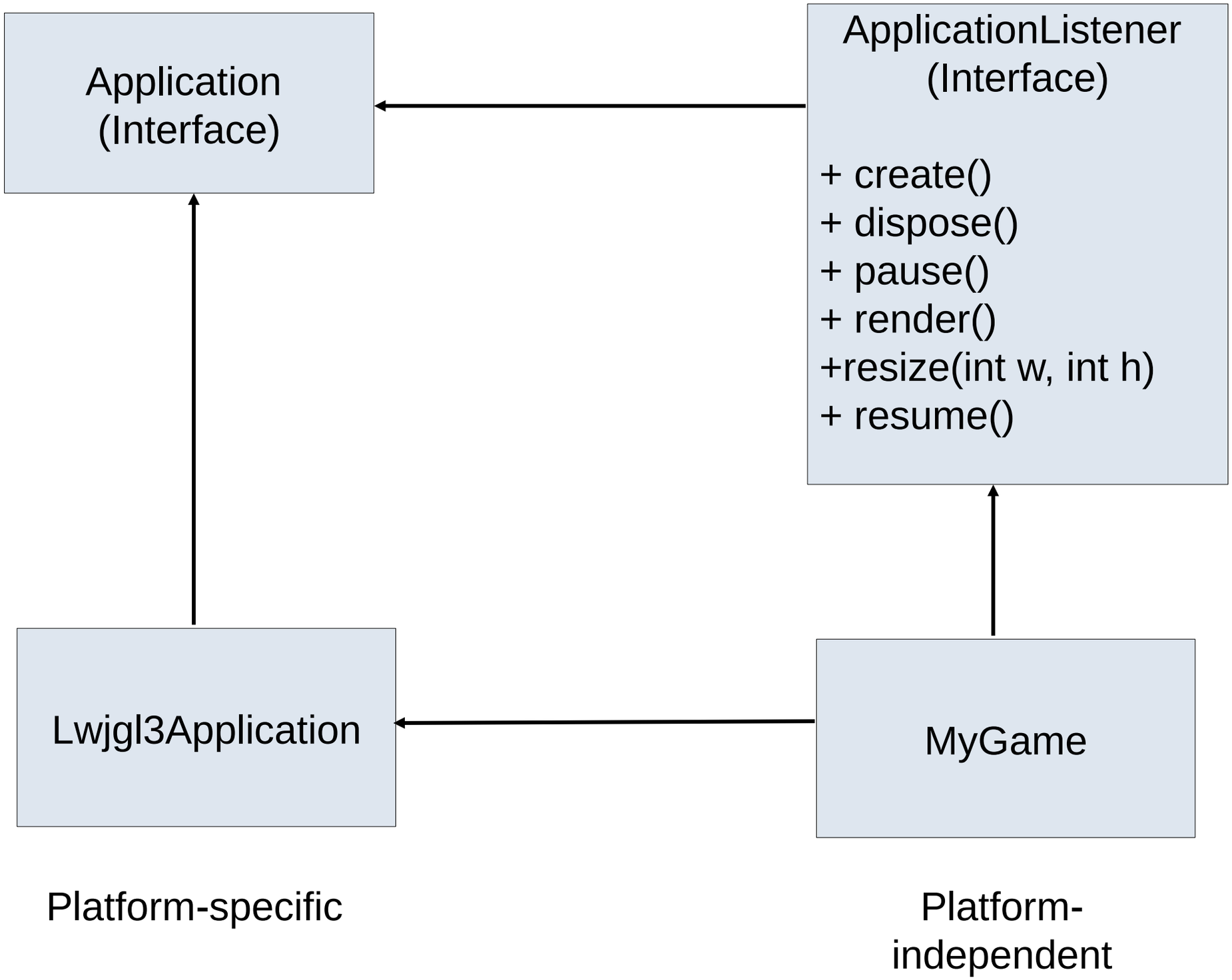
- Maintained by Khronos Group
- Originally designed by Silicon Graphics Inc.
- OpenGL is being replaced by Vulkan standard.

