Google™ 11 I O

Kick-Ass Game Programming with Google Web Toolkit

Ray Cromwell, Philip Rogers

May 11th, 2011

Demo time!

What’s covered in this session

- Intro / Why GWT?
- Architecting a game
- Introducing ForPlay
- Let’s build a game
  - and share it
- Advanced Topics
  - Android
  - Flashy

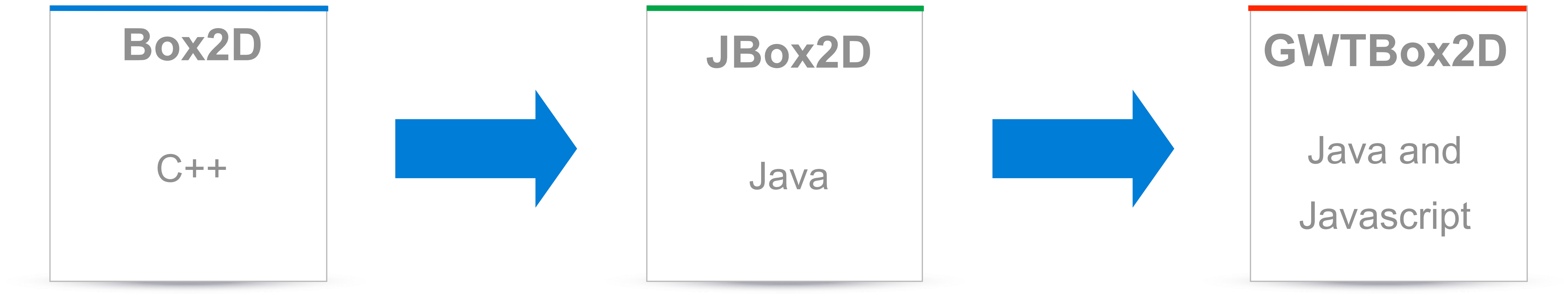
Why use GWT for games?

GWT = Game Web Toolkit?

- Leverage familiar Java toolchain (debugger, IDE, etc.)
- Share code between client, server... and other platforms?
- Produce small, fast JavaScript & HTML5

Leveraging familiar Java tools and libraries

Port a physics engine



- Box2D
  - C++ 2D Physics engine by Erin Catto
- JBox2D
  - A port of Box2D from C++ to Java
- GWTBox2D
  - A port of JBox2D from Java to JavaScript
  - “Porting” 11,000 lines took ~30 minutes. (removed ThreadLocal, etc.)

Google 11 I/O

Faster and Smaller

- GWT Compiler optimizes code for size
  - Removes unused code
  - Evaluates, when possible, code at compile time
  - Inlines functions
  - Heavily obfuscates the result
- Smaller code loads faster
- Perfect Caching avoids need for network I/O
- Inlining helps code run faster

HTML5 for games

- Formal HTML5
  - New HTML Elements
    - 2D Canvas, great for curves and text (GWT 2.2)
    - Audio, Video (GWT 2.2)
  - Application Cache (GWT 3.0)
  - Much, much more...
- Colloquial "HTML5"
  - 3D Canvas (WebGL) (GWTGL)
    - OpenGL ES 2.0 made JavaScript friendly
    - Important for 2D games due to acceleration
  - CSS3
    - Supports hardware accelerated transforms

Architecting a game

Overview of a game

The Game Loop

init()

update(delta)

paint(delta)

I/O System

keyboard, pointer, touch

audio, graphics

storage, network

Asset Management

loading

saving

caching

10

Google 11 I/O

Challenges of a cross-platform game

The Game Loop

Keeping graphics, update, and I/O in sync.

RequestAnimationFrame

Interpolation in paint()

I/O System

Not always available/accel:

WebGL, CSS3, Canvas

Audio

Inputs: touch, mouse, etc.

Asset Management

Loading

Caching assets

Application Cache

11

Google™ 11 I/O

Abstraction is key

GWT abstracts away browser differences.

