Commit d1c9f80

Browse files

ggcrunchy committed on Aug 7, 2025

```
Fix for stupid format issue, now have floating-point render targets (stumped about how to truly test ES2
Laying the groundwork for ES3 formats
```

FPTexture

1 parent 1c16443 commit d1c9f80

3 files changed +62 -30 lines changed

Search within code

librt/Renderer
Rtt_GL.h
Rtt_GLTexture.cpp
platform/android/sdk/src/co...
CoronaActivity.java

librt/Renderer/Rtt_GL.h +2

```
@@ -22,6 +22,8 @@
 #elif defined( Rtt_ANDROID_ENV ) || defined( Rtt_WEBOS_ENV ) || defined( Rtt_WIN_PHONE_ENV ) || defined( Rtt_EMSCRIPTEN_ENV )
 	#include <GLES2/gl2.h>
 	#include <GLES2/gl2ext.h>
+	#include <GLES3/gl3.h>
+	#include <GLES3/gl3ext.h>
 #elif defined( Rtt_POWERVR_ENV )
 	#include <GLES/egl.h>
 	#include <GLES/gl.h>
```

librt/Renderer/Rtt_GLTexture.cpp +59 -29

```
@@ -169,7 +169,10 @@ struct TextureFormatInfo {
 		kIsStencilRelated = 0x04,
 		kIsCompressed = 0x08,
 		kHasLinearFiltering = 0x10,
-
+
+		// TODO? kDecodeToUnorm... (for ASTC compression,
+		// so could repurpose another flag above, even)
+
 		kGreenBit = 0x20,
 		kBlueBit = 0x40,
 		kAlphaBit = 0x80
@@ -221,11 +224,6 @@ struct TextureFormatInfo {
 	U8 fBlockWidth; // for various compressed formats (we COULD repurpose fType and fSource...)
 	U8 fBlockHeight;
 	U8 fSizeFactor;
-
-	// ASTC: 128 bits in any
-	//       various
-	// BC*
-	// ETC2: 4x4, 8 bytes
-	// EAC: r, rg...
 };

 static TextureFormatInfo sInfo[32]; // arbitrary size; expand as necessary
@@ -629,13 +627,50 @@ EnumerateSupportedFormats()
 
 	if ( isES3 )
 	{
-		// TODO! are these in ES3 headers?
-	//	AllocInfo( compressedForm )->InitializeBlocked( "etc2", 4, 4 )
+	//	AllocInfo( compressedForm )->InitializeBlocked( "etc2", 4, 4 )
+		AllocInfo( compressedForm )->InitializeBlocked( "etc2-rgb", GL_COMPRESSED_RGB8_ETC2, 4, 4, 8 );
+		AllocInfo( compressedForm )->InitializeBlocked( "etc-rgba1", GL_COMPRESSED_RGB8_PUNCHTHROUGH_ALPHA1_ETC2, 4, 4, 8 );
+		AllocInfo( compressedForm )->InitializeBlocked( "etc2-rgba", GL_COMPRESSED_RGBA8_ETC2_EAC, 4, 4, 16 );
+		// EAC
 	}
 
 	bool hasASTC = false;
+#if defined(GL_KHR_texture_compression_astc_ldr)
+	hasASTC = (isES3 && 2 == esMinorVersion) || NULL != strstr( extensions, "GL_KHR_texture_compression_astc_ldr" );
+#endif
 
 	if ( hasASTC )
 	{
 		TextureFormatInfo astcForm = compressedForm;
+		// TODO: if can decode to UNORM8, set flag...
+
+		#define ASTC_PARAMS( WIDTH, HEIGHT ) "astc-" #WIDTH "x" #HEIGHT, GL_COMPRESSED_RGBA_ASTC_ ## WIDTH ## x ## HEIGHT ## _KHR, WIDTH, HEIGHT, 64
+
+		AllocInfo( astcForm )->InitializeBlocked( ASTC_PARAMS(4, 4 ) );
+		AllocInfo( astcForm )->InitializeBlocked( ASTC_PARAMS(5, 4 ) );
+		AllocInfo( astcForm )->InitializeBlocked( ASTC_PARAMS(5, 5 ) );
+		AllocInfo( astcForm )->InitializeBlocked( ASTC_PARAMS(6, 5 ) );
+		AllocInfo( astcForm )->InitializeBlocked( ASTC_PARAMS(6, 6 ) );
+		AllocInfo( astcForm )->InitializeBlocked( ASTC_PARAMS(8, 5 ) );
