

# NVIDIA Ansel Photo Mode and NVIDIA Highlights Video Capture Tool

Bryan Dudash, Senior Manager Developer Technology

March 23<sup>rd</sup>, 2018



Booth #223 - South Hall

[www.nvidia.com/GDC](http://www.nvidia.com/GDC)







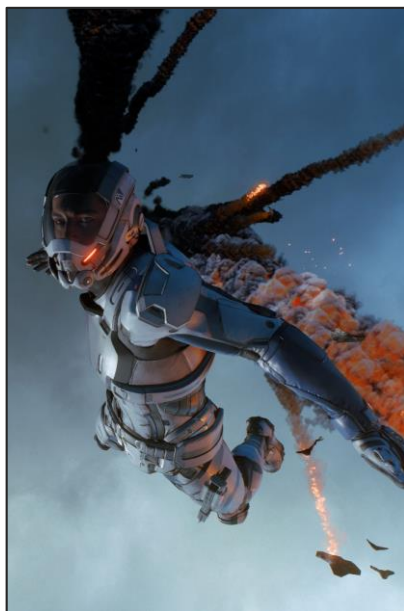
# Ansel takes in-game photography further



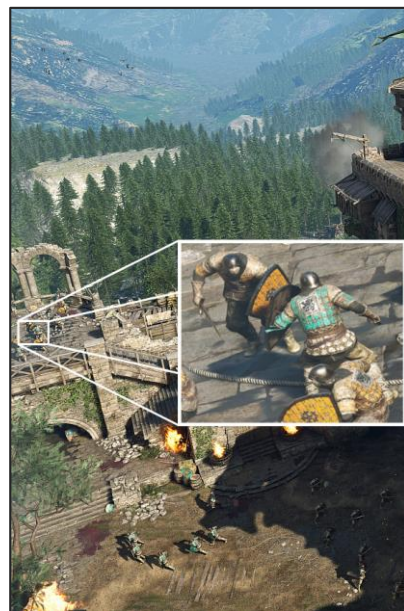
FREE CAMERA



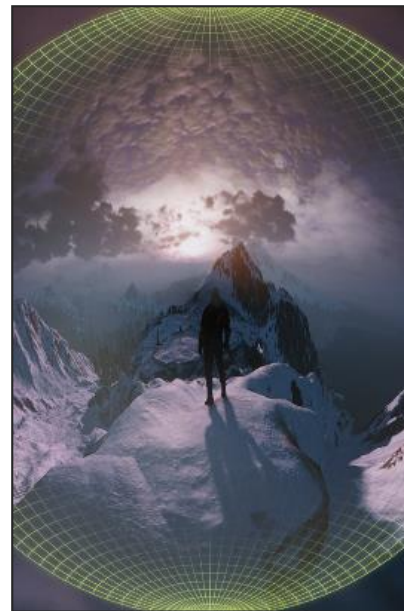
FILTERS



EXR



SUPER  
RESOLUTION



360



# MASS EFFECT: ANDROMEDA



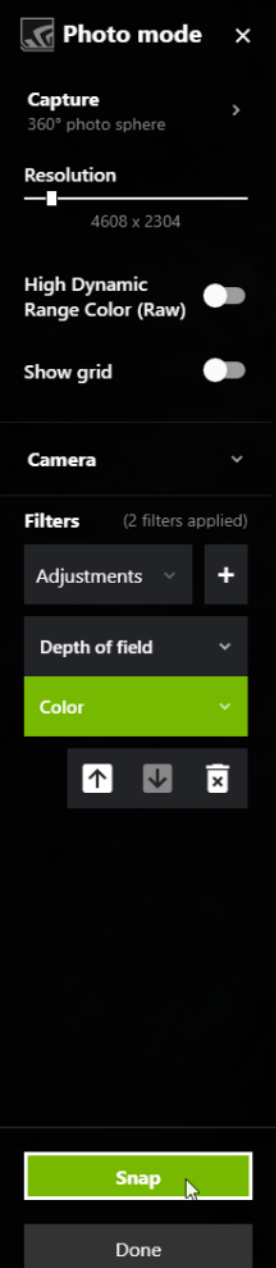


**STAR WARS™ BATTLEFRONT™ II**



# HELLBLADE: SENUA'S SACRIFICE

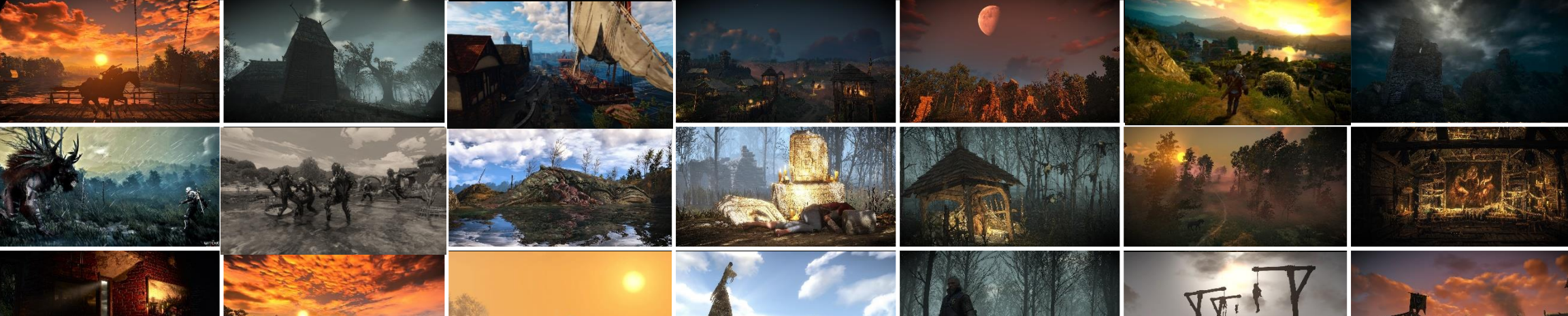




# EASY GAME INTEGRATION

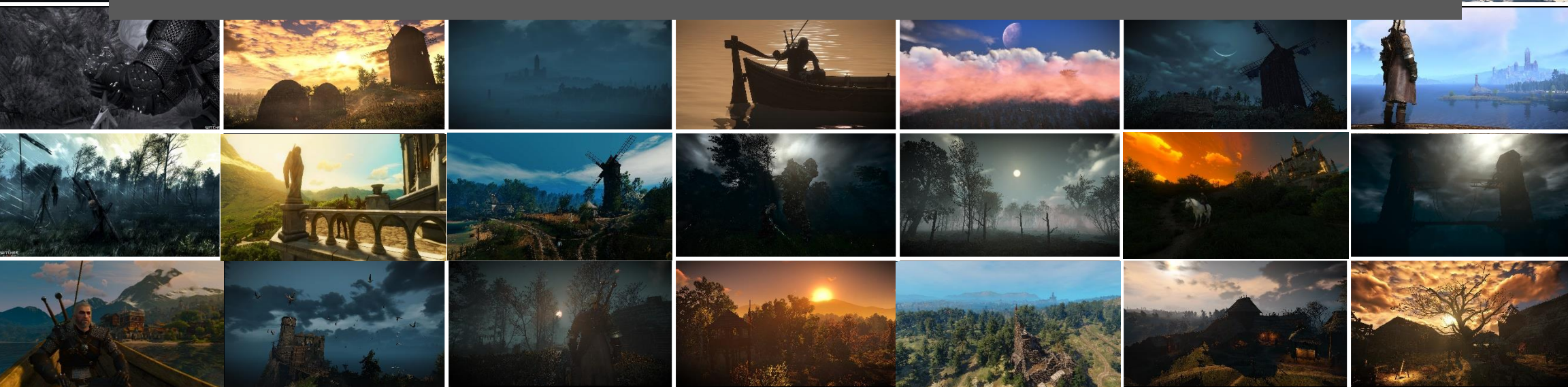
THE WITCHER 3 HAD 150 LINES OF CODE





# THE WITCHER 3: WILD HUNT

## 2M ANSEL PHOTOGRAPHS





# SHOT WITH GEFORCE®

THE GALLERY FOR GAMERS

Community Ansel Photographs

Super Resolution/ 360 / VR support

Upload directly from GeForce  
Experience

SHOTWITHGEFORCE.COM

# ANSEL SDK



[Gameworks Github](#)



[UE 4.14+](#)