Let’s apply that to cross-platform games.

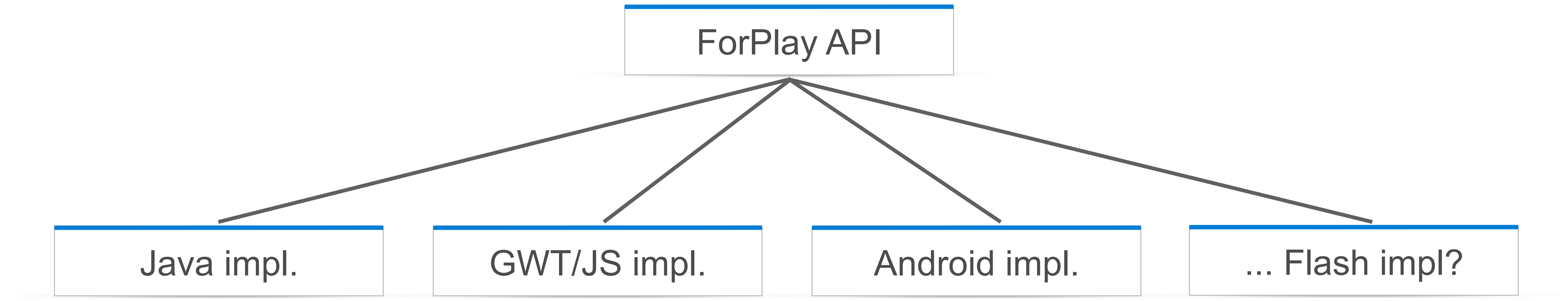
Introducing ForPlay

GWT abstraction layer for games

- A small API for building fast cross-platform games
  - Core game can be platform-agnostic
- Written in Java so you get a familiar language and toolset
  - And a great debugging environment
- GWT-compatible so you can compile to JavaScript/HTML5
- Free and open source (alpha version)
  - http://code.google.com/p/forplay

Introducing ForPlay

- Think: Service Provider Interface (SPI)



Demo time 🤢 🤢 🤢

Components of ForPlay

The Game Loop	I/O System	Asset Management

Extend ForPlay.Game
init()
update(float delta)
paint(float delta)
	import static core.ForPlay.*
	audio(), graphics(), net()
	
	Pointers & Keyboard input
	Layers for fast graphics
		assetManager()
		.getImage(String path)
		.getSound(String path)
		.getText(String path)
		also ResourceCallback

- ForPlay provides the cross platform magic to make these abstractions transparent
  - Core game has no platform-specific calls

The Game Loop | I/O System | Asset Management

Components of ForPlay

- Just extend implement ForPlay.Game

```
public class MyGame implements Game {
    public void init() {
        // init game
    }
    public void update(float deltaTime) {
        // update
    }
    public void paint(float deltaTime) {
        // paint
    }
}
```

Components of ForPlay

The Game Loop

I/O System

Asset Management

- Delta in paint() can be used for interpolation

```
public class MyGame implements Game {

    public void paint(float delta) {
        float xInterp = (delta) * x + (1-delta) * prevX;
        float yInterp = (delta) * y + (1-delta) * prevY;
        layer.setTranslation(xInterp, yInterp);
    }

}
```

