CGI Studio 3.7

Release Date: December 2018

General 3.7

Restructuring of platform definitions

The CourierPlatformDefs folders have been removed from the applications. The content is platform dependent and not application dependent. With this, every application can share the platform definitions. With this change we also removed the force of CMake flags in the platform definitions. These flags are not platform but application specific. This might have impact on your applications as the default settings may have changed. The CourierPlatformDefs name might have also changed to a more generalized name for the device (e.g. removal of some OS information in the platform name).

Toolchain files are moved from cgi_studio_candera to the appropriate device in cgi_studio_device.

Player changes

Player application renaming

The former CourierPlayer has been renamed to Player. Class names and code are adapted accordingly. The former CGIPlayer (Candera only) has been renamed to LightPlayer. The LightPlayer is a light weight player application which uses candera only. It is recommended to use CGIStudio with Courier support. In some cases this is not possible or has restricitons e.g. small devices. For this, the LightPlayer still exists. The LightPlayer has been moved from cgi_studio_dev to cgi_studio_player/LightPlayer.

Network connection replacing DLL approach

> This has been changed for Player. LightPlayer is still using the DLL approach.

The panel has loaded the player application as DLL. This has changed, so executable builds (Player.exe) are required on host.

> NativeDLL is only buildable for specific target builds which require a dll/shared object approach.

The connection has to be established by "Start application..." or "Attach" if the application is already running. Debugging under host starts the application - therefore an "attach" is required. The Panel connects over TCP/IP. Target application is server, therefore the panel has to know the IP of the target. New CMake flags have been added to:

- enable and disable the Network control feature (CGIAPP_NETWORK_CONTROL_ENABLED)
- set a port for the application (CGIAPP_NETWORK_CONTROL_PORT) The application can now be blocked right before the asset loading begins. The application has to be unblocked in this case. The application has command line parameters to change e.g. listener port in

case of port is blocked or already in use.

Player render order configuration change

The camera sequence number is now global for 2D and 3D by default. If the previous behavior that 2D gets rendered before 3D shall be preserved, then either change the camera sequence number accordingly or change myRenderConfiguration.Use2DBefore3D(false); to true.

Player action recording and replay

A recording feature has been added as first version to the panel. With this, requests from panel to application can be recorded and replayed again.

Player target support

The CourierPlatformDefs have been adapted. With this we officially support the Player on all targets but NoOS and targets without multi-threading. This includes the connection of Panel over TCP/IP if TCP/IP is available.

Candera 3.7

To keep backwards compatibility for any interface change in the public API the old interface has been kept and marked as deprecated. The deprecated interfaces will be removed in future versions. Compiler warnings will be generated when using a deprecated interface through the usage of Candera Macro CANDERA_DEPRECATED_3_7_0.

Candera Platform

Owner Requirement

OpenGL Offscreen Render Targets (Frame Buffer Objects) are now required to have an owner RenderTarget to ensure that the used EGL context is safe. For this reason RDenderTarget now has the "IsOwnerRequired" function which returns if a parent render target is needed. The concrete RenderTarget implementation needs to implement the function accordingly. If IsOwnerRequired returns true, an Upload of the RenderTarget will only succeed if the owner is set. RenderTarget3D has additionally a protected SetOwner function that classes derived from RenderTarget may call to set their parent render target. This mechanism can of course also be used for non-OpenGL implementations. Widgets and Behaviors need to take care that owners are set correctly.

GlInclude Header File

From now on, the GlInclude header file only includes Open GL header files if a CGIDEVICE_OPENGLES_xx has been defined. E.g. the NullRenderDevice does no longer include them.

Interface to load / store binary shader programs

Binary Shader Programs or Object can be retrieved from the driver and stored to an arbitrary persistence mechanism (e.g. a file.). This is supported by the RenderDevice and can be used by Device Packages. Therefoe RenderDevice::UploadStoredShaderBinaryProgram and RenderDevice::RetrieveShaderBinaryProgram have been added if the CANDERA_SHADER_PROGRAM_PERSIST_INTERFACE_ENABLED flag is defined.