3D models and textures

LibGDX uses 3D graphics for 2D games



LibGDX application structure



ApplicationAdapter, Game and Screen classes

ApplicationAdapter class implements ApplicationListener interface

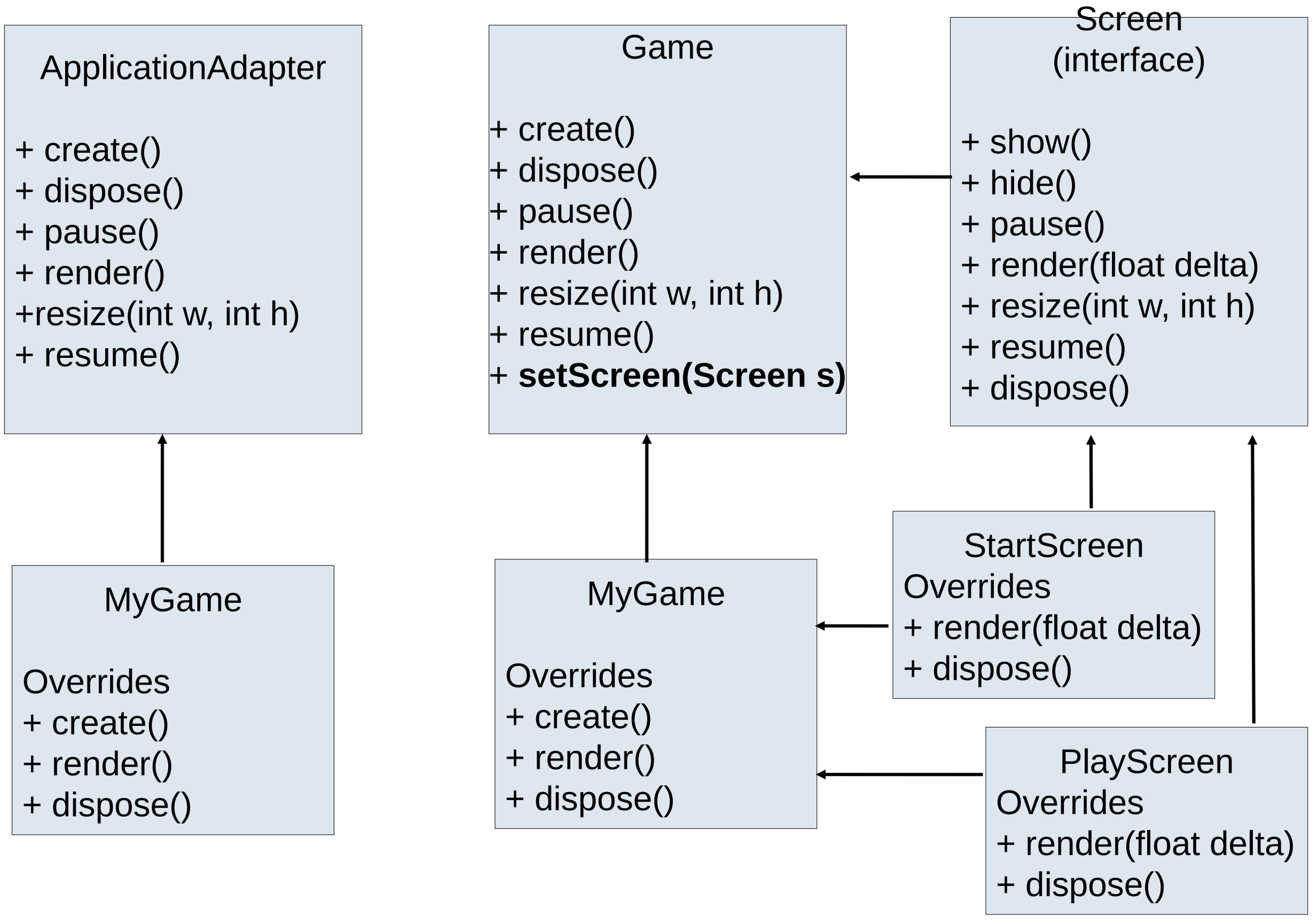
Provides default implementations for all methods

It is common for games to subclass ApplicationAdapter and override only some of the methods

Game class is analogous to ApplicationAdapter class, but provides additional methods to handle games with multiple screens.

Screen class works in tandem with Game class to provide screen-specific implementation of render() and other methods

Adapters


ApplicationAdapter
+ create()
+ dispose()
+ pause()
+ render()
+resize(int w, int h)
+ resume()
Game
+ create()
+ dispose()
+ pause()
+ render()
+ resize(int w, int h)
+ resume()
+ setScreen(Screen s)
Screen
(interface)
+ show()
+ hide()
+ pause()
+ render(float delta)
+ resize(int w, int h)
+ resume()
+ dispose()
MyGame
Overrides
+ create()
+ render()
+ dispose()
MyGame
Overrides
+ create()
+ render()
+ dispose()
StartScreen
Overrides
+ render(float delta)
+ dispose()
PlayScreen
Overrides
+ render(float delta)
+ dispose()