18

Google 11 I/O

The Game Loop | I/O System | Asset Management

Components of ForPlay

- Input devices

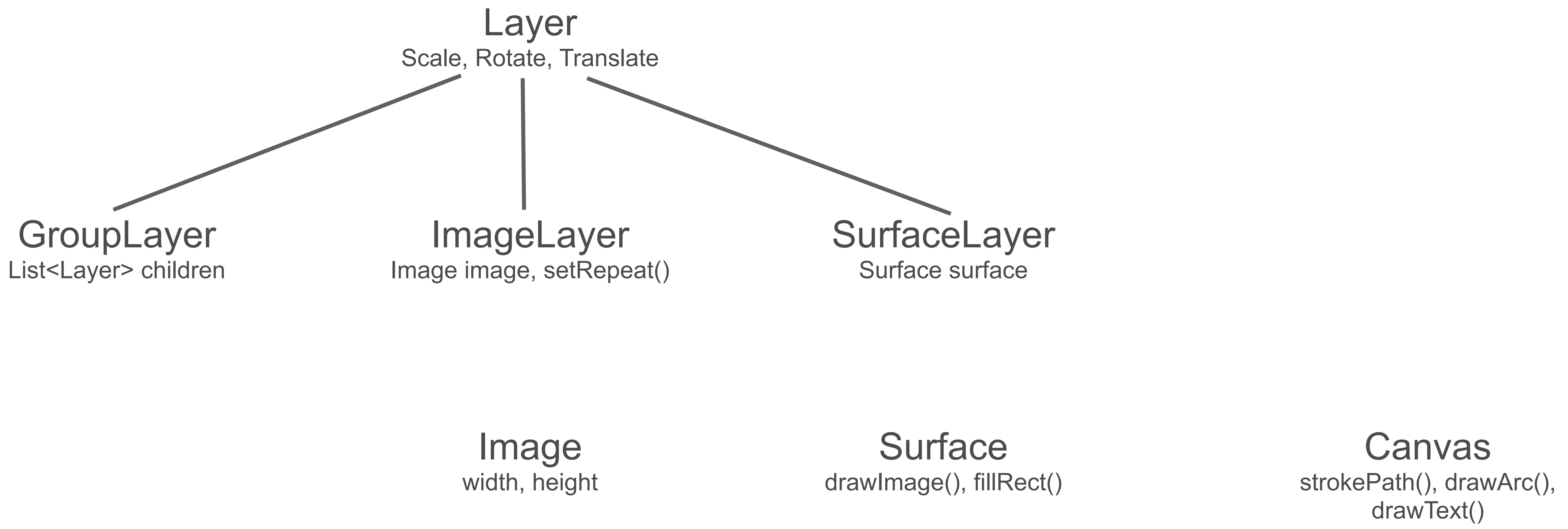
```
public class MyGame implements Game, Pointer, Keyboard {
    public void onPointerMove(int x, int y) {
        // handle pointer movement
    }
    public void onPointerScroll(int velocity) {
        // handle zoom, etc.
    }
    public void onKeyDown(int keyCode) {
        // handle keypress
    }
}
```