Candera Base

Post Processing System

The new Post Processing System provides an interface to the user to implement their own post processing effects. Such effects are attached to cameras and get callbacks from the system every time the corresponding camera has finished rendering its scene. The effect then processes the camera's render target, and the system routes the render target automatically through the stack of effects.

For ease of use, post processing effects can also be scripted using Lua.

Math numeric stabilization

Standard rotation values like 0, 90, 180, 270 degree should result in stable matrix values. But, due to the calculation of the rotation matrix values the resulting rotated position values have been quite unstable. Therefore, these values are handled separately. Including an optimized handling of rotation value 0 (which no longer leads to sine and cosine calculations). The Math interface of Candera has been extended by NormalizeDegree and DegreeSineAndCosine to implement this optimization.

Candera::WidgetBase::SetStringId with disposer deprecated

The SetStringId interface with a disposer function of WidgetBase has been deprecated due to optimization of the the StringId memory consumption.

Touch event extended

Due to extension in the controls the touch event has been extended by a camera. For details please take a look into controls change log. Beside the additional camera parameter of the touch event the GraphicDeviceUnit interface has been extended by a GetSourceId method to enable an improved sorting of touchable behaviours in the controls.

Glyph Atlas Outline

The glyph atlas has been extended to support glyph outlines.

ThreadedAssetProviderDispatched Outline

Class ThreadedAssetProviderDispatched was updated to offer access the worker thread. This allows to set the worker thread priority.

Candera Engine 2D

Renderer 2D Statistics

Candera's 2D renderer statistics have been extended to keep statistics about uploaded bitmap images, and render camera calls.

Asset References

A scene can now reference assets that are not referenced by scene graph objects, and upload/unload them whenever the scene is uploaded/unloaded.

Candera Engine 3D

Renderer 3D Statistics

Candera's 3D renderer statistics now keep statistics about vertex buffers, shaders, cubemaps, direct textures, render camera calls, and render state cache hit/miss rates (if enabled).

The renderer statistics overlay has also been extended to support the new values.

Canvas Text Outline

The canvas text has been extended to support outlines.

Texel Size Auto Uniform

New auto uniforms have been added, called "u_TexelSize", "u_TexelSize1", .. "u_TexelSize8", which pass a vec2(1.0/(textureWidth*4), 1.0/(textureHeight*4)) to the uniform corresponding with the texture at the given unit. The only purpose for this is to support the outline reference shader.

Asset References

A scene can now reference assets that are not referenced by scene graph objects, and upload/unload them whenever the scene is uploaded/unloaded.

Renderer Functions

New functions were introduced in the Renderer to support implementation of post processing effects. Render targets can now be retrieved from the Renderer for temporary use, while a Blit function provides a convenient way to apply a shader to a full screen quad.

Candera Transitions

Transition reverse improvements

Optimizations on the transition reverse algorithm were done. Reverting an already reversing transition is possible now.

Transition requests

A Candera::Transitions::Request::End() might now return a null pointer depending on whether the request was filled with content or not. The framework changed so that it uses lazy object creation now to minimize memory operations.

Text engine

http://dev.doc.cgistudio.at/APILINK/class_

Candera Scripting

Lua Post Processing Scripts

The post processing system has also been integrated into the Lua script system. Candera Lua Post Processing offers an easy way to create post processing effects directly in Scene Composer using Lua.

Lua Bindings

Candera's Lua bindings have been extended with various functions to facilitate development of post processing effects.

Candera System

Component Handle

The ComponentHandle class from the entity component system has been refactored to use FeatStd's GenericHandle.

Courier V3.7

ViewScene2D/3D

New getter function was added to return the Scene Context

Control Behaviors 3.7

To keep backwards compatibility for any interface change in the public API the old interface has been kept and marked as deprecated. The deprecated interfaces will be removed in future versions. Compiler warnings will be generated when using a deprecated interface through the usage of Candera Macro CANDERA_DEPRECATED_3_7_0.