+		AllocInfo( astcForm )->InitializeBlocked( ASTC_PARAMS(8, 6 ) );
+		AllocInfo( astcForm )->InitializeBlocked( ASTC_PARAMS(8, 8 ) );
+		AllocInfo( astcForm )->InitializeBlocked( ASTC_PARAMS(10, 5 ) );
+		AllocInfo( astcForm )->InitializeBlocked( ASTC_PARAMS(10, 6 ) );
+		AllocInfo( astcForm )->InitializeBlocked( ASTC_PARAMS(10, 8 ) );
+		AllocInfo( astcForm )->InitializeBlocked( ASTC_PARAMS(10, 10 ) );
+		AllocInfo( astcForm )->InitializeBlocked( ASTC_PARAMS(12, 10 ) );
+		AllocInfo( astcForm )->InitializeBlocked( ASTC_PARAMS(12, 12 ) );
+
+		#undef ASTC_PARAMS
 	}
 
 	//
+	// GL_EXT_texture_compression_astc_decode_mode (not core in 3.2)
+	// glFePameteri(GL_TEXTURE_2D, GL_TEXTURE_ASTC_DECODE_MODE_EXT, GL_DECODE_MODE_ASTC_UNORM8_EXT);
+	// */
+#endif
+
 	bool hasRG = false;
 #if defined(GL_EXT_texture_rg)
 	hasRG = NULL != strstr( extensions, "GL_EXT_texture_rg" );
-	hasRG = isES3 || NULL != strstr( extensions, "GL_EXT_texture_rg" );
+	hasRG = isES3 || NULL != strstr( extensions, "GL_EXT_texture_rg" );
 	if ( hasRG )
 	{
 		AllocInfo( form )->initialized "red", GL_RED_EXT );
@@ -644,6 +679,7 @@ EnumerateSupportedFormats()
 #endif
 
 #if defined(GL_OES_texture_half_float)
-	bool has16BitFloats = NULL != strstr( extensions, "GL_OES_texture_half_float" ); // TODO: core in 3?
+	bool has16BitFloats = isES3 || NULL != strstr( extensions, "GL_OES_texture_half_float" ); // TODO: core in 3?
 	if (has16BitFloats)
 	{
@@ -650,7 +685,8 @@ EnumerateSupportedFormats()
 		bool hasLinearFiltering = NULL != strstr( extensions, "GL_OES_texture_half_float_linear" );
+		bool hasLinearFiltering = isES3 || NULL != strstr( extensions, "GL_OES_texture_half_float_linear" );
 		TextureFormatInfo f16Info = {};
 
 		f16Info.fFlags = hasLinearFiltering ? TextureFormatInfo::kHasLinearFiltering : 0;
@@ -656,6 +692,8 @@ EnumerateSupportedFormats()
 		AllocInfo( f16Info )->Initialize( "rgb16f", GL_RGB );
 		AllocInfo( f16Info )->Initialize( "rgba16f", GL_RGBA );
+		AllocInfo( f16Info )->Initialize( "rg16f", GL_RG, GL_RG16F_EXT );
+		AllocInfo( f16Info )->Initialize( "rgba16f", GL_RGBA, GL_RGBA16F_EXT );
 
 		// also _linear for each?
 		// Linear for magnification
@@ -663,6 +701,7 @@ EnumerateSupportedFormats()
 		if ( hasRG )
 		{
 			AllocInfo( f16Info )->Initialize( "r16f", GL_RED_EXT );
 			AllocInfo( f16Info )->Initialize( "rg16f", GL_RG_EXT );
+			AllocInfo( f16Info )->Initialize( "r16f", GL_RED_EXT, GL_R16F_EXT );
+			AllocInfo( f16Info )->Initialize( "rg16f", GL_RG_EXT, GL_RG16F_EXT );
 		#endif
 		}
 	}
@@ -705,7 +744,7 @@ EnumerateSupportedFormats()
 
 #if defined(GL_OES_texture_float)
-	bool has32BitFloats = NULL != strstr( extensions, "GL_OES_texture_float" ); // TODO: ditto...
+	bool has32BitFloats = isES3 || NULL != strstr( extensions, "GL_OES_texture_float" ); // TODO: ditto...
 	if (has32BitFloats)
 	{
@@ -712,6 +751,7 @@ EnumerateSupportedFormats()
 		// n.b. not core in 3.0