19

Google™ 11 I/O

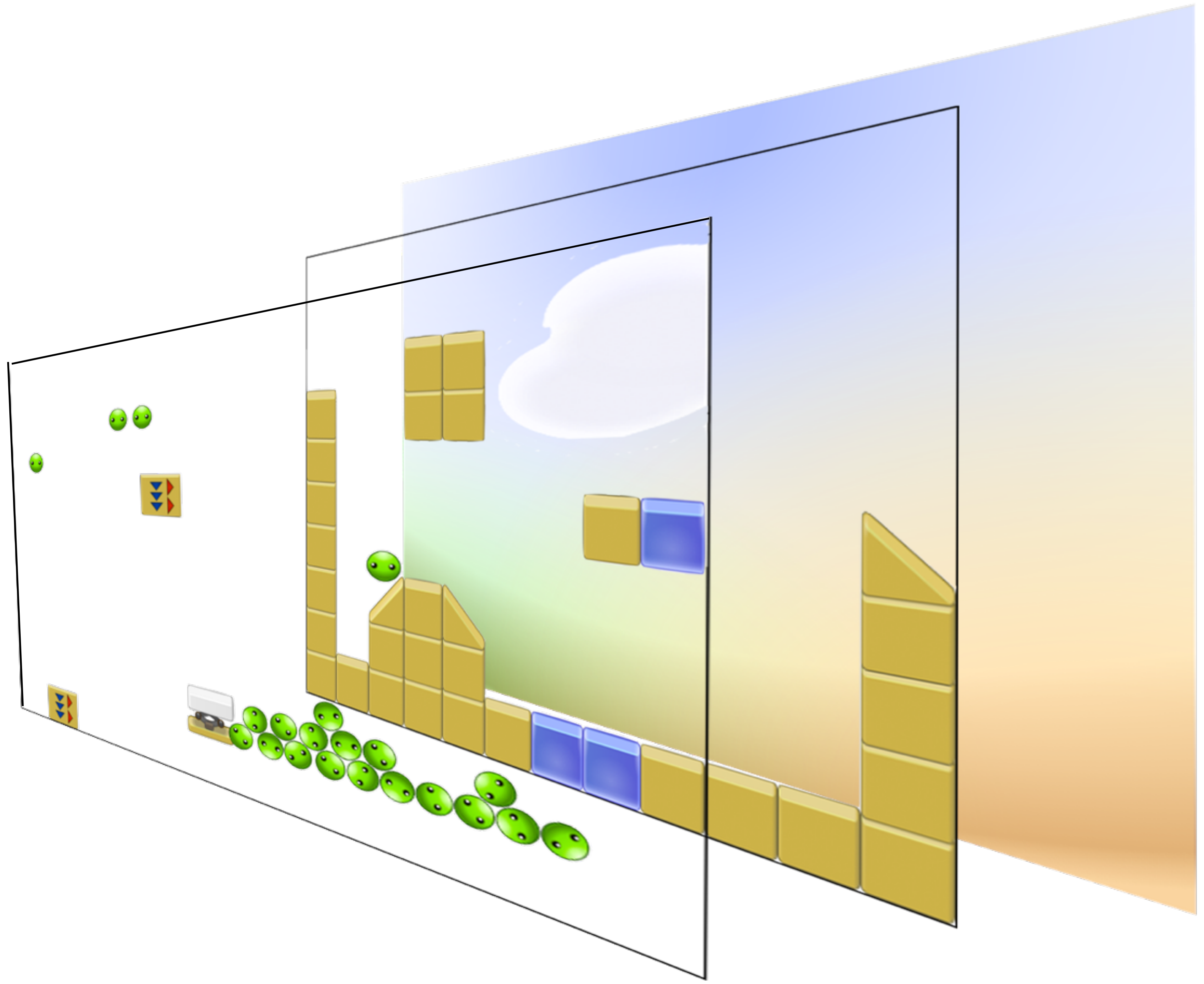
Components of ForPlay

- Images - using the ForPlay layer system



Components of ForPlay

The Game Loop | **I/O System** | Asset Management



The Game Loop

I/O System

Asset Management

Components of ForPlay

Loading and displaying images

```
Image image = assetManager().getImage(“myImage.png”);
image.addCallback(new ResourceCallback<Image>() {
    public void done(Image image) {
        // handle new image
    }
}
```

Google 11 I/O

Components of ForPlay

The AssetWatcher

```
AssetWatcher watcher = new AssetWatcher(new Listener() {
  public void done() {
    startMyGame();
  }
});

watcher.add(image1);
watcher.add(image2);
watcher.start();
```

ForPlay cross platform magic

- Game uses core ForPlay abstractions, is unaware of which platform is running
- The only platform-specific code is in the entry point for each platform:

```
public class MyGameHtml extends HtmlGame {
    public void start() {
        HtmlPlatform.register();
        ForPlay.run(new MyGame());
    }
}
```

```
public class MyGameJava {
    public static void main(String[] args){
        JavaPlatform.register();
        ForPlay.run(new MyGame());
    }
}
```

24

ForPlay wrap-up

- Open source, cross-platform game abstraction layer
  - Where the core game is platform-agnostic
- ForPlay abstracts away the core components of a game
  - The game loop, I/O system, and asset management
- Write in Java, get performance on multiple platforms
  - Java provides a great debug environment
  - GWT allows compilation to fast JavaScript/HTML5
- Lets build a game!