Behaviors CMake Build System Changes

Added flags:

Name	Description	Possible values
CONTROLS_HOVER_EVENT_ENABLED	Activate hover events from Control Touch Session. An explicit call to ControlTouchSession::SetHoverEnabled is not required in application.	ON/OFF

Behaviors Changes

Converted 2D/3D behaviors to "mixed" behaviors

Several behaviors had a dedicated 2D and 3D variant. Such behaviors have been cleaned up and replaced by a mixed behavior providing a 2D or 3D implementation based on the node attached to. The 2D and 3D specific behaviors have been deprecated, a solution conversion will automatically convert existing Scene Composer solutions.

Affected behaviors:

- ArcAlignBehaviorBase2D deprecated, replaced by ArcAlignBehavior
- TouchIndicationBehavior2D and TouchIndicationBehavior3D deprecated, replaced by TouchIndicationBehavior
- PositionProcessValueBehavior2D and PositionProcessValueBehavior3D deprecated, replaced by PositionProcessValueBehavior
  - property TranslationAxis has been renamed to TranslationAxis2D, a new property TranslationAxis3D has been introduced
- RotateProcessValueBehavior2D and RotateProcessValueBehavior3D deprecated, replaced by RotateProcessValueBehavior
  - property RotationAxis added
- ScaleProcessValueBehavior2D deprecated, replaced by ScaleProcessValueBehavior
  - property IsScalingAlongXEnabled has been renamed to ScaleAlongX
  - property IsScalingAlongYEnabled has been renamed to ScaleAlongY
  - new property ScaleAlongZ added
  - property VectorXY has been renamed to ScaleVector2D
  - new property ScaleVector3D added
- TranslateProcessValueBehavior2D and TranslateProcessValueBehavior3D deprecated, replaced by TranslateProcessValueBehavior
  - property VectorXY (Vector2) has been replaced by MultipliedVector (Vector3)
- TrackPositionProcessValueBehavior2D and TrackPositionProcessValueBehavior3D deprecated, replaced by TrackPositionProcessValueBehavior
- AlphaProcessValueBehavior2D and AlphaProcessValueBehavior3D deprecated, replaced by AlphaProcessValueBehavior
- ColorActionBehavior2D deprecated, replaced by ColorActionBehavior
- EffectColorBehavior2D deprecated, replaced by EffectColorBehavior
- ImageActionBehavior2D deprecated, replaced by ImageActionBehavior
  - property Texture has been added
- SetImageBehavior2D deprecated, replaced by SetImageBehavior

    - new property Textures added
- TextBehavior2D deprecated, replaced TextBehavior
- ZoomProcessValueBehavior2D deprecated, replaced by ZoomProcessValueBehavior.
- ScrollbarBehavior2D and ScrollbarBehavior3D deprecated, replaced by ScrollbarBehavior
- SwitchActionBehavior2D deprecated, replaced by SwitchActionBehavior
- SwitchImageProcessValueBehavior2D deprecated, replaced by SwitchImageProcessValueBehavior
    - property RenderNode has been renamed to TargetNode
    - new property Textures added
- SwitchRangeProcessValueBehavior2D deprecated, replaced by SwitchRangeProcessValueBehavior

Further changes:

- Constructor ClippingSetterTraverser(const Candera::Rectangle& clippingArea, Candera::Node2D& node) has been deprecated
    - Please use ClippingSetterTraverser(const Candera::Rectangle& clippingArea, Candera::AbstractNodePointer& node) instead
- PropertyProviderBehaviorBase: properties Texture, Appearance, Value have been added

---

ControlTouchSession considers Camera for sorting

In the past the camera was not considered by the touch handling. This resulted in ambiguous touched controls.

The ControlTouchSession has to sort behaviors according their "visual" order to assure, that touch inputs are forwarded to the "nearest" touchable behaviors. For this purpose a sub-struct ControlTouchSession::TouchedBehavior has been introduced. The camera will be used to retrieve the according Candera::RenderTarget. Consequently HW layer order, SW Layer order, surface id and camera sequence number are considered during sorting of behaviors.

