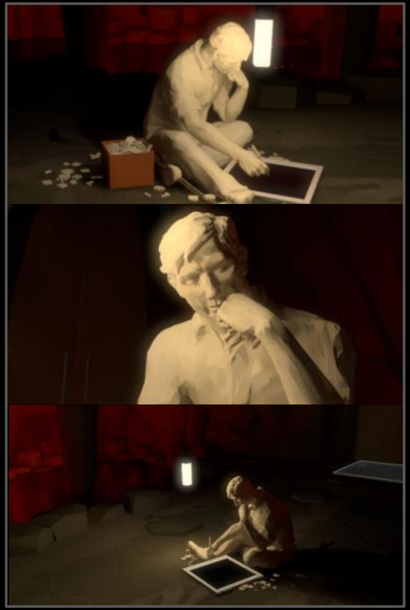
Adding photo mode to your game with NVIDIA Ansel

Halldor Fannar, 2017-03-02

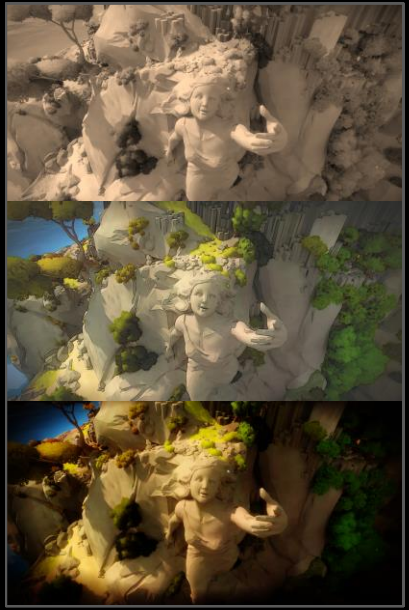
To cut a long story short: we built Ansel

- Standardized photo mode for all games running on GeForce
- Built into the display driver where all the heavy lifting is done
- Ansel photo mode is a participatory feature – each game must integrate a minimal SDK