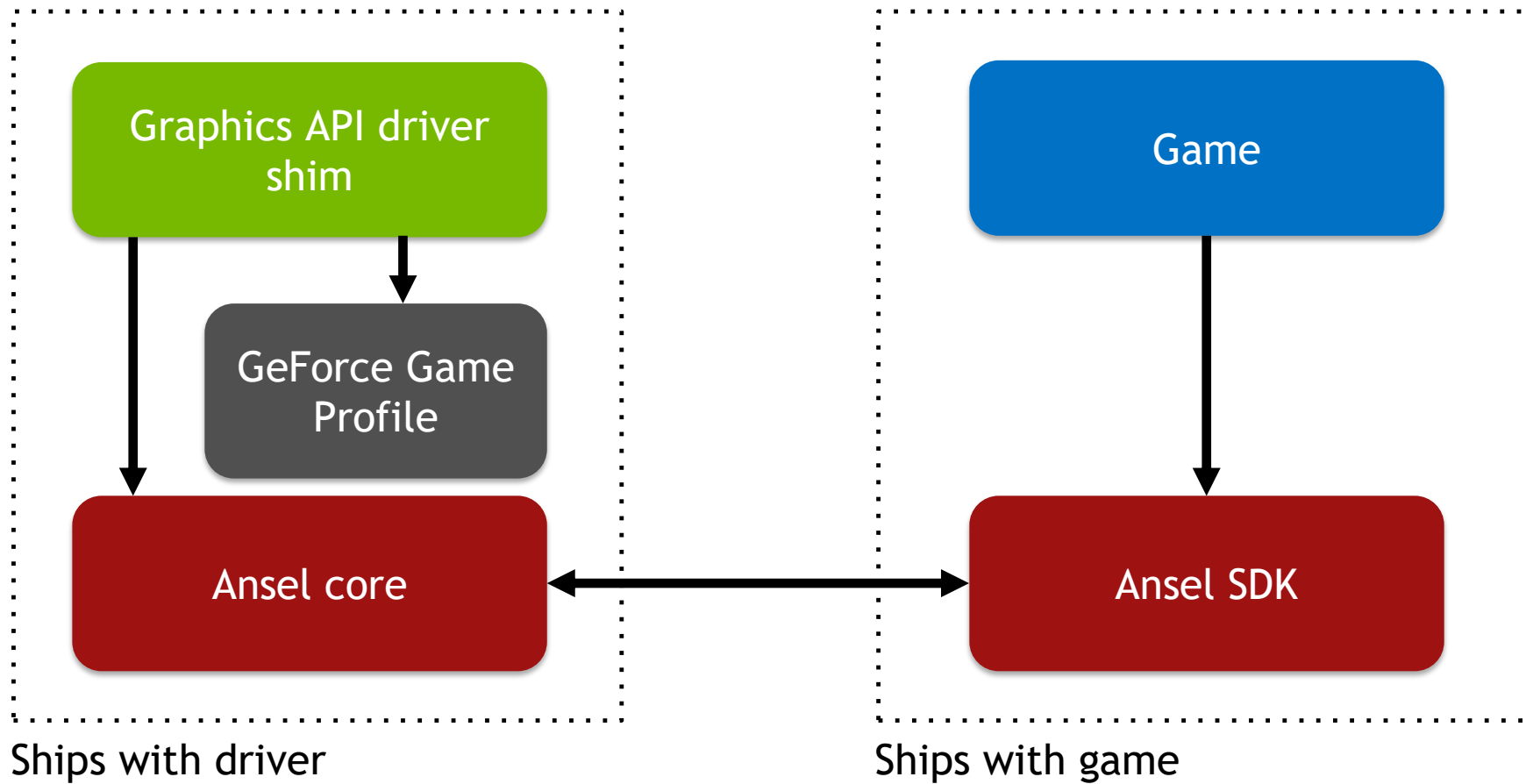


[Unity Plugin](#)

Available for all games and major engines  
Download from [developer.nvidia.com/Ansel](https://developer.nvidia.com/Ansel)



# Ansel Architecture





# Join the GameWorks developer program

- <https://developer.nvidia.com/Ansel>
- The above link will instruct you to do the following:
  - If you don't have an account on developer.nvidia.com or are not a registered member of the NVIDIA GameWorks developer program then register here: <https://developer.nvidia.com/developer-program/signup>
  - If you are logged in, accept the EULA and enter your GitHub username at the bottom of the form: <https://developer.nvidia.com/content/apply-access-nvidia-physx-source-code>
  - You should receive an invitation within an hour




# <https://github.com/NVIDIAGameWorks/AnselSDK>


 [NVIDIAGameWorks](#) / [AnselSDK](#) Private

 Watch ▾


21


 Star


25


 Fork

11


 Code


 Pull requests 0


 Projects 0


 Insights

This is where the SDK for NVIDIA Ansel is released to the public.

 32 commits

 6 branches

 9 releases

 2 contributors

Branch: master ▾


New pull request

Create new file








Upload files


Find file

Clone or download ▾

 **dmitryduka** committed 3 days ago Updating Ansel SDK

Latest commit [cd6e497](#) 3 days ago

|   |  |             |
|---|--|-------------|
|  <a href="#">docs</a>                        | Updating Ansel SDK   | 3 days ago  |
|  <a href="#">include</a>                     | Updating Ansel SDK   | 3 days ago  |
|  <a href="#">lib</a>                         | Updating Ansel SDK   | 3 days ago  |
|  <a href="#">redist</a>                      | Updating Ansel SDK   | 3 days ago  |
|  <a href="#">samples/AnselSDKIntegration</a> | Updating Ansel SDK 1.5, no functional changes. Fixing startSession/st... | a month ago |
|  <a href="#">.gitignore</a>                  | The import libs were missing because standard (global) .gitignore was... | a year ago  |
|  <a href="#">README.md</a>                   | Updating Ansel SDK   | 3 days ago  |

 README.md

## Ansel SDK

- Version: 1.5.427.6e7c9960
- Minimum driver version: 388.13

# Agenda

- Ansel Photo Mode
  - Run through the integration of Ansel into a game
  - Stop along the way to discuss some common issues and how to address them
  - Will not cover every corner case but that is what the docs are for



# The four concepts involved in integration

1. Configuration
2. Session
3. Camera
4. Hints (optional)

# Setting the Configuration

```
enum SetConfigurationStatus
{
    // successfully initialized the Ansel SDK
    kSetConfigurationSuccess,
    // the version provided in the Configuration structure is not the same as
    // the one stored inside the SDK binary (header/binary mismatch)
    kSetConfigurationIncompatibleVersion,
    // the Configuration structure supplied for the setConfiguration call is not consistent
    kSetConfigurationIncorrectConfiguration,
    // the Ansel SDK is delay loaded and setConfiguration is called before the SDK is actually loaded
    kSetConfigurationSdkNotLoaded
};

// Called during startup by the game. See 'Configuration' for further documentation.
ANSEL_SDK_API SetConfigurationStatus setConfiguration(const Configuration& cfg);
```



# Configuration contents

```
struct Configuration
{
    // Basis vectors used by the game. They specify the handedness and orientation of
    // the game's coordinate system. Think of them as the default orientation of the game
    // camera.
    nv::Vec3 right, up, forward;
    // The speed at which camera moves in the world
    float translationalSpeedInWorldUnitsPerSecond;
    // The speed at which camera rotates
    float rotationalSpeedInDegreesPerSecond;
    // How many frames it takes for camera update to be reflected in a rendered frame
    uint32_t captureLatency;
    // How many frames we must wait for a new frame to settle - i.e. temporal AA and similar
    // effects to stabilize after the camera has been adjusted
    uint32_t captureSettleLatency;
    // Game scale, the size of a world unit measured in meters
    float metersInWorldUnit;
    // Integration will support Camera::projectionOffsetX/projectionOffsetY
    bool isCameraOffcenteredProjectionSupported;
    // Integration will support Camera::position
    bool isCameraTranslationSupported;
    // Integration will support Camera::rotation
    bool isCameraRotationSupported;
    // Integration will support Camera::horizontalFov
    bool isCameraFovSupported;
```