 		bool hasLinearFiltering = NULL != strstr( extensions, "GL_OES_texture_float_linear" );
 		TextureFormatInfo f32Info = {};
 
 		f32Info.fFlags = hasLinearFiltering ? TextureFormatInfo::kHasLinearFiltering : 0;
@@ -719,6 +759,8 @@ EnumerateSupportedFormats()
 		AllocInfo( f32Info )->Initialize( "rgb32f", GL_RGB );
 		AllocInfo( f32Info )->Initialize( "rgba32f", GL_RGBA );
+		AllocInfo( f32Info )->Initialize( "rg32f", GL_RG, GL_RG32F_EXT );
+		AllocInfo( f32Info )->Initialize( "rgba32f", GL_RGBA, GL_RGBA32F_EXT );
 
 		if ( hasRG )
 		{
 #if defined(GL_EXT_texture_rg)
 			AllocInfo( f32Info )->Initialize( "r32f", GL_RED_EXT );
 			AllocInfo( f32Info )->Initialize( "rg32f", GL_RG_EXT );
+			AllocInfo( f32Info )->Initialize( "r32f", GL_RED_EXT, GL_R32F_EXT );
+			AllocInfo( f32Info )->Initialize( "rg32f", GL_RG_EXT, GL_RG32F_EXT );
 		#endif
 		}
 	}
@@ -732,7 +774,7 @@ EnumerateSupportedFormats()
 	bool hasDepthTexture = false;
-	bool hasDepthTexture = false; // and again
+	bool hasDepthTexture = false;
 #if defined(GL_OES_depth_texture)
 	hasDepthTexture = NULL != strstr( extensions, "GL_OES_depth_texture" );
 	// is depth-related
@@ -758,15 +800,15 @@ EnumerateSupportedFormats()
 	bool has24 = NULL != strstr( extensions, "GL_OES_depth24" );
 	bool has32 = NULL != strstr( extensions, "GL_OES_depth32" );
 	*/
-
-	// TODO: if ES3+, i.e. 3 == esMajorVersion
+	// https://stackoverflow.com/a/7313411 / https://stackoverflow.com/a/56318581
 	// ^^^ some searching suggested ES3 relaxes this, but not confirmed
 	// does have implications for external textures, though
 	// web? (ANGLE plugins, etc.)
-	// TODO: isES3OrBetter()? 3.0, 3.1, 3.2
 
 	// GL_EXT_texture_compression_s3tc
 	// OES_depth_texture (GL_DEPTH_COMPONENT)
 	// OES_packed_depth_stencil (GL_DEPTH24_STENCIL8)
-
-	// astc, etc2 waiting for ES 3
 #endif
 
 	bool has53TC = false, has8DXT1 = false;
@@ -773,14 +815,12 @@ bool has53TC = false, has8DXT1 = false;
 	// there are also some ANGLE / WebGL things; sRGB
 }
 
-// TODO? can skip some of these with ARB_color_buffer_float or similar, per docs...
-// also, some seem to be guaranteed
-
 struct FBOCompleteChecks {
 	FBOCompleteChecks()
 	{
 		glGenTextures( 1, &fTexName );
 		GL_CHECK_ERROR();
+
 		glGenFramebuffers( 1, &fFBOName );
 		GL_CHECK_ERROR();
 	}
```

…tform/android/sdk/src/com/ansca/corona/CoronaActivity.java +1 -1

```
@@ -212,7 +212,7 @@ protected void onCreate(Bundle savedInstanceState) {
 			android.content.pm.ApplicationInfo applicationInfo;
 			applicationInfo = getPackageManager().getApplicationInfo(
 					getPackageName(), android.content.pm.PackageManager.GET_META_DATA);
-			esVersion = 32;
+			esVersion = 32;
 		} //  esVersion = 32;
 		if (applicationInfo != null && applicationInfo.metaData != null) {
 			wantsDepthBuffer = applicationInfo.metaData.getBoolean( "wantsDepthBuffer" );
```

Comments 0

Lock conversation

Comment

Unsubscribe You're receiving notifications because you're subscribed to this thread.