Ansel takes in-game photography further

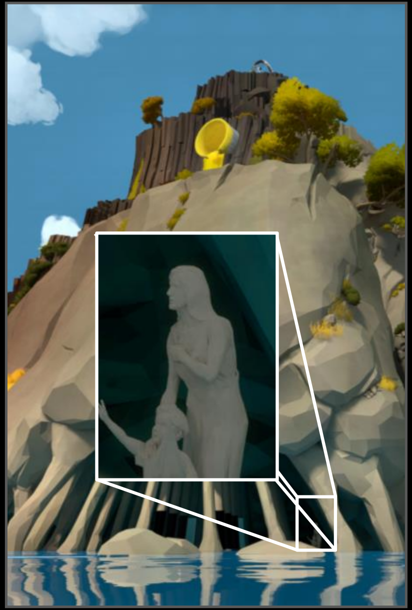


FREE CAMERA

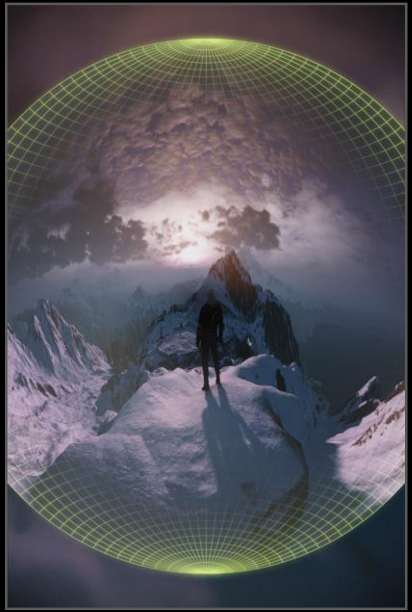


FILTERS

RAW



SUPER RESOLUTION



360

Want photo mode? We've got you covered

UNREAL ENGINE

UE 4.14+

unity

Unity Plugin

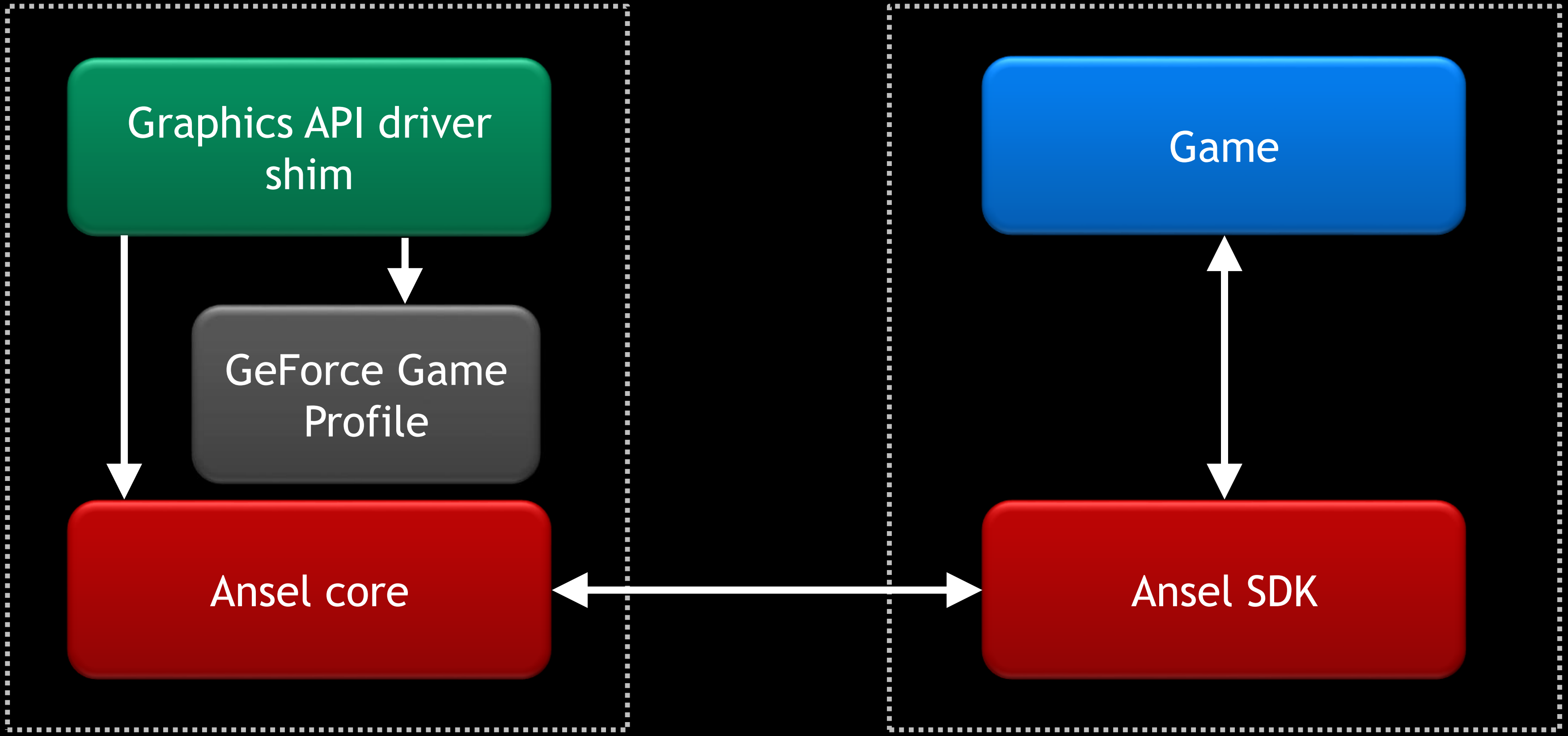
lumberyard BY amazon

Coming soon

NVIDIA ANSEL SDK

Gameworks Github

Ansel Architecture



Graphics API driver shim

GeForce Game Profile

Ansel core

Game

Ansel SDK

Ships with driver

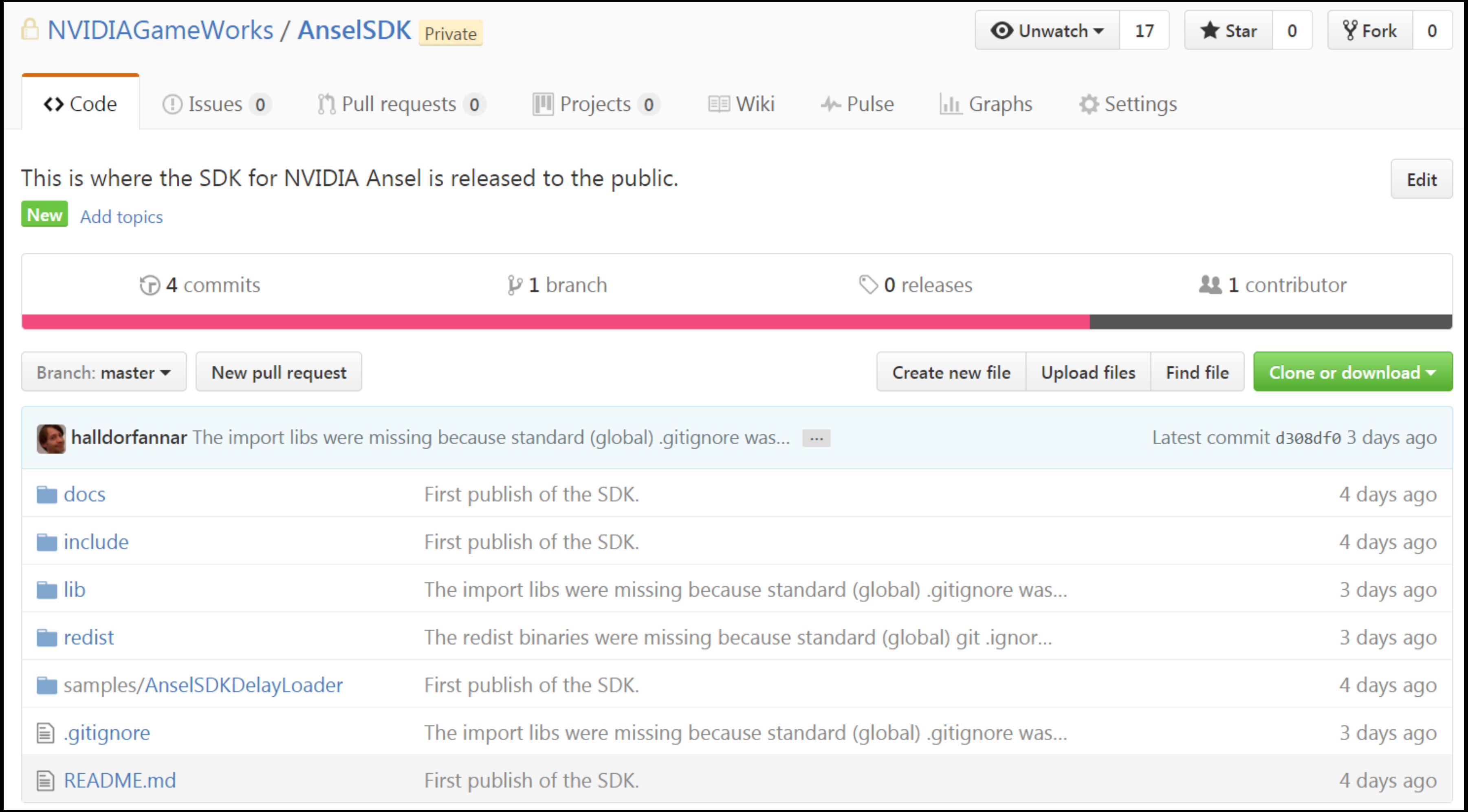
Ships with game

Join the GameWorks developer program

- 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: http://developer.nvidia.com/registered-developer-programs

- If you are logged in, accept the EULA and enter your GitHub username at the bottom of the form: http://developer.nvidia.com/content/apply-access-nvidia-gameworks-source-code

- You should receive an invitation within an hour https://github.com/NVIDIAGameWorks/AnselSDK



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

halldorfannar The import libs were missing because standard (global) .gitignore was… … Latest commit d308df0 3 days ago

docs	First publish of the SDK.	4 days ago
include	First publish of the SDK.	4 days ago
lib	The import libs were missing because standard (global) .gitignore was…	3 days ago
redist	The redist binaries were missing because standard (global) git .ignor…	3 days ago
samples/AnselSDKDelayLoader	First publish of the SDK.	4 days ago
.gitignore	The import libs were missing because standard (global) .gitignore was…	3 days ago
README.md	First publish of the SDK.	4 days ago