# Default orientation in game coordinates

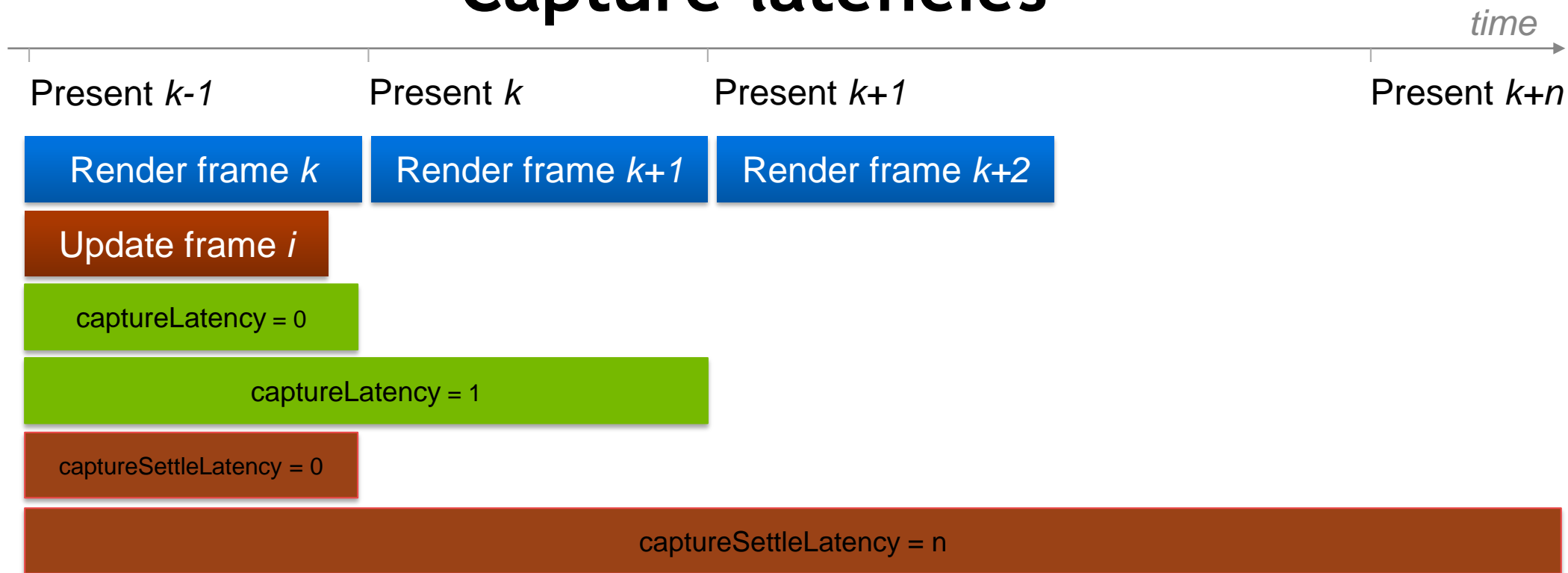


```
// Ansel SDK Sample default camera configuration
config.right    = { 1.0f, 0.0f, 0.0f };
config.up       = { 0.0f, 1.0f, 0.0f };
config.forward  = { 0.0f, 0.0f, 1.0f };
```

```
// Witcher 3 default camera orientation
config.right    = { 1.0f, 0.0f, 0.0f };
config.up       = { 0.0f, 0.0f, 1.0f };
config.forward  = { 0.0f, 1.0f, 0.0f };
```

```
// UE4 default camera orientation
config.right    = { 0.0f, 1.0f, 0.0f };
config.up       = { 0.0f, 0.0f, 1.0f };
config.forward  = { 1.0f, 0.0f, 0.0f };
```

# Capture latencies



**captureLatency** is the number of D3D present calls between update and present for a frame

**captureSettleLatency** is the number of D3D present calls between first present and final accumulation for a frame (temporal AA, etc)



# Discontinuous camera movement



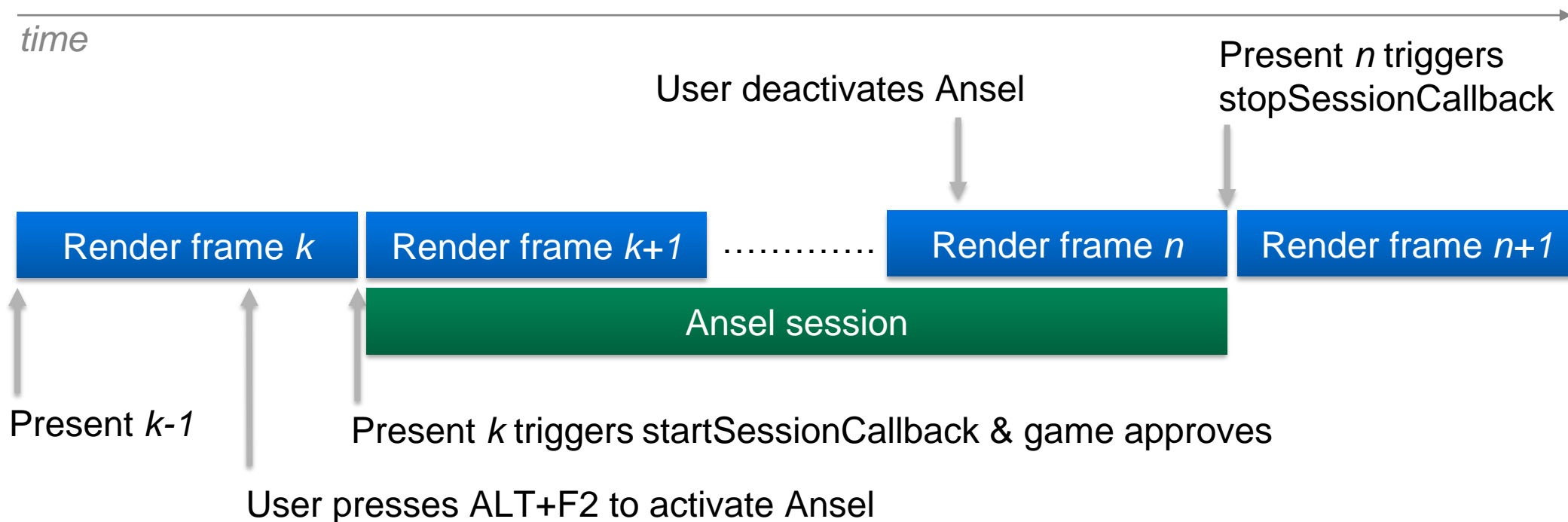
- Frame accumulation effects need time to settle (via `captureSettleLatency`)
- Or disabled during multipart shots

# Session

- Session is the period when a player is in Ansel mode
- Session is typically started & stopped by the player

```
// Called when user activates Ansel. Return kDisallowed if the game cannot comply with the
// request. If the function returns kAllowed the following must be done:
// 1. Change the SessionConfigruation settings, but only where you need to (the object
//    is already populated with default settings).
// 2. On the next update loop the game will be in an Ansel session. During an Ansel session
//    the game :
//    a) Must stop drawing UI and HUD elements on the screen, including mouse cursor
//    b) Must call ansel::updateCamera on every frame
//    c) Should pause rendering time (i.e. no movement should be visible in the world)
//    d) Should not act on any input from mouse and keyboard and must not act on any input
//        from gamepads
// 3. Step 2 is repeated on every iteration of update loop until Session is stopped.
StartSessionCallback startSessionCallback;
```

# Event timeline for a Session



```
typedef StartSessionStatus(__cdecl *StartSessionCallback)(SessionConfiguration& settings, void* userPointer);  
typedef void(__cdecl *StopSessionCallback)(void* userPointer);
```



Capture  
Screenshot

Resolution  
3840 x 2160

Save as HDR

Hide grid

Camera

Filters (0 filters applied)

Adjustments +

Add a filter to begin



Snap

Done





# Camera

```
struct Camera
{
    // Position of camera, in the game's coordinate space
    nv::Vec3 position;
    // Rotation of the camera, in the game's coordinate space. I.e. if you apply this
    // rotation to the default orientation of the game's camera you will get the current
    // orientation of the camera (again, in game's coordinate space)
    nv::Quat rotation;
    // Field of view in degrees. This value is either vertical or horizontal field of
    // view based on the 'fovType' setting passed in with setConfiguration.
    float fov;
    // The amount that the projection matrix needs to be offset by. These values are
    // applied directly as translations to the projection matrix. These values are only
    // non-zero during Highres capture.
    float projectionOffsetX, projectionOffsetY;
    // Values of the near and far planes
    float nearPlane, farPlane;
    // Projection matrix aspect ratio
    float aspectRatio;
};

// Must be called on every frame an Ansel session is active. The 'camera' must contain
// the current display camera settings when called. After calling 'camera' will contain the
// new requested camera from Ansel.
ANSEL_SDK_API void updateCamera(Camera& camera);
```

# Camera update during Ansel Session

```
// After Ansel SDK integration:
if (g_AnseSessionIsActive)
{
    ansel::Camera cam;
    // set up ansel::Camera object with the current camera parameters
    cam.position = { game_cam_position.x,game_cam_position.y,game_cam_position.z };
    cam.rotation = { game_cam_orientation.x,game_cam_orientation.y,game_cam_orientation.z,game_cam_orientation.w };
    cam.fov = get_game_fov();

    ansel::updateCamera(cam);

    // This is where the game would do collision detection against the new camera position, rotation
    // and adjust those values to any limits it may desire

    game_cam_position = { cam.position.x,cam.position.y,cam.position.z };
    game_cam_orientation = { cam.rotation.x,cam.rotation.y,cam.rotation.z,cam.rotation.w };
    set_game_fov(cam.fov);

    offset_game_projection_matrices(cam.projectionOffsetX, cam.projectionOffsetY);
}
```



# Offset and view angle for Super resolution





# Rotation and view angle for 360 photos



# Post-effects and multipart shots

- Most post-effects work fine but:
  - Non-uniform frame effects, like vignette, need to be disabled during multipart capture
  - Temporal frame effects like **motion blur** and **LOD fading** should also be disabled
  - Ansel has Vignette functions, so should disable in game vignetting unless really necessary.

# High bang to developer buck ratio

- Ansel SDK is easy to integrate
- Ansel engine plug-ins are even easier
- Most games require only minor modifications to support Ansel - multiplayer games with no replay functionality can be a challenge though
- Your players will thank you
- The beauty and wonder of your game will be captured and shared in stunning photos



# Links

<https://developer.nvidia.com/Ansel>

<https://github.com/NVIDIAGameWorks/AnselSDK>

“NVIDIA Ansel” in the Unity Asset Store

<https://assetstore.unity.com/packages/tools/camera/nvidia-ansel-74758>

Unreal Engine 4 contains Ansel integration in the main branch. Yay!