Demo of building peas 🟢 🟢 🟢

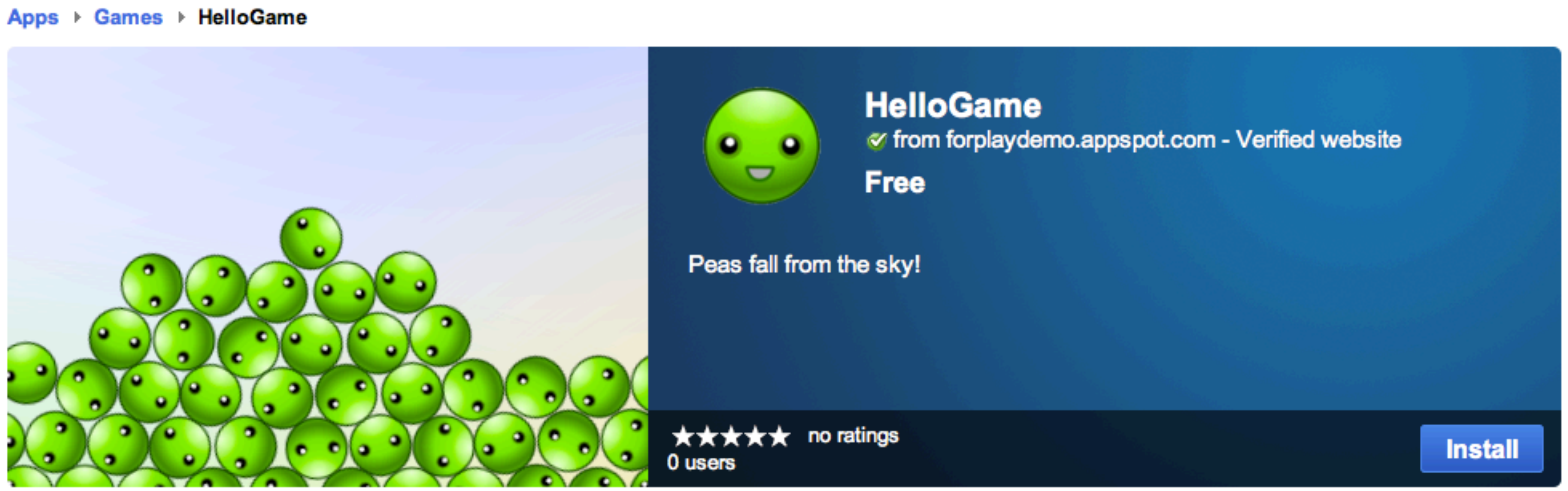
Building Peas

- Familiar tools
  - Write and debug in Java
  - Eclipse and GPE
  - Compile to Java and JavaScript/HTML5
  - Hosting on AppEngine is a click away

Releasing to the Chrome Web Store

1. Visit http://appmator.com, enter the URL of your game, and download a zip file.
2. Upload the zip file to the Chrome Web Store using the Developer Dashboard

...


Apps ▸ Games ▸ HelloGame

HelloGame

from forplaydemo.appspot.com - Verified website

Free

Peas fall from the sky!