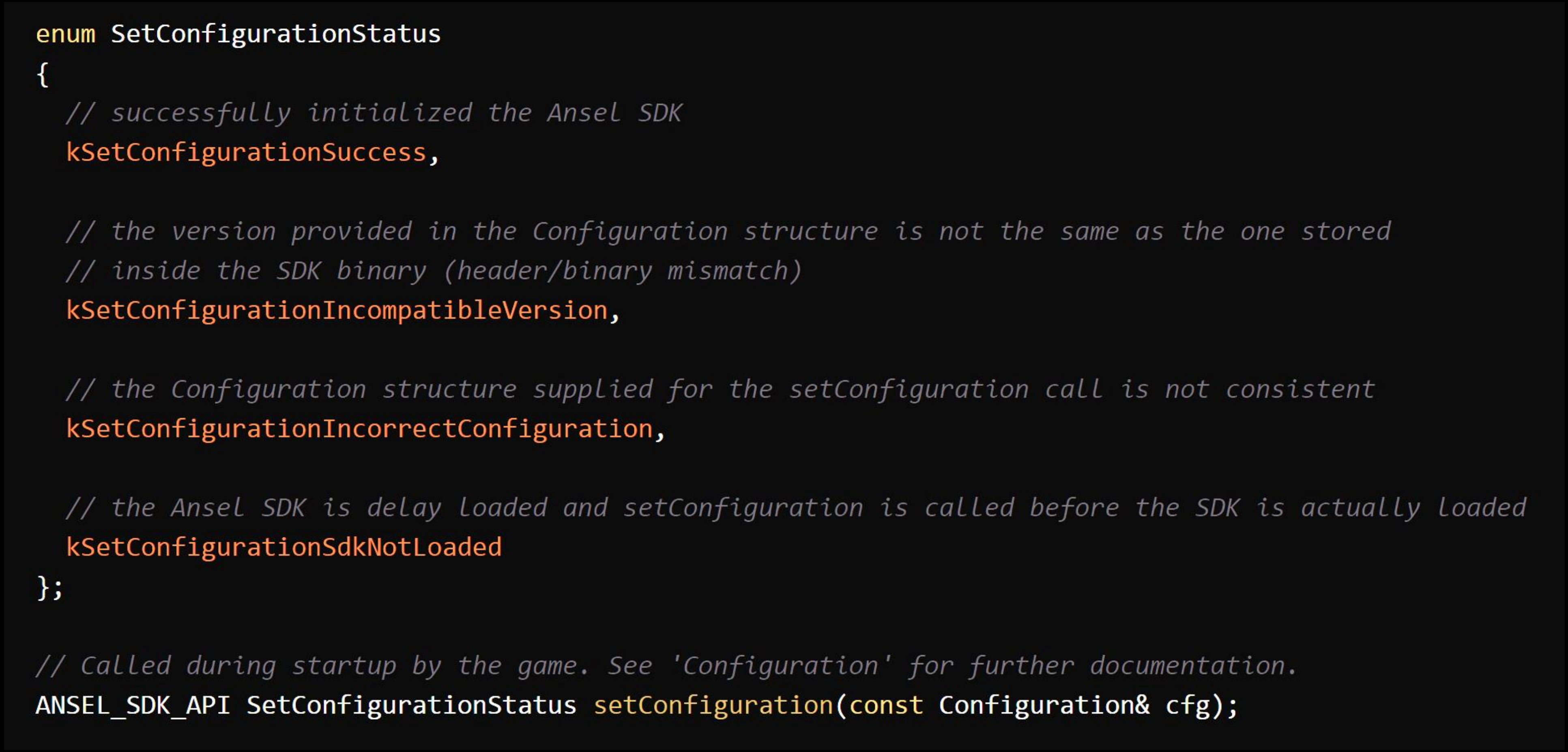
Agenda

- Walk through the integration of Ansel into a game
- Stop along the way to discuss 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

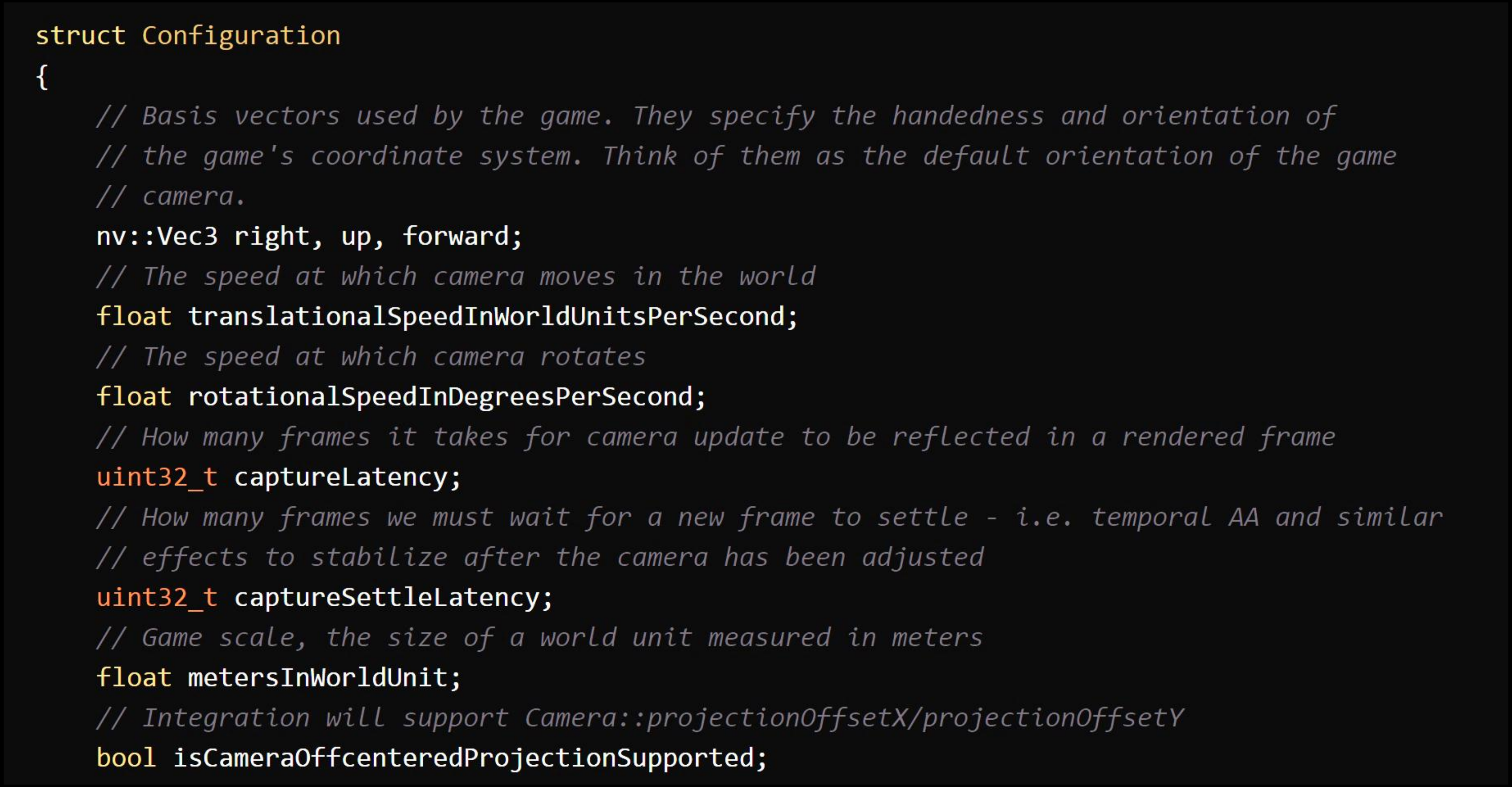
1. Configuration
2. Session
3. Camera
4. Hints (optional)