# NVIDIA HIGHLIGHTS

FULLY AUTOMATIC CAPTURE

Captures Video Automatically  
Based On Game Events

Leverages GeForce  
Experience overlay and  
ShadowPlay technology

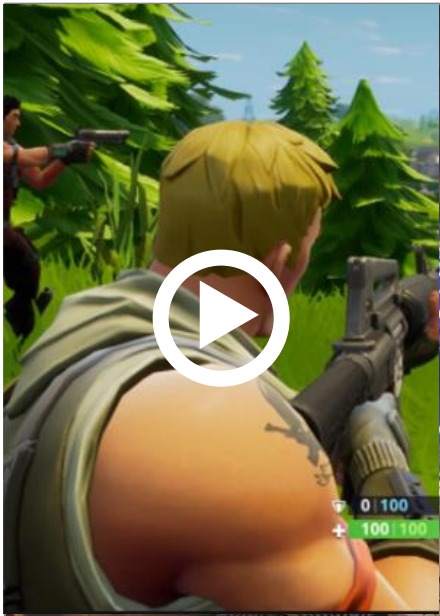
Easy Upload To YouTube,  
Facebook, Imgur, or Weibo



# What is Highlights?

- HIGHLIGHTS SDK allows games to automatically save screenshots and/or videos of important events

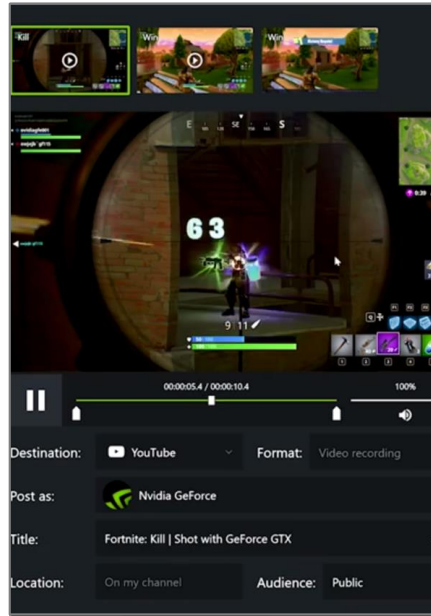
## VIDEO



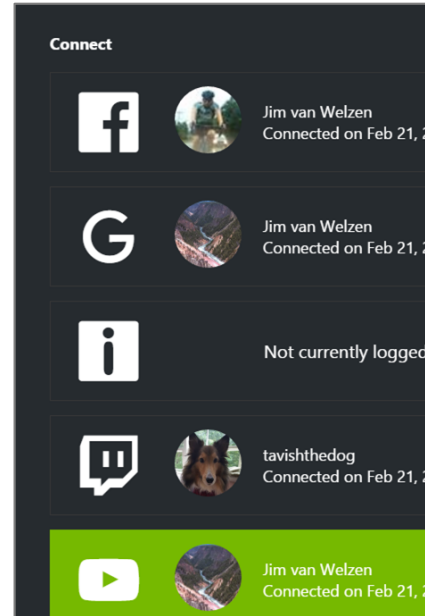
## SCREENSHOTS



## PREVIEW



## SHARE



## USER CONTROL



# HIGHLIGHTS IN FORTNITE BATTLE ROYALE





# What are the benefits?

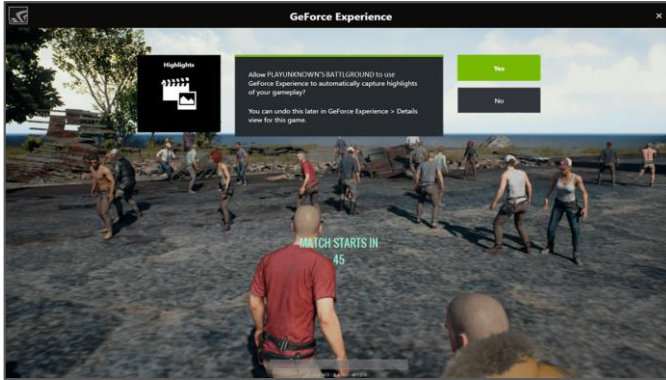
## ■ Features for Gamers

- Fully automatic
  - Hands free captures
  - No need to guess when something interesting might happen
- Increased immersion
- Increased replay value using highlights as achievements

## ■ Value for Developers

- Low overhead
  - Leverages existing ShadowPlay technology
- Simple interface
  - No need to implement from scratch
  - No need to incorporate a complicated or heavy API