Each behavior-camera pair will be treated separately, thus a behavior rendered by two cameras will be distinguished for each camera.

Beside the internal handling in the ControlTouchSession also the touch related events (TouchEvent, ClickEvent, PressedEvent, DragDropEvent and HoverEvent) have been extended by a touched camera. The old constructors have been deprecated. Please use the one with the additional camera instead.

For the same reason also KeyboardFocusManager has been enhanced by new methods KeyboardFocusManager::SetFocused(const Candera::AbstractNodePointer& node, const Candera::AbstractCameraPointer& camera) and KeyboardFocusManager::GetFocusedCamera(). Please also use the SetFocused with the extra camera parameter in future.

Implementation of geometric and tab order focus navigation

A new event class KeyboardFocusNavigationEvent has been added to reflect navigation input like a rotary input to navigate between the visual controls. Furthermore the controls will use the new RegisterNavigation and DeregisterNavigation interface of the KeyboardFocusManager to register themselves for the navigation.

An interface has been added to extend the ControlTouchSession by gesture detectors

The detection of Click, press/release and drag&drop has been transformed into such a gesture detector. Futhermore, a gesture detector for swipe/flick and a gesture detector for transformations (rotate, pan and pinch) has been added.

Further minor changes

- The return value of EventToMessageBridge::PostAsMessage has been changed to an enum type to reflect the exact outcome of the Post and to reduce the logging to errors only.
- A Getter/Setter interface has been added to the ControlTouchSession to enable/disable the consumption of touch events by the ControlTouchSession. If the touch event is not consumed the Courier Touch Session will forward the touch messages to all widgets in the old touch handling style as message.

Property Provider supports Value

- Property Provider (PropertyProviderBehavior) supports Value variant type

- Send Value As Event has now PropertyReference that can be set to Other and point to Property Provider

---

SelectionGroup memory leak patched

- Patched SelectionGroup memory leak when creating new ChangeSelectionEvent

---

Arithmetic operation behavior created

- Arithmetic operation behavior (ArithmeticProcessValueBehavior) can be placed in ProcessValue chain to perform simple arithmetic operations on provided values.
- It allows to perform addition, subtraction, multiplication and division operations on the passed value.

Send ChangeValueEvent on ProcessValueEvent behavior created

- <Send ChangeValueEvent on ProcessValueEvent> (SendChangeValueOnProcessValueEventBehavior) can be used as a termination of ProcessValue chain to invoke value change on connected control
- This Behavior sends a ChangeValueEvent which is meant for changing the value of an Value Behavior.
- The value can be absolute or relative (value will be added).

ValueBehavior supports broadcast feature

- If enabled, ValueBehavior can now be used to forward input value even if it is the same as current value stored in the behavior, which is useful in broadcasting context

---

Provide Alignment adds Alignment property to the control

- Provide Alignment can be used to add a controllable Alignment property to the control
- AlignmentBehavior enables setting the horizontal and vertical alignment of a node.

Tracks angle on touch with regard to control's base point

- Track angle behavior allows to obtain the angle (from the position of user interaction) with regard to control's base point, using the mathematical function atan2.
- TrackAngleProcessValueBehavior listens to the touch of the selected node, returns the angle between pressing, moving and releasing.

DialBehavior supports Dial control

- DialBehavior uses the angle obtained to calculate the rotation of the knob and change the position of the twisted knob to the value which is passed by ProcessValue function and ValueChangedEvent.
- DialBehavior receives angle in ProcessValueEvent or value in ChangeValueEvent. It changes to appropriate rotation for knob. Transmits the changed value through events (ValueChangedEvent and ProcessValueEvent).

CircularSlider behavior supports CircularSlider control

- CircularSliderBehavior draws a blending mask and rotates attached thumb.
- Creates respective effects on runtime, according to Candera build options.