Setting the Configuration



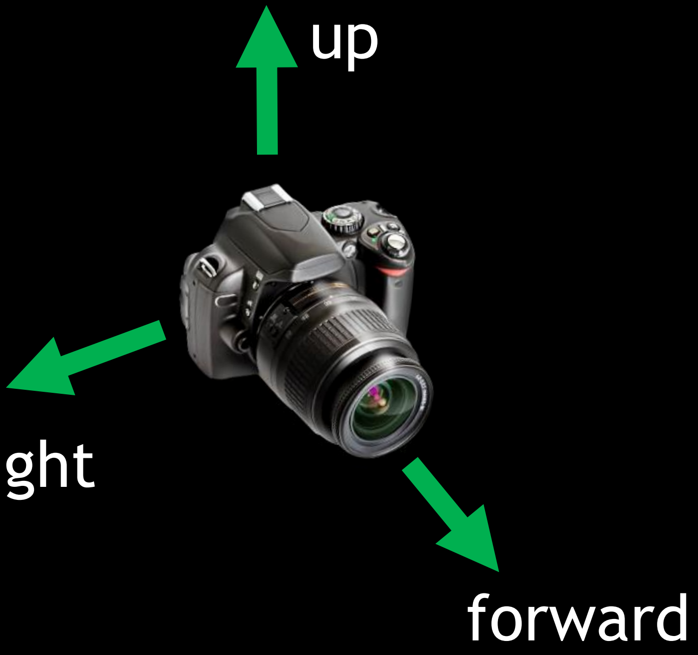
```
enum SetConfigurationStatus
{
  // successfully initialized the Ansel SDK
  kSetConfigurationSuccess,

  // 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);
```

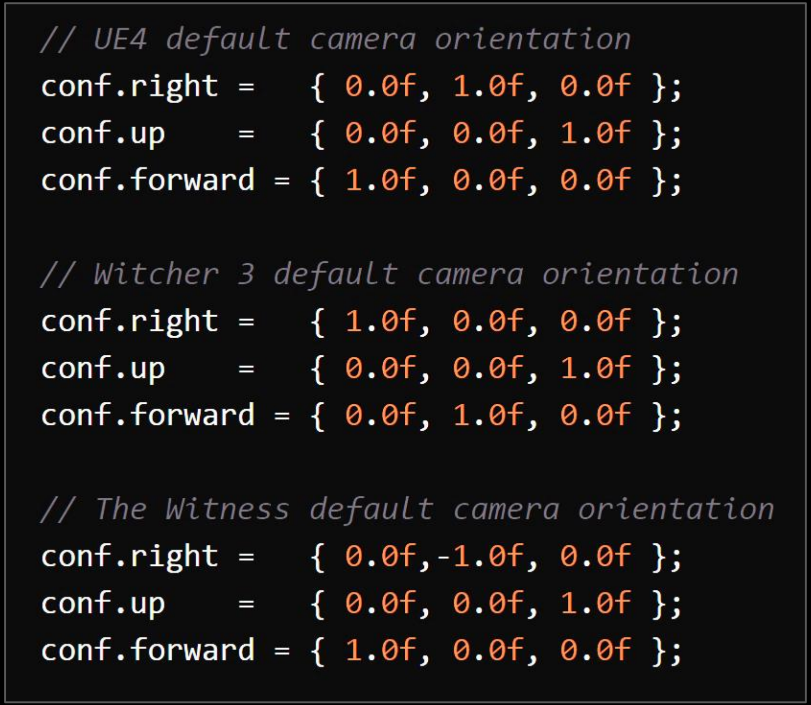
Configuration contents



```
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;
```