# EXAMPLE INTEGRATION



Begin  
Match

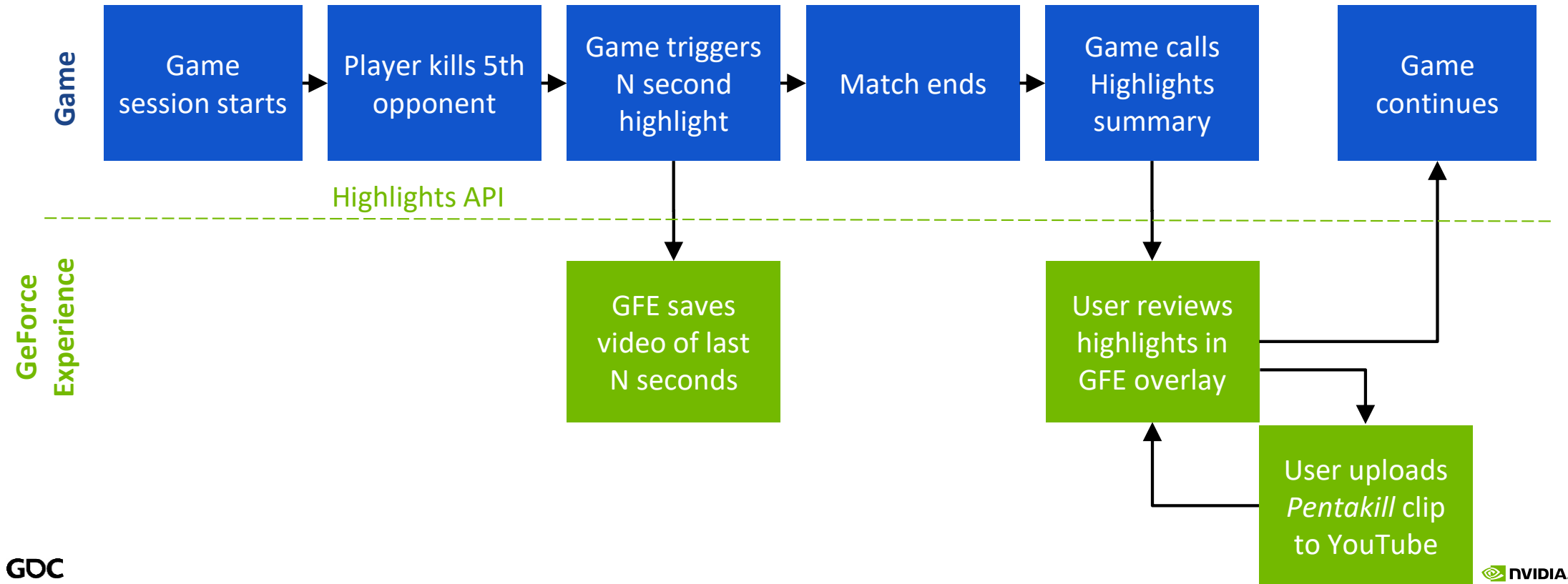


Highlight  
Notification



End of Game  
Summary

# Example Game Flow using Highlights

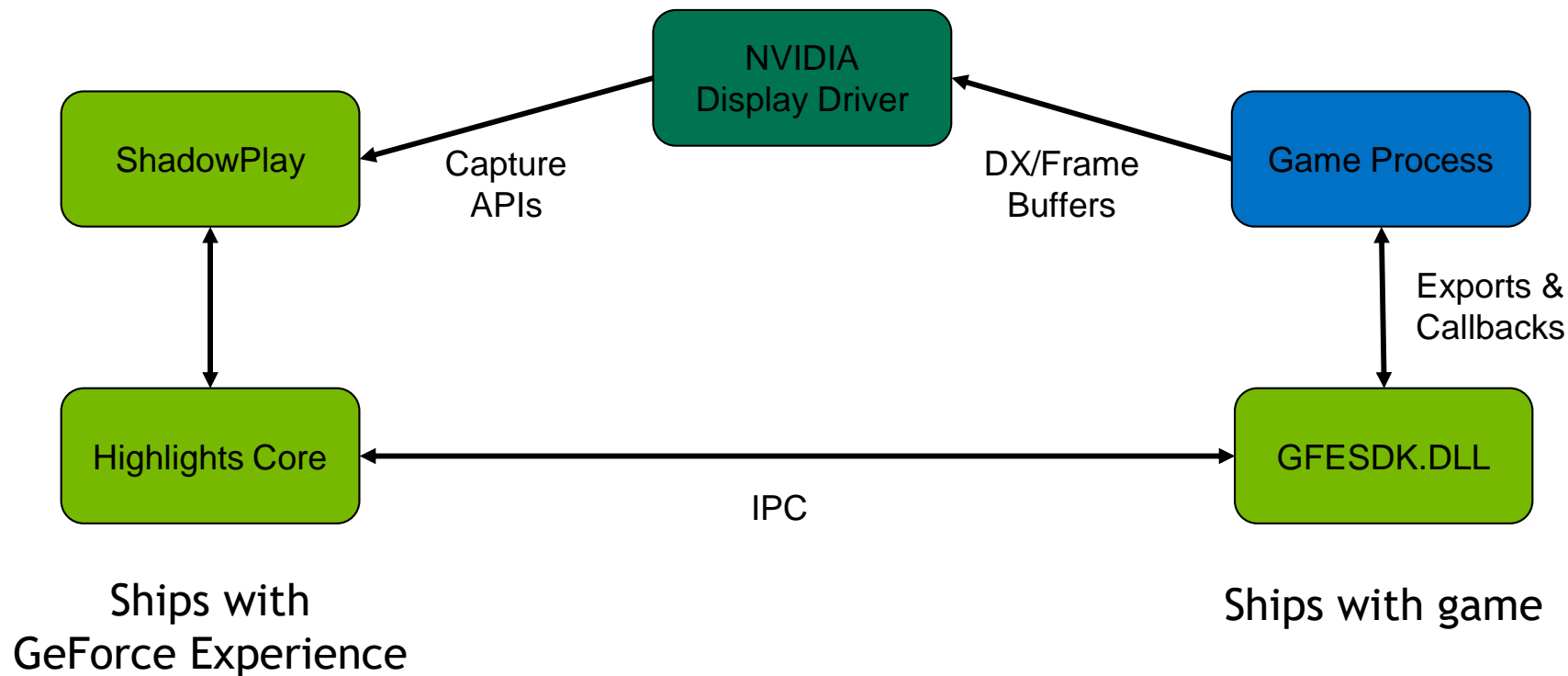


# What Makes a Good Highlight?

- A good highlight is an event that users will want to save
  - Something they will take pride in, find funny, or want to share via social media
- Some examples
  - Killing a boss
  - Multi-kills
  - Revenge kills
  - Discovering a hidden area
  - Rare accomplishments/treasure/combos
  - Odd or interesting deaths



# Highlights Architecture



# HIGHLIGHTS SDK AVAILABLE TODAY



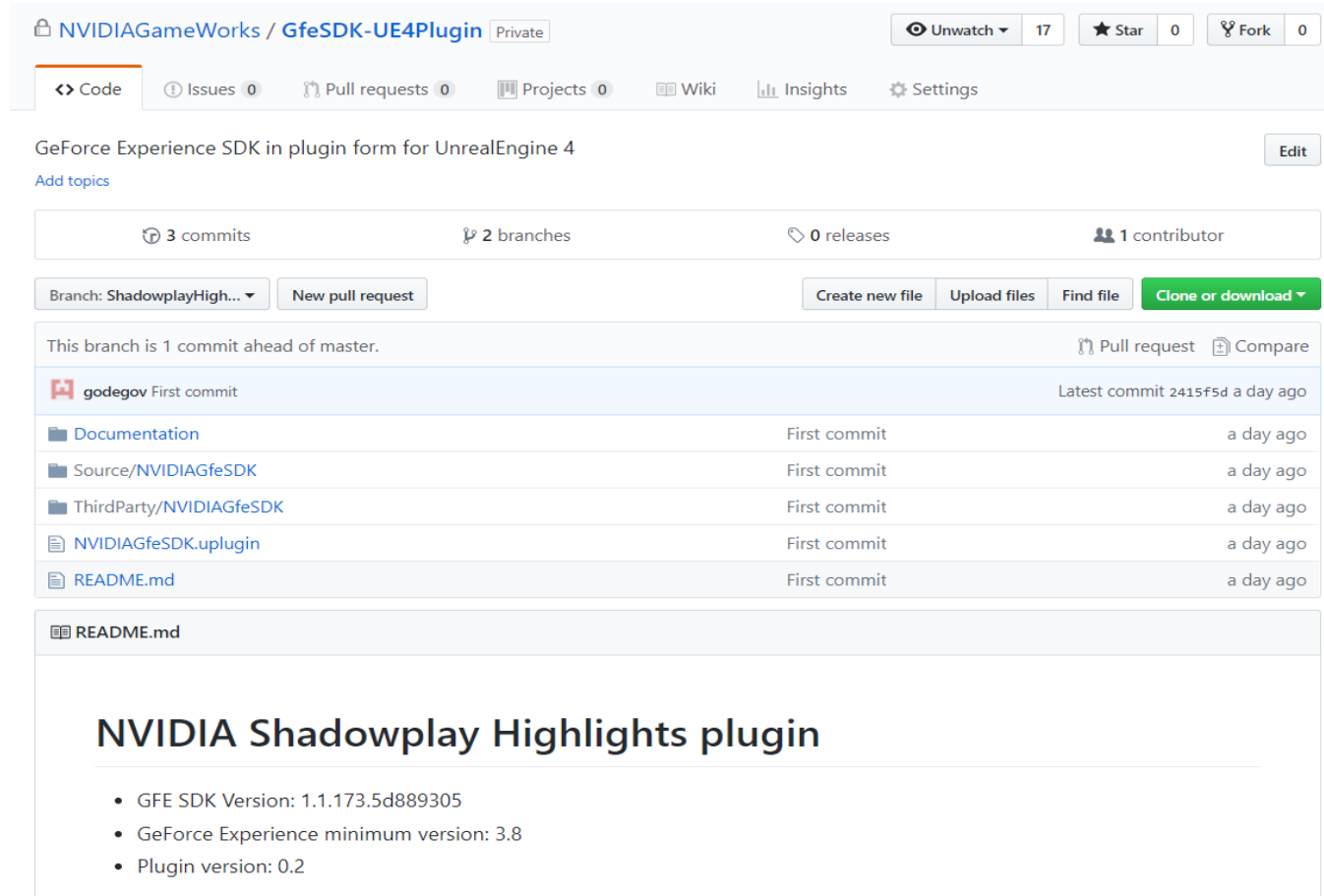
Available for all games and major engines  
Download from [developer.nvidia.com/highlights](https://developer.nvidia.com/highlights)

# 4 steps to access Highlights in Ue4

<https://github.com/NVIDIAGameWorks/GfeSDK-UE4Plugin>

In your copy of UE4 source:

1. In git bash/cmd navigate to  
Engine\Plugins\Runtime\Nvidia
2. `git submodule add -b 4.18`  
<https://github.com/NVIDIAGameWorks/GfeSDK-UE4Plugin.git> GfeSDK
3. Rebuild engine
4. Enable via plugins menu in Editor



The screenshot shows the GitHub repository page for `NVIDIAGameWorks / GfeSDK-UE4Plugin`. The repository is private and has 17 stars, 0 forks, and 0 issues. The description is "GeForce Experience SDK in plugin form for UnrealEngine 4". The repository has 3 commits, 2 branches, 0 releases, and 1 contributor. The current branch is `ShadowplayHigh...`. The commit history shows a single commit by `godegov` titled "First commit" from a day ago. The file list includes `Documentation`, `Source/NVIDIAGfeSDK`, `ThirdParty/NVIDIAGfeSDK`, `NVIDIAGfeSDK.uplugin`, and `README.md`. The `README.md` file content is visible, showing the title "NVIDIA Shadowplay Highlights plugin" and a list of requirements: GFE SDK Version: 1.1.173.5d889305, GeForce Experience minimum version: 3.8, and Plugin version: 0.2.

NVIDIAGameWorks / GfeSDK-UE4Plugin Private

Unwatch 17 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

GeForce Experience SDK in plugin form for UnrealEngine 4 Edit

Add topics

3 commits 2 branches 0 releases 1 contributor

Branch: ShadowplayHigh... New pull request

Create new file Upload files Find file Clone or download

This branch is 1 commit ahead of master. Pull request Compare

godegov First commit Latest commit 2415f5d a day ago

|                         |              |           |
|-------------------------|--------------|-----------|
| Documentation           | First commit | a day ago |
| Source/NVIDIAGfeSDK     | First commit | a day ago |
| ThirdParty/NVIDIAGfeSDK | First commit | a day ago |
| NVIDIAGfeSDK.uplugin    | First commit | a day ago |
| README.md               | First commit | a day ago |

README.md

## NVIDIA Shadowplay Highlights plugin

- GFE SDK Version: 1.1.173.5d889305
- GeForce Experience minimum version: 3.8
- Plugin version: 0.2



# Also in Unity

<https://assetstore.unity.com/packages/tools/video/nvidia-highlights-100938>

1. Search for “NVIDIA Highlights” on the asset store
2. Download & Import into current project
3. Access Highlights C# functions through `NVIDIA.Highlights` namespace



The screenshot shows the Unity Asset Store page for "NVIDIA Highlights" by NVIDIA CORPORATION. The main image features the text "NVIDIA HIGHLIGHTS" and "GEFORCE EXPERIENCE" over a game scene. To the left is a vertical strip of five small game highlights. The page includes a "FREE" label, an "Add to My Assets" button, and a description of the feature. A "REQUIRES" section lists system requirements. At the bottom, there are tabs for "Package contents", "Releases", and "Supported Unity versions", along with "Share" and "Add to List" buttons.