BreadcrumbBehavior supports breadcrumb control

- A breadcrumb is used as a navigational aid in user interfaces.
- It allows users to keep track and maintain awareness of their location.
- Child nodes like Text, TextValue, TextButton or other nodes can react on TextEvent which is sent to child nodes.
- Also BreadcrumbBehavior sends ValueChangedEvent to the state machine with the last value that has been received.
- BreadcrumbActionBehavior allows you to control the breadcrumb.

Provide Appearance adds Appearance property to the control

- Provide Appearance can be used to add a controllable 3D Appearance property to the control
- Also extends Property Provider by Appearance.

Control State Behavior properly handles Disabled State

Previously it was possible to still interact disabled Control via Inputs. In 3.7 this is not happening anymore (e.g. a disaled button cannot be pressed).

Interpolate Value can be set to Instant Mode

Using the ChangeInterpolationModeEvent in a Behavior allows to control which Mode the Interpolate Value will operate.

Check Processed Value got extended by Almost Equal

Since comparing exact Float Values can lead to issues, the Check Processed Value Behavior now also supports checking if they are almost equal.

TrackPositionProcessValue can be set to only react on dragging the Knob

Normally the TrackPositionProcessValue would also react to Inputs outside of the Knob Node. By enabling the Drag On Knob Only property, this can be prevented.

Format String now supports all Variant Types

Format String did only support Integers and Float Values previously. This is now resolved.

DrawerActionBehavior / "Change Drawer State"

The DrawerActionBehavior has been created to support the new Drawer Control.

ParticleEmitterBehavior / "Particle Emitter"

The ParticleEmitterBehavior has been created to support the new Point Sprite Emitter Control.

Change Animation Mode by Event

The Animation Mode of InterpolateProcessValueBehavior can be changed by event now.

Input Range for MapProcessValueBehavior

The MapProcessValueBehavior has been enhanced by an input range for use cases where parts of the input range of the ProcessValueEvent shall be processed by several MapProcessValueBehavior. The input range is optional. If not set, then the input range is taken from the ProcessValueEvent.

RenderingEnabled Property now bindable

The property RenderingEnabled of behavior RenderNodeBehavior / "Render Node" is now bindable.

See also:

RenderNodeBehaviorBase::SetEnableRendering

FeatStd 3.7.0

FeatStd 3.7.0

FeatStd Stream interface

An interface for streams has been added. This shall be an abstraction layer for read and write actions. A stream implementation has been added for TCP/IP: See also FeatStd TCP/IP reimplementation Classes for buffered streaming and classes for primitive datatype reading and writing from and to streams have been added.

FeatStd TCP/IP reimplementation

The concept of TCP/IP has been reworked. There are two new classes available: TcpServer and TcpClient. The server can open a port to listen to. A connection to such a server will return a stream which is of type TcpClient. The client can then be used for communication. This concept allows multiple connections at the same time with the convenient stream interface. The old TcpIp interface is still available and extended by an own socket concept. Using the old functions work completely equal to the old concept. Therefore, only one connection to the server is allowed. After the connection is lost, it cannot be reused. A shutdown is required.

> The old TcpIp will be deprecated in future versione with an upgrade of Monitor to the new concept.

State machine

The state machine has been extended by a monitor feature. This feature can be enabled by the CMake flag: FEATSTD_STATEMACHINE_MONITOR_ENABLED. This feature adds events on different locations to track changes within the state machine. Depending on the type of change a MonitorEvent or a TransitionMonitorEvent will be sent. To receive this events, an event listener has to be attached to Candera::Internal::StateMachineBehaviorData::State::GetMonitorEventSource().

FeatStd 3.7.0

Generic Handle

Relevant functions were inlined, and a reverse lookup was added.

Thread Priority

Class Thread was enhanced with the posibility to set/get the thread priority. This feature is implemented for Posix, Windows and Integrity platform. For all other platforms setting the priority has no effect. See FeatStd::Thread class for more details.