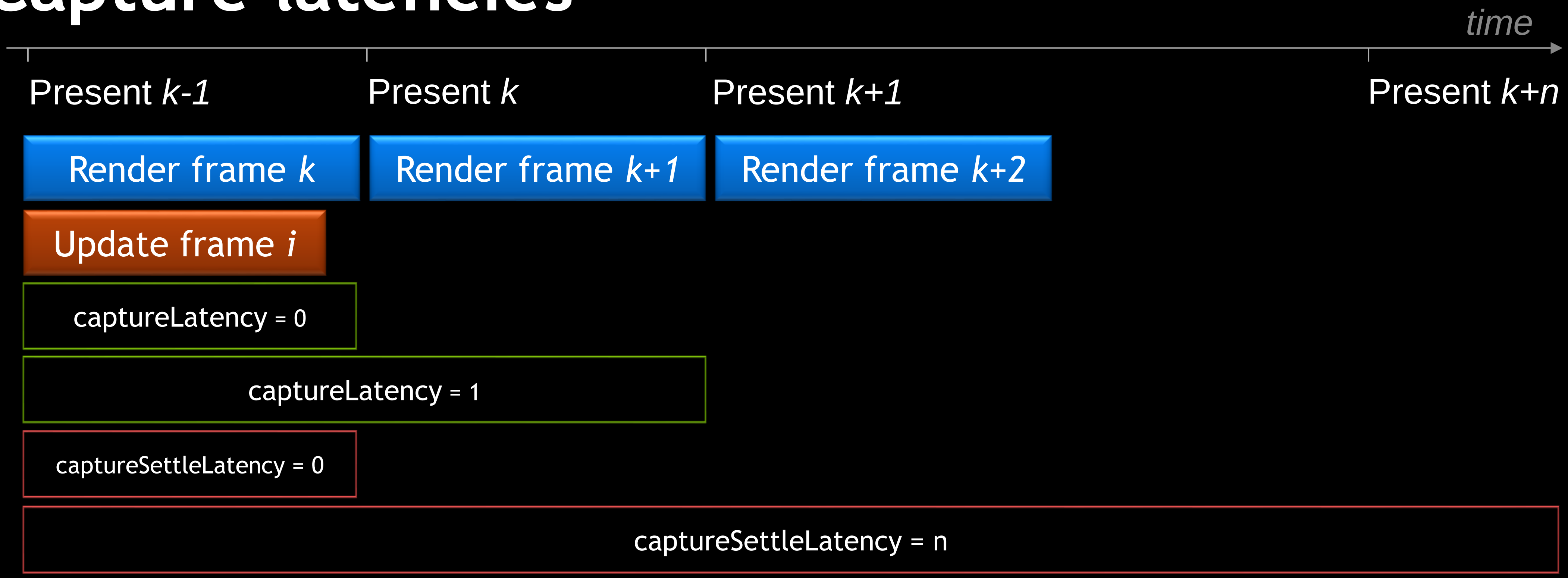
Default orientation in game coordinates





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

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

// The Witness default camera orientation
conf.right =   { 0.0f,-1.0f, 0.0f };
conf.up      =   { 0.0f, 0.0f, 1.0f };
conf.forward = { 1.0f, 0.0f, 0.0f };
```

Capture latencies

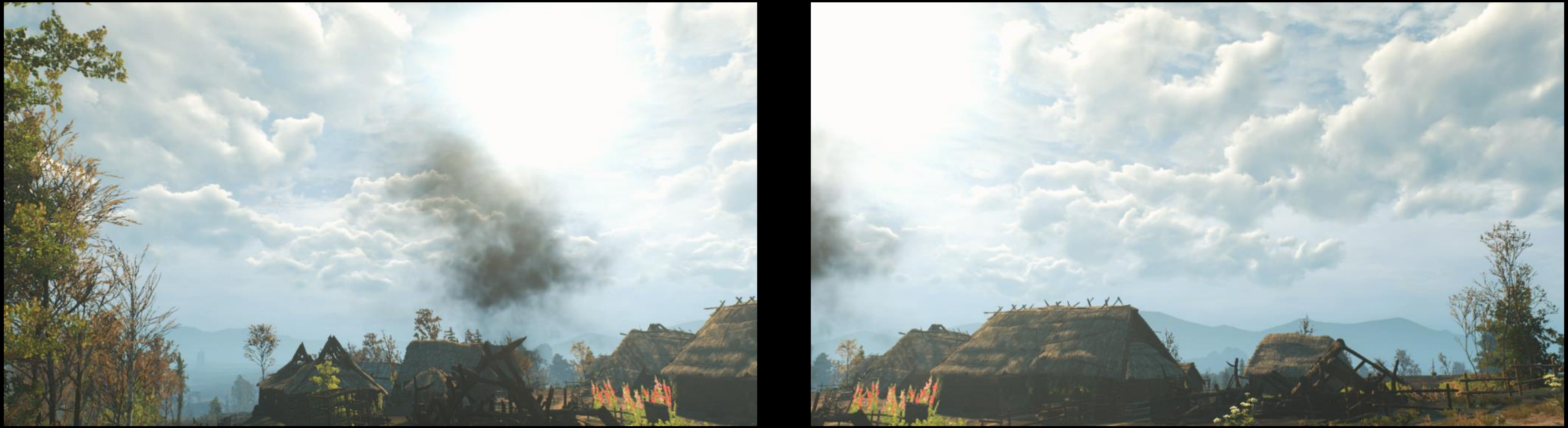


`captureLatency` is the number of D3D present calls between update and present for a frame