**NVIDIA CORPORATION**  
**NVIDIA® Highlights**  
(not enough ratings)

**FREE**  
[Add to My Assets](#)

**NVIDIA Highlights** is a feature of GeForce Experience that enables automatic video capture of key moments, clutch kills, and match-winning plays, ensuring gamers' best gaming moments are always saved. Once a Highlight is captured, gamers can upload their favorite highlights as videos, GIFs, or screenshots directly to Facebook, YouTube, Weibo right from GeForce Experience's in-game overlay. This popular feature has been integrated into top games such as PLAYERUNKNOWN'S BATTLEGROUNDS, Final Fantasy XV, and Call of Duty: World War II. Try out this popular capture tool today!

**REQUIRES**  
-Windows PC with Windows 7 (64-bit) or newer  
-GeForce GTX 600 series or newer  
-Shadowplay-ready display driver. Any NVIDIA display driver of version 368.81 or higher will meet this requirement.  
-Unity version 5.6+  
*GeForce Experience v 3.9 or above*

**Package contents** 2.2 MB  
**Releases** current ver. 1.0 [Show More](#)  
**Supported Unity versions** 5.6.0 or higher

[Support website](#) [Publisher website](#) [Share](#) [Add to List](#) [Report this asset](#)

# About the Engine Plugins

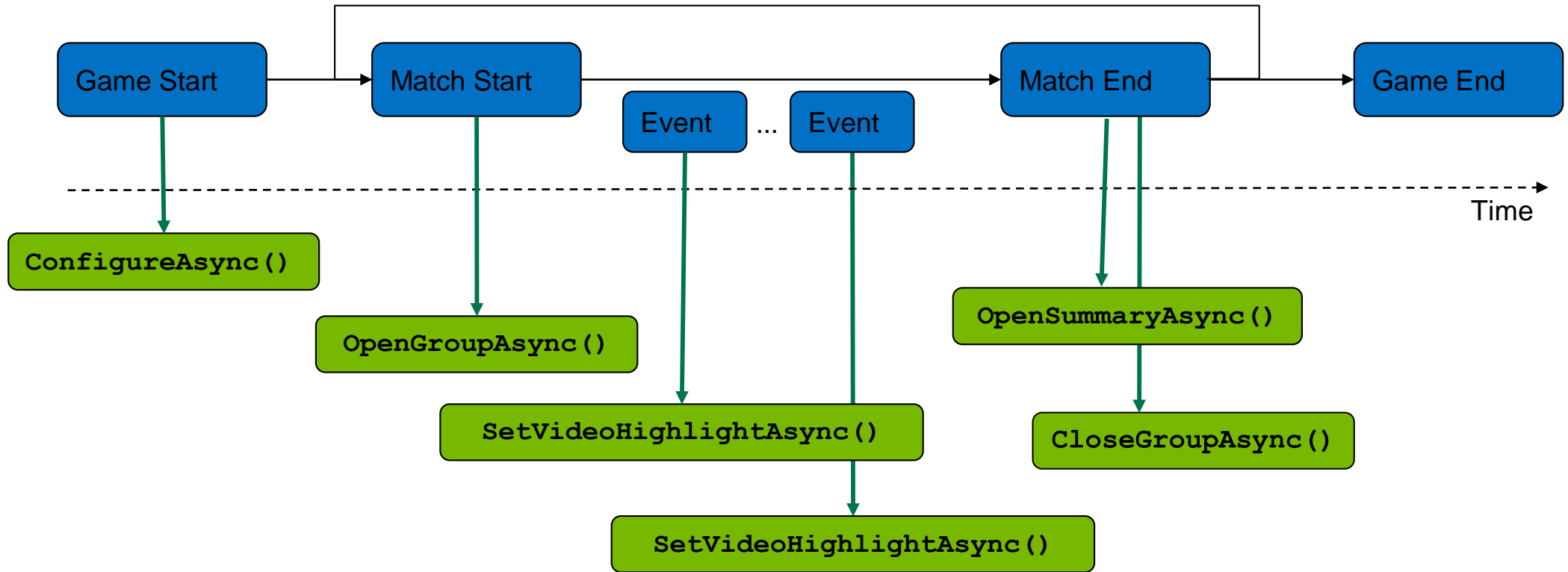
- Unreal Engine 4
  - C++ and Blueprint support
  - Currently Unreal Engine 4.18
  - <https://github.com/NVIDIAGameWorks/GfeSDK-UE4Plugin>
- Unity
  - Provides C# interfaces
  - Unity 5.6(Basic or Pro) or Unity 2017
    - Unity 2017 required for sample projects
  - <https://www.assetstore.unity3d.com/en/#!/preview/100938/302229>

# Custom Integration Overview

- Include “gfesdk/isdk.h”
  - Basic GFE SDK initialization
- Include “gfesdk/highlights/ihighlights.h”
  - Highlights specific functions
- Add GFESDK.dll into game package
  - At runtime load DLL in game process
    - Link via lib, or manually load, etc
    - Recommend perform Digital Signature Check to prevent malicious/rogue code
- Call various SDK interfaces we’ll explain in this talk.



# Example Highlights API Flow





# NVGSDK\_Create()

- Initializes the SDK interface with the requested highlight scopes
- Parameters:
  - Handle to the interface
  - Create input parameter structure
  - Response structure
- Success returns (more than one!):
  - NVGSDK\_SUCCESS
  - NVGSDK\_IPC\_SUCCESS\_OLD\_SDK
  - NVGSDK\_IPC\_SUCCESS\_OLD\_GFE

```
NVGSDK_Scope scopes[] = { NVGSDK_SCOPE_HIGHLIGHTS,  
NVGSDK_SCOPE_HIGHLIGHTS_VIDEO,  
NVGSDK_SCOPE_HIGHLIGHTS_SCREENSHOT };
```

```
NVGSDK_ScopePermission scopePermissions[3];  
NVGSDK_RetCode returnCode = NVGSDK_SUCCESS;  
NVGSDK_HANDLE * handle = INVALID_HANDLE;
```

```
NVGSDK_CreateInputParams createParams = {0};  
createParams.appName = "MyGame";  
createParams.scopeTable = &scopes[0];  
createParams.scopeTableSize = 3;  
createParams.pollForCallbacks = false;
```

```
NVGSDK_CreateResponse response;  
response.scopePermissionTable = &scopePermissions[0];  
response.scopePermissionsTableSize = 3;
```

```
returnCode = NVGSDK_Create(handle, createParams,  
response);  
if(NVGSDK_SUCCEEDED(returnCode)  
{  
    // Success path  
}  
else  
{  
    // Failure handling  
}
```

# NVGSDK\_ConfigureAsync()

- Provides a list of all possible Highlights to GFE
  - This is the point GFE knows about the highlights
- Parameters:
  - Handle to the interface
  - Table of highlights
  - Callback for response (can be NULL)
  - Context point for callback (can be NULL)
- No return code (async)

```
int const NUM_HIGHLIGHTS = 2;
NVGSDK_Highlight highlights[NUM_HIGHLIGHTS] = {0};

NVGSDK_LocalizedPair highlight1Translation[2] = {
    { "en-US", "Highlight One" }, { "es-MX", "Resalte Uno" }
};
highlights[0].id = "highlight1";
highlights[0].userInterest = true;
highlights[0].highlightTags = NVGSDK_HIGHLIGHT_TYPE_ACHIEVEMENT;
highlights[0].significance = NVGSDK_HIGHLIGHT_SIGNIFICANCE_VERY_GOOD;
highlights[0].nameTable = &highlight1Translation[0];
highlights[0].nameTableSize = 2;

NVGSDK_LocalizedPair highlight2Translation[2] = {
    { "en-US", "Highlight Two" }, { "es-MX", "Resalte Dos" }
};
highlights[1].id = "highlight2";
highlights[1].userInterest = true;
highlights[1].highlightTags = NVGSDK_HIGHLIGHT_TYPE_MILESTONE;
highlights[1].significance = NVGSDK_HIGHLIGHT_SIGNIFICANCE_NEUTRAL;
highlights[1].nameTable = &highlight2Translation[0];
highlights[1].nameTableSize = 2;

NVGSDK_HighlightConfigParams params = {0};
params.highlightDefinitionTable = &highlights[0];
params.highlightTableSize = NUM_HIGHLIGHTS;
params.defaultLocale = "en-US";

returnCode = NVGSDK_Highlights_ConfigureAsync(handle, &params, NULL,
NULL);
if(NVGSDK_SUCCEEDED(returnCode)
{
    // Success path
}
else
{
    // Failure handling
}
```