★★★★★ no ratings

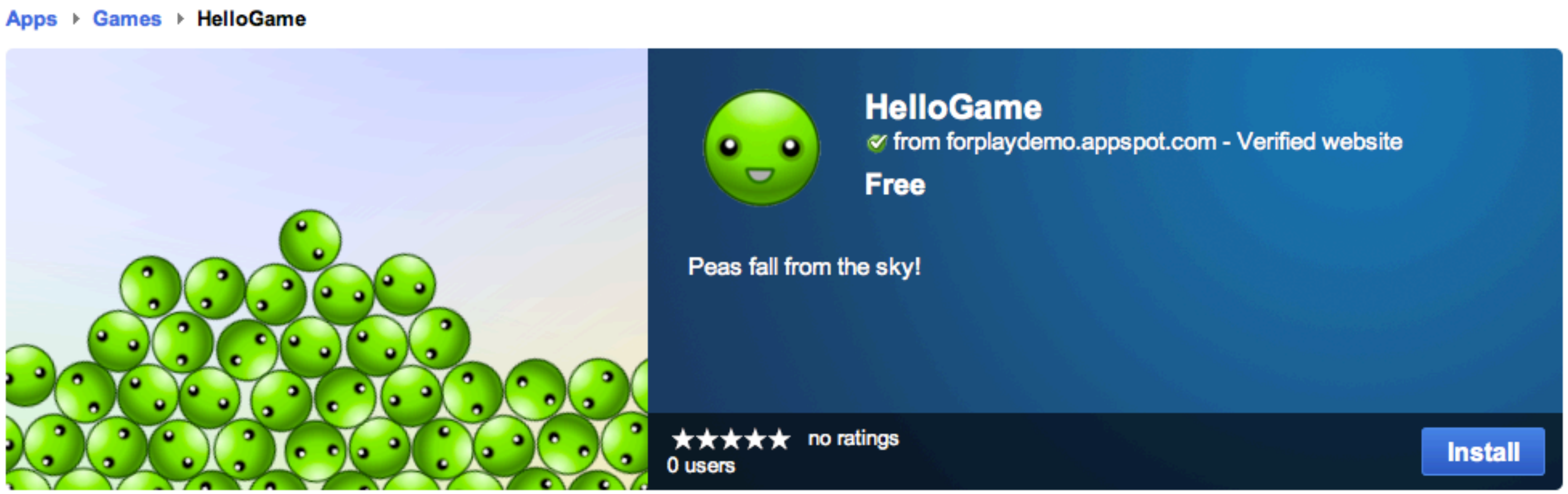
0 users

Releasing to the Chrome Web Store

1. Visit http://appmator.com, enter the URL of your game, and download a zip file.
2. Upload the zip file to the Chrome Web Store using the Developer Dashboard

 ...

3. Profit!


Apps ▸ Games ▸ HelloGame

HelloGame

✓ from forplaydemo.appspot.com - Verified website

Free

Peas fall from the sky!

★★★★★ no ratings

0 users

One more thing...

- That game we just built using ForPlay?
  - It runs as a Java desktop app too

One more thing...

- That game we just built using ForPlay?
  - It runs as a Java desktop app too
  - And an Android app

ForPlay for Android

- GWT code is already in Java
- Compile straight to Dalvik and package as an APK

```
public class MyGameAndroid {
    public static void main(String[] args){
        AndroidPlatform.register();
        ForPlay.run(new MyGame());
    }
}
```

One more thing...

- That game we just built using ForPlay?
  - It runs as a Java desktop app too
  - And an Android app
  - And in Flash

ForPlay for Flash

- GWT to Flex compiler
  - Compiles to ActionScript3/Flex4 SDK
  - Provides Java overlay types for flash.{events|net|display|media|etc} API
  - Specialized GWT linker produces SWF

```
public class MyGameFlash {
    public static void main(String[] args){
        FlashPlatform.register();
        ForPlay.run(new MyGame());
    }
}
```

Wrap-up

- Why GWT?
  - Performance, ease of writing code, portability
- Architecting a game
  - Core components: the game loop, I/O system, and asset management
- Abstraction is key
  - ForPlay is a cross-platform game abstraction layer
  - Easy to write in, performant
- Go build a game!
  - ForPlay is open source, easy to use, and really fun!
  - http://code.google.com/p/forplay

Links and Thanks

- ForPlay (alpha)
  - http://code.google.com/p/forplay
- Angry Birds
  - http://chrome.angrybirds.com

Links and Thanks

- ForPlay (alpha)
  - http://code.google.com/p/forplay
- Angry Birds
  - http://chrome.angrybirds.com

Google™ 11 I O