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

Discontinuous camera movement

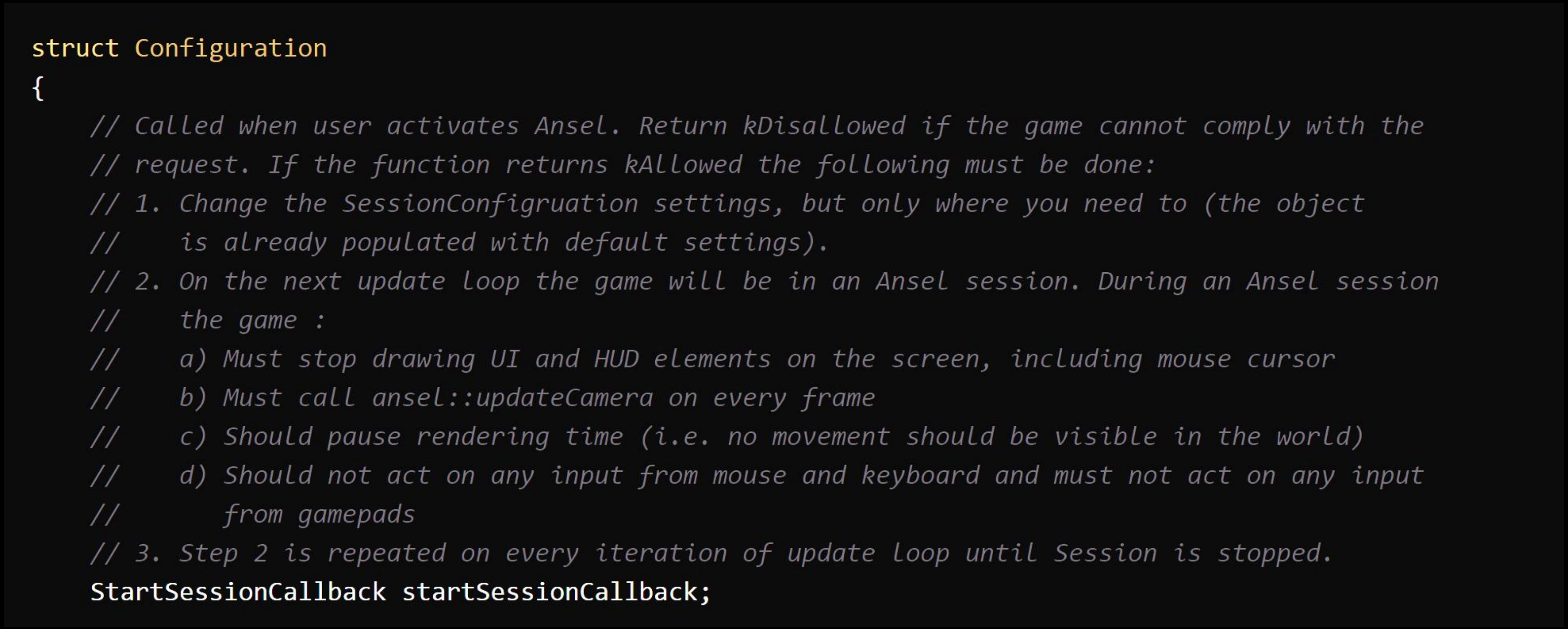
Two adjacent frames from a 360 capture (yaw angle for camera is changed in these shots):



- Frame accumulation effects need time to settle (via captureSettleLatency)
- Or disabled during multipart shots (more on this later)

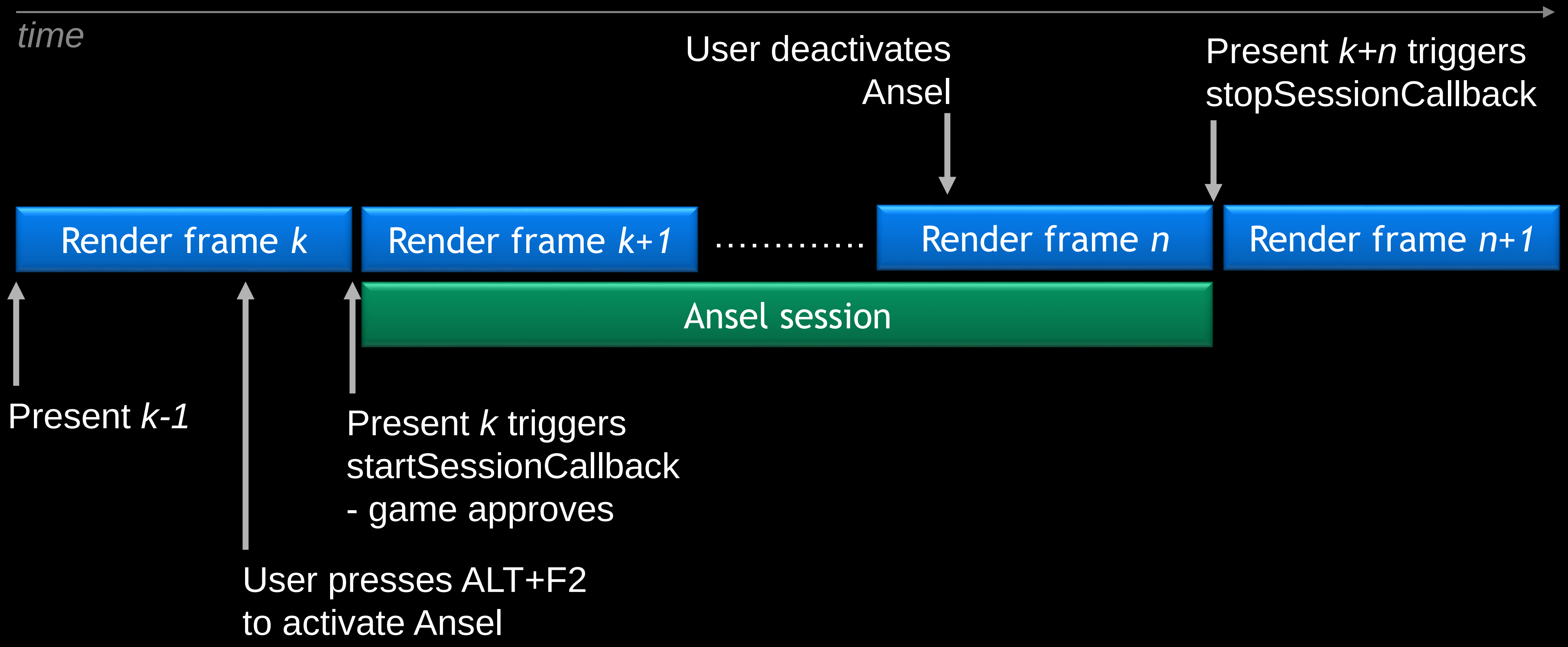
Session

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

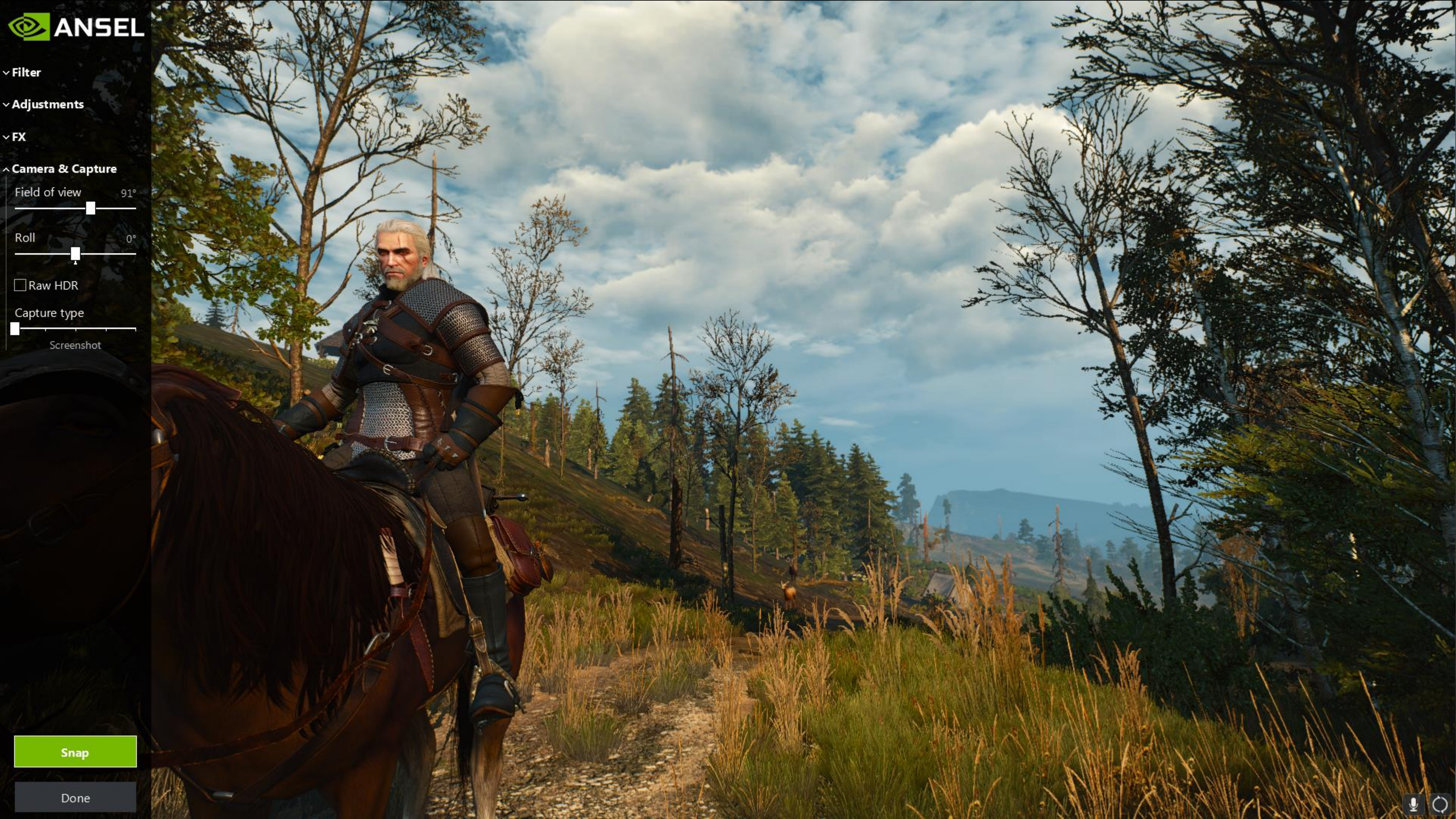