# NVGSDK\_OpenGroupAsync()

- Opens a group container to put highlights into
- Parameters:
  - Handle to the interface
  - OpenGroupParams structure
  - Callback for response (can be NULL)
  - Context point for callback (can be NULL)
- No return code (async)

```
NVGSDK_HighlightOpenGroupParams params = {0};
params.groupId = "group1";
params.groupDescriptionTable = {
    { "en-US", "Group One" }, { "es-MX", "Groupa Uno" }
};
params.groupDescriptionTableSize = 2;
returnCode = NVGSDK_Highlights_OpenGroupAsync(handle, &params, NULL,
NULL);
if(NVGSDK_SUCCEEDED(returnCode)
{
    // Success path
}
else
{
    // Failure handling
}
```

## NVGSDK\_SetScreenshotHighlightAsync()

- Creates a screenshot highlight in a group
- Parameters:
  - Handle to the interface
  - Screenshot highlight params structure
    - Includes group info for working with multiple groups
  - Callback for response of result
  - Context point for callback
- No return code (async)

```
NVGSDK_ScreenshotHighlightParams params = {0};  
params.groupId = "Group1"  
params.highlightId = "WinShot";
```

```
NVGSDK_Highlights_SetScreenshotHighlightAsync(handle, params,  
callback, context);
```



## NVGSDK\_SetVideoHighlightAsync()

- Creates a video highlight in a group
- Parameters:
  - Handle to the interface
  - Video highlight params structure
    - Includes group info for working with multiple groups
    - Sets start and end times of video relative to time of function call
  - Callback for response of result
  - Context point for callback
- No return code (async)

```
NVGSDK_VideoHighlightParams params = {0};  
params.groupId = "Group1";  
params.highlightId = "QuadKillHighlight";  
params.startDelta = -2000;           // start 2 seconds ago  
params.endDelta = 1000;             // end 1 second from now
```

```
NVGSDK_Highlights_SetVideoHighlightAsync(handle, params, callback,  
context);
```

3c4c-04cd6d419eb01bd3862a23-079

owjejb g1113

nvidiagfe001

#1

Victory Royale!



Win saved



✓ 0:59 1 1

HIGHLIGHTS

FORTNITE  
BATTLE  
ROYALE

View Match Stats

Previous Teammate

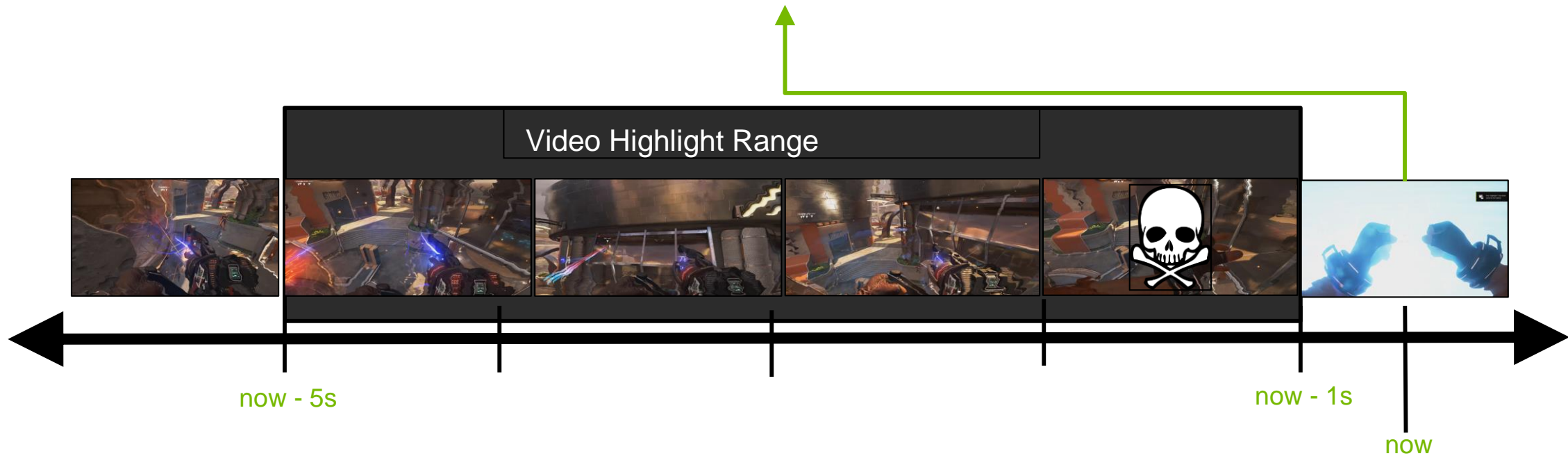
Next Teammate

Report Player

Return To Lobby

# Capture Example Timeline

```
NVGSDK_Highlights_SetVideoHighlight  
( hSDK, "5v5Fight", {"kill", startDelta = -5000, endDelta = -1000} );
```



# NVGSDK\_OpenSummaryAsync()

- Requests GFE to open the summary dialog for highlight(s) in group(s)
  - Will not open if no highlights meet criteria
- Parameters:
  - Handle to the interface
  - Summary structure
    - Includes group and filter data
  - Callback for response (can be NULL)
  - Context point for callback (can be NULL)
- No return codes (async)
- Should Pause game or ignore inputs while summary window is open

```
NVGSDK_GroupView groupView = {0};
groupView.groupId = "Group1";
groupView.tagsFilter = NVGSDK_HIGHLIGHT_TYPE_ACHIEVEMENT;
groupView.significanceFilter = NVGSDK_HIGHLIGHT_SIGNIFICANCE_GOOD;

NVGSDK_SummaryParams params = {0};
params.groupSummaryTable = &groupView;
params.groupSummaryTableSize = 1;

NVGSDK_Highlights_OpenSummaryAsync(handle, &params, callback,
context);
```





WINNER WIN

TEAM RANK

Highlights



Destination:



Post as:



Nvtest NvTest

Title:

Highlights | Shot with GeForce GTX

34 / 120

Location:

My timeline

Audience:

Friends

Save to Gallery

Share

Done

| KILLS | REVIVES | DAMAGE |
|-------|---------|--------|
| 3     | 1       | 120    |
| 3     | 3       | 30     |
| 1     | 0       | 20     |
| 5     | 0       | 69     |

Exit to Lobby

# UPLOAD HIGHLIGHTS AS ANIMATED GIFS



FACEBOOK



GOOGLE PHOTOS



WEIBO



# Video Walkthrough of Highlights Flow



# NVGSDK\_CloseGroupAsync()

- Closes up a highlight group, potentially deleting any unsaved highlights
- Parameters:
  - Handle to the interface
  - Close Group structure
    - Includes the data on group to close
    - Includes boolean to decide if unsaved highlights should be deleted or left behind
  - Callback for response (can be NULL)
  - Context point for callback (can be NULL)
- No return code (async)

```
NVGSDK_HighlightCloseGroupParams params = {0};  
params.groupId = "Group1";  
params.destroyHighlights = false; // delete highlights when closed?  
  
NVGSDK_Highlights_CloseGroupAsync(handle, params, callback, context);
```



## NVGSDK\_GetNumberOfHighlightsAsync( )

- Retrieves number of highlights in a group
  - Useful to decide if a Summary button should be shown to the user
- Parameters:
  - Handle to the interface
  - Group View parameter structure
  - Callback for response (can be NULL)
  - Context point for callback (can be NULL)
- No return code (async)

```
NVGSDK_GroupView params = {0};  
params.groupId = "Group1";  
params.tagsFilter = NVGSDK_HIGHLIGHT_TYPE_ACHIEVEMENT;  
params.significanceFilter = NVGSDK_HIGHLIGHT_SIGNIFICANCE_GOOD;
```

```
NVGSDK_Highlights_GetNumberOfHighlightsAsync(handle, params,  
callback, context)
```

```
// Callback snippet  
GetNumberOfHighlightsResponse response;  
response.numHighlights = data->numberOfHighlights;
```

```
if(response.numHighlights > 0)  
{  
    // Show Summary button  
}  
else  
{  
    // No highlights of group, tags, and significance combo  
}
```

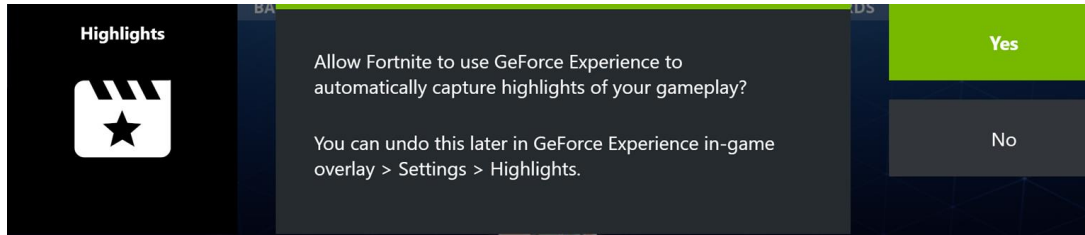
# Best Practices

- Highlights should be short/concise
  - Keeps drive space usage to a minimum
  - Good citizen in shared & size-controlled temp Highlights folder
  - Allows easy sharing to social media and limitations of social media
  - Prevents user from having to manually rework/trim the highlights