Thread Nice Value

Class Thread was enhanced with the posibility to set/get the thread nice value. This feature is supported only for Linux Posix platforms. On all other platforms it has no effect. See FeatStd::Thread::SetNiceLevel for more details.

FeatStd 3.7.0

Updated AsyncRequestDispatcherWorkerThread interface

Class AsyncRequestDispatcherWorkerThread was updated with the option to set the worker thread priority on thread start.

SceneComposer 3.7

HMI Contract Synchronization Plugin

The Plugin does not show up anymore at startup if there are only unmachted items found. It only shows up if there are matched or missing items detected.

Behavior Path

The generation of paths for behaviors has been enhanced. The behavior groups and behavior building blocks are considered for the path generation. This leads to understandable paths, expecially if the same behavior name is used inside multiple building blocks our behavior groups. The path is stored in the generated asset and has impact on application code that dealing with Candera paths for behaviors.

Custom Properties Support

The plugins can register (extra) properties for solution items. These properties can be edited using the property grid and are stored in the solution files.In previous version of Scene Composer, these properties were called “Annotations”. In this new version, they were renamed to “Custom properties”, because a new Annotations concept was introduced.

The Type Manager service contains a method to register properties and another one to retrieve all properties registered (by the plugins) for a type. A reference to this service can be obtained by the plugin in the initialization method using the Unity Container. Plugin compatibility is preserved. The plugins can used the old methods, but they were marked as deprecated.

More information is available in chapter Custom Properties Support.

Controls 3.7

New Controls

SpinBox Control

A spin box or spin button is a control that enables users to increase or decrease the number value by a specific increment (often by 1, 5 or 10) via clicking an up or down arrow button.

IconButton Control

IconButton has similar functionality as the standard button, but in addition an icon (image) (defined via multiple properties per control state) is placed on top of the button.

Dial Control

A dial is used to set a value within a specified range or to select between different options which are represented by the value of the dial. The control knob changes the value when user touches and turns it. Degree of rotation corresponds to the desired value change. The dial snaps into its predefined positions defined by its step size.

Breadcrumb Control

A breadcrumb is used as a navigational aid in user interfaces. It allows users to keep track and maintain awareness of their location. A Breadcrumb shows the history/flow through the screen hierarchy.

Each breadcrumb element must contain BreadcrumbAction behavior for work properly. By default, we can use any element that contains the text control (for example: Text, TextValue or TextButton).

It is possible to create your own control for your own needs (e.g. adding a separator between items), that will work properly with the breadcrumb. Remember to add BreadcrumbAction behavior in own control.

When breadcrumb gets more values than it can display it will always collapse elements, replacing the display of the name with the ellipsis text (e.g. three dots). First element is always fully visible. Breadcrumb display can be controlled by event sent by BreadcrumbAction behavior.

Circular Slider Control

Circular slider is a control that allows user to set the value using circular slide movement or directly indicate value on slider's bar. In default version the slider is a full circle, but using properties user easily can operate on e.g. quarter of the circle. In case of circular slider user can set the start angle and the end angle and map that to output value which is also customizable. Change of the value is indicated by slider's bar and (optional) thumb position.

Drawer Control

The Drawer is a control that allows to hide content that can be shown again when it is needed.

Point Sprite Emitter Control

The Point Sprite Emitter is control for 3D to make Particle Effects.

Control Modifications

Text Value Control now supports Disabled color

Text Value Control now supports Disabled color and Enabled property (bindable through HandleControlState), so it can be directly used inside of other control supporting disabled state.

Text Button Control now supports Text color and Data Binding

Text Value Control now supports text color as property. The following properties are now bindable:

- PressedImage
- HoveredColor
- DisabledColor

Button Control: New Properties and Data Binding

The Button Control has following new properties:

- HoveredColorPropertyReference
- HoveredColorPropertyProvider
- DisabledColorPropertyReference
- DisabledColorPropertyProvider

The following properties are now bindable:

- PressedImage
- HoveredColor
- DisabledColor