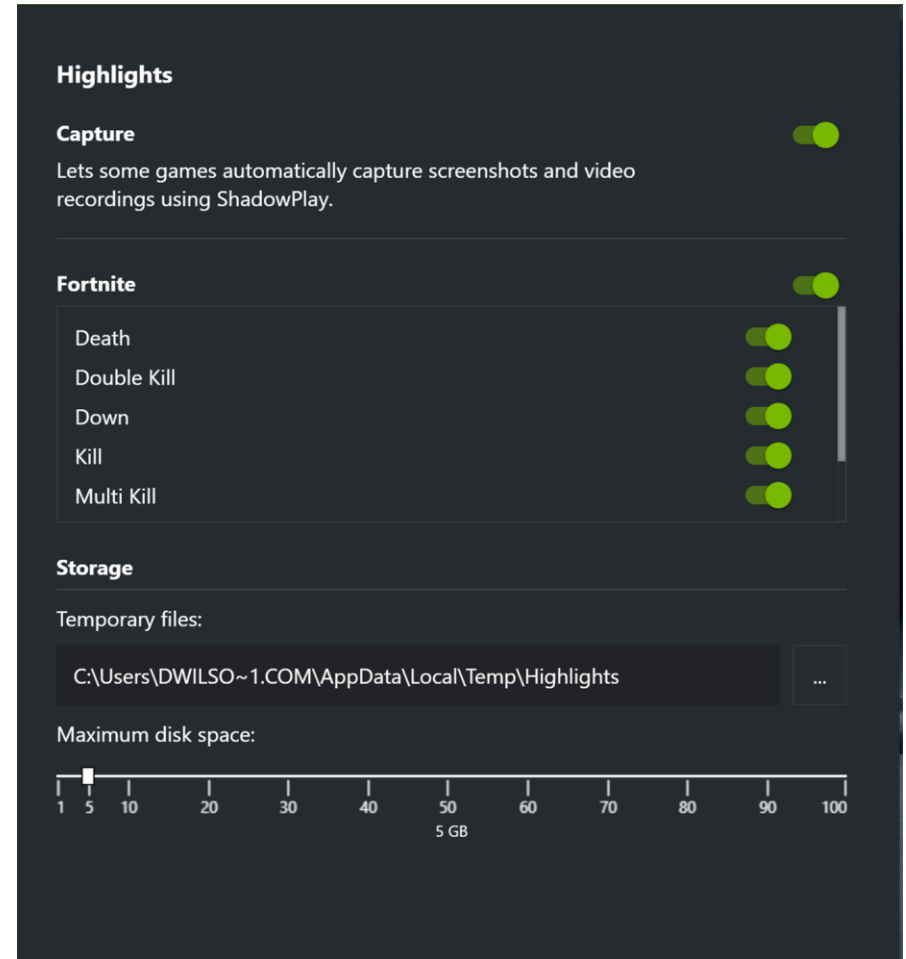


# Best Practices

- No need for an Highlights enable option in Game Settings
  - Permission dialog created on initial API call allows user to control feature state

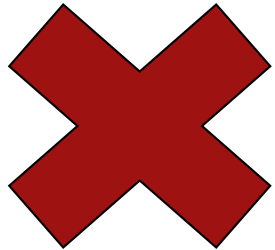


- Settings in Share UI allows user to control individual highlights



# Best Practices

- Limit Highlights to special events
- Find situations users would be proud to show off
- If something can be accomplished regularly by an average player, it is not a highlight



**Single Kill!**  
**Single Kill!**  
**Single Kill!**  
**Single Kill!**

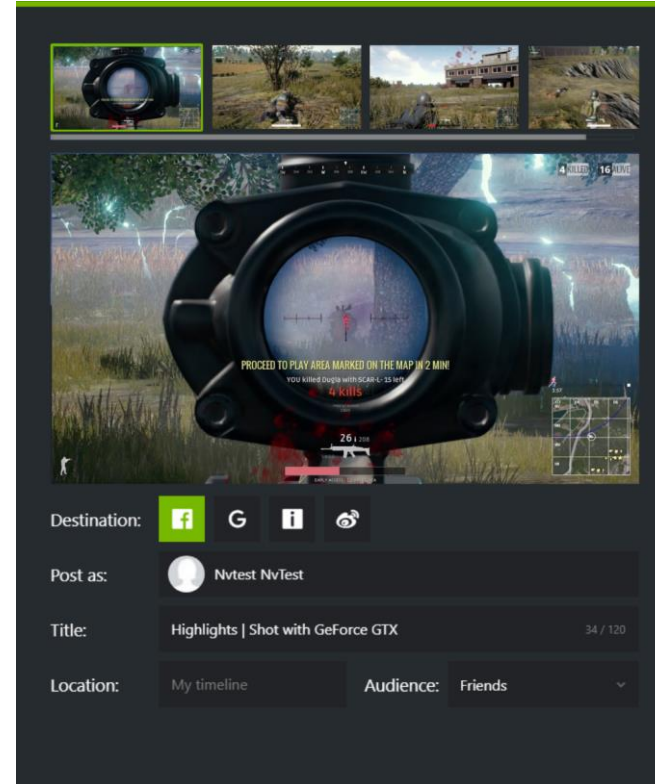


**Triple Kill!**  
**Revenge Kill!**  
**Long Range Kill!**  
**Last Kill!**



# Best Practices

- Keep groups logical with limited number of highlights
- Unique group per match or each level playthrough
- Limited highlights helps users when reviewing



# Best Practices

- Summary should be accessible after a match or level
- Limit invocation to when user has plenty of time to interact with the Highlights
  - Review
  - Trim
  - Upload to social media
- Do not automatically show it, provide a button/control to let user invoke
  - Follow NVIDIA control guidelines



Preferred Button  
on Black Background



Preferred Badge  
on White Background

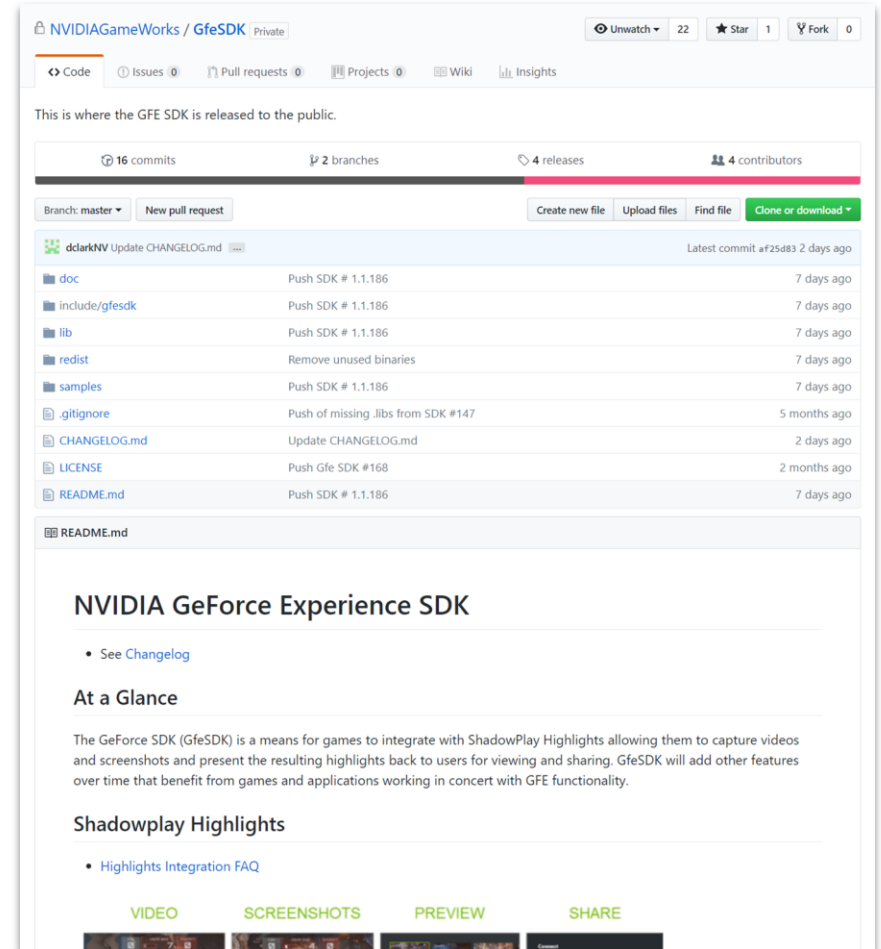
# Getting the Highlights SDK

Sign up for GameWorks access at:

<https://developer.nvidia.com/>

Then access the GitHub repro at:

<https://github.com/NVIDIAGameWorks/GfeSDK>



# Register your game!

<https://portal-developer.nvidia.com/#/signin>

- All unregistered games will list as “UNREGISTERED”.
  - Won’t save to your proper game name in the gallery
  - Will (at some point) contain a watermarking on all videos/shots
- Registration is simple and auto-approved.
  - Use your GFE credentials to sign in
  - Make an org(if you don’t have one already)
  - Add an application
- More detailed instructions on GIT docs at:  
<https://github.com/NVIDIAGameWorks/GfeSDK/blob/master/doc/DeveloperContentPortal.md>





# Questions?



Booth #223 - South Hall  
[www.nvidia.com/GDC](http://www.nvidia.com/GDC)