```
struct Configuration
{
    // Called when user activates Ansel. Return kDisallowed if the game cannot comply with the
    // request. If the function returns kAllowed the following must be done:
    // 1. Change the SessionConfigruation settings, but only where you need to (the object
    //    is 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



time
User deactivates Ansel
Present *k+n* triggers stopSessionCallback
Render frame *k*
Render frame *k+1*
....
Render frame *n*
Render frame *n+1*
Ansel session
Present *k-1*
Present *k* triggers startSessionCallback - game approves
User presses ALT+F2 to activate Ansel



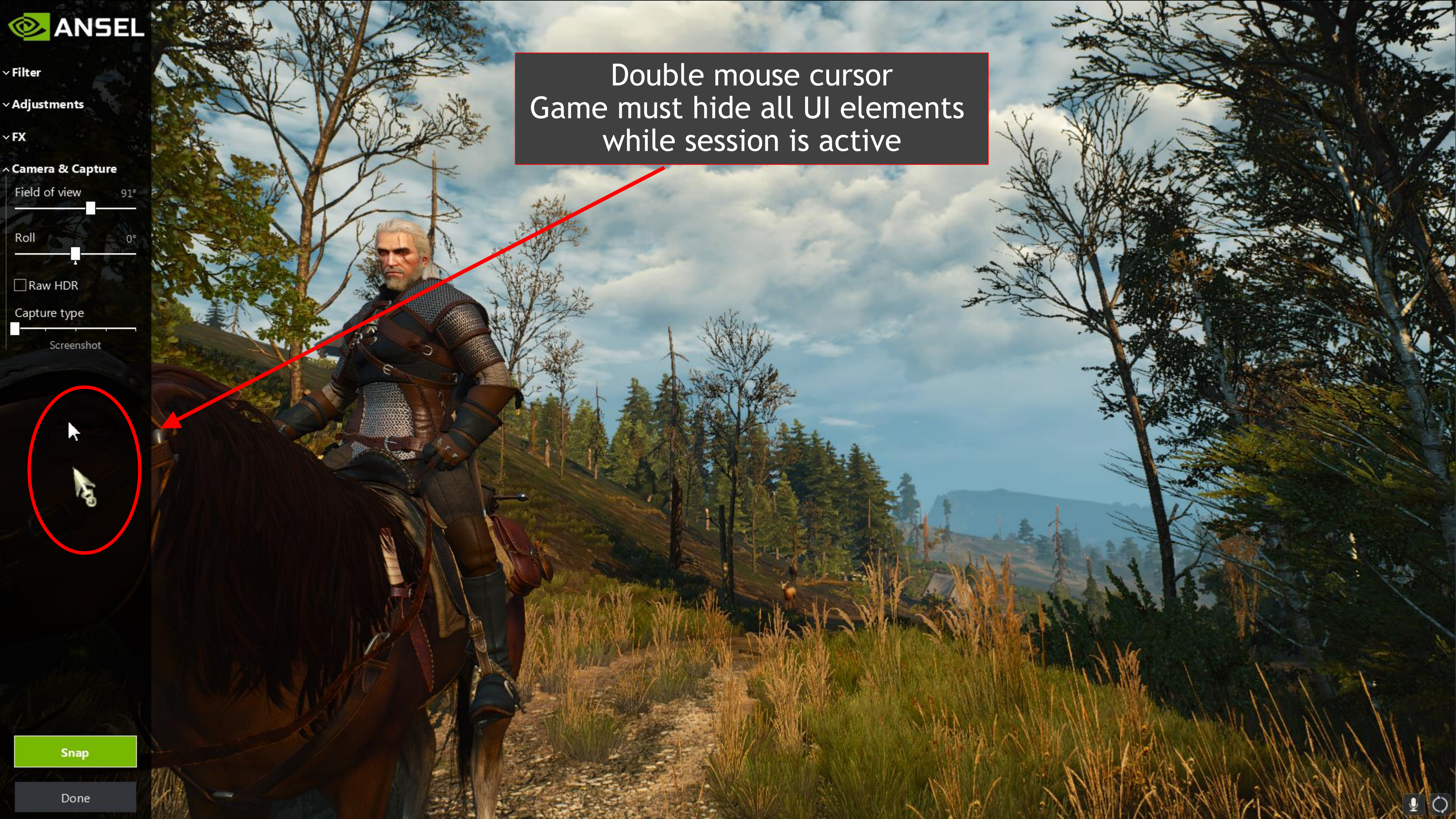
Field of view 91°

Roll 0°

□ Raw HDR

Capture type

Screenshot



Double mouse cursor
Game must hide all UI elements
while session is active

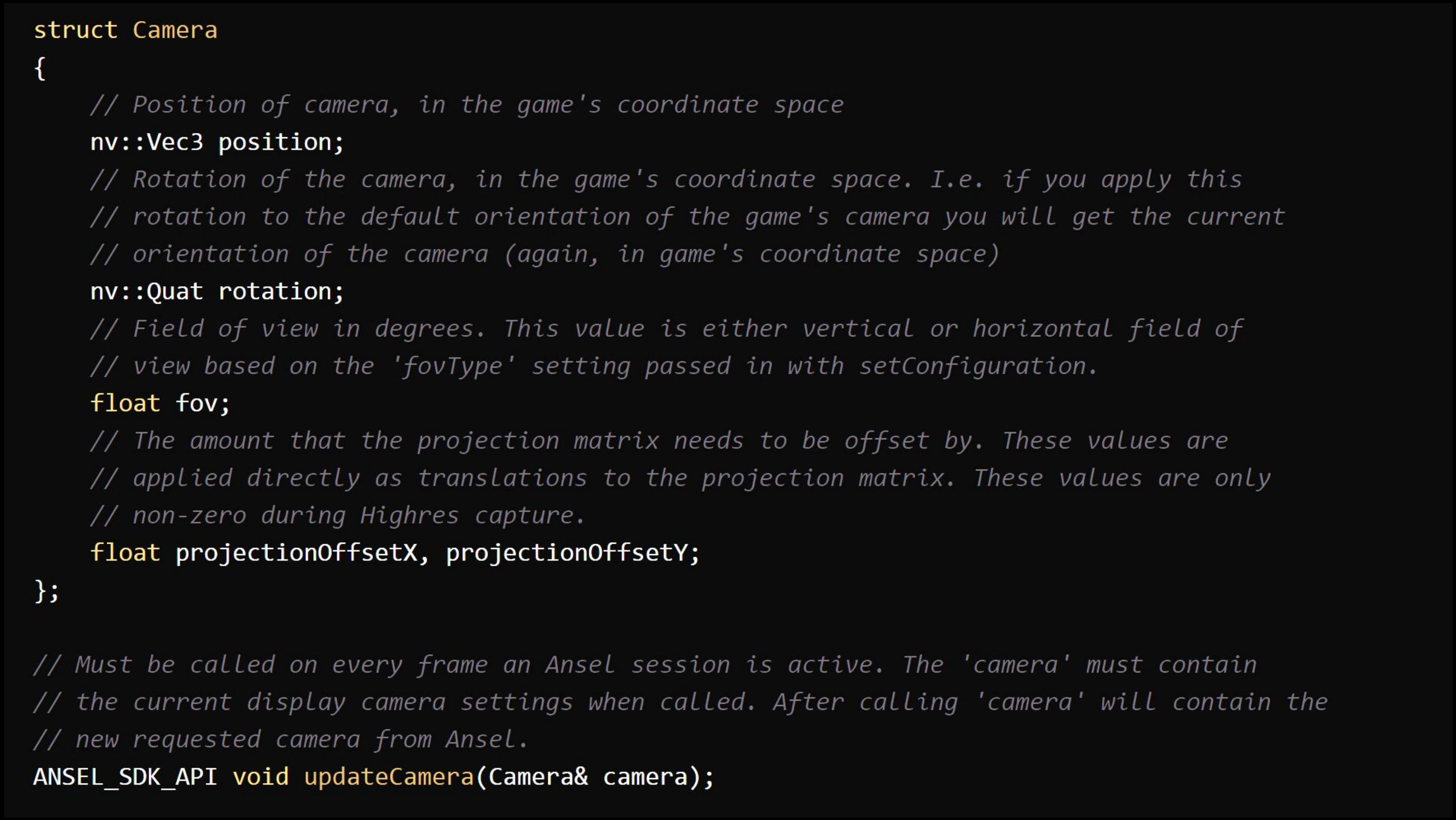
Field of view 91°

Roll 0°

□ Raw HDR

Capture type
Screenshot

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;
};

// 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

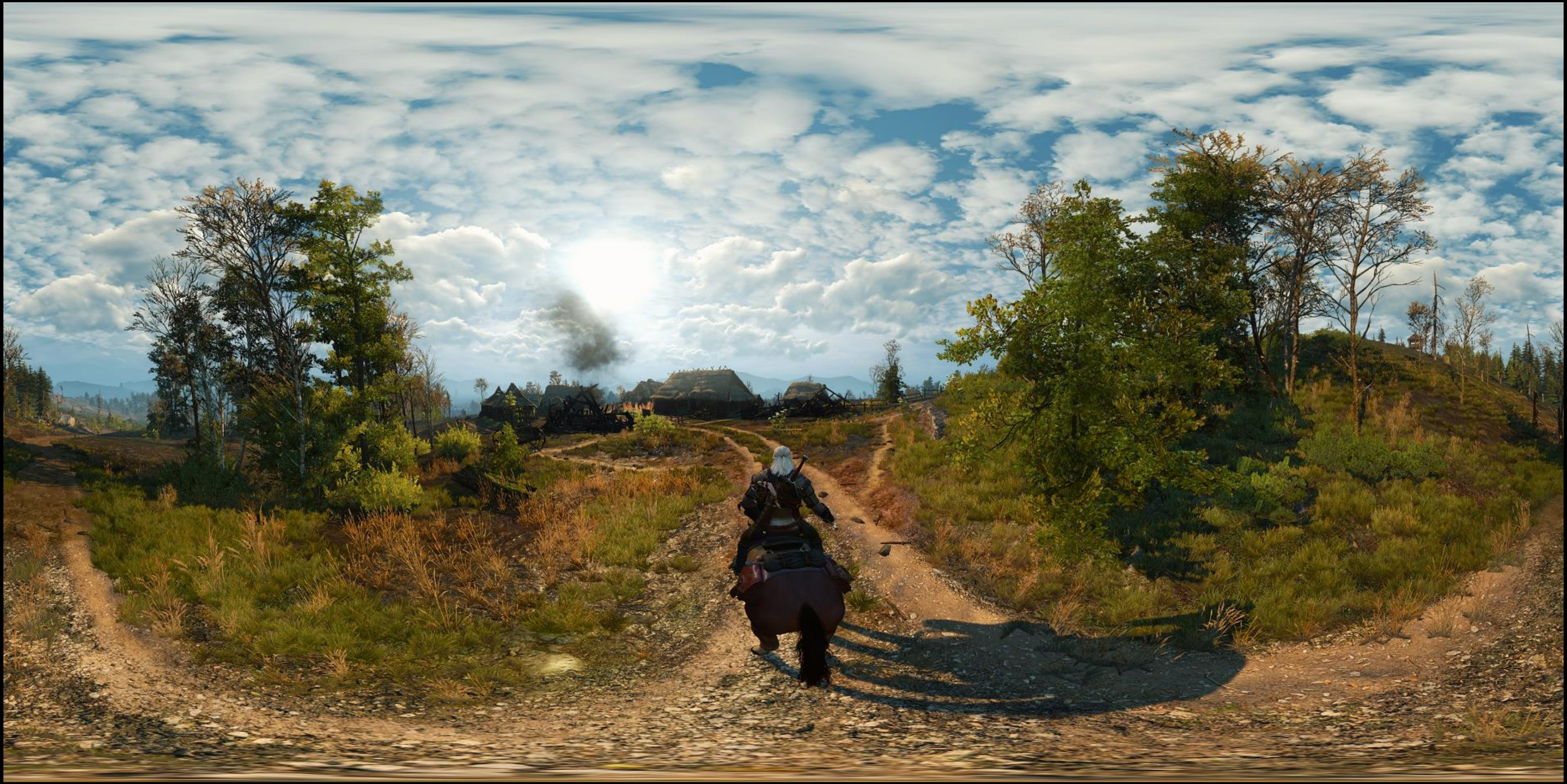
```
if (g_isAnselSessionActive)
{
  ansel::Camera cam;
  cam.fov = get_game_fov_degrees();
  cam.position = {game_cam_position.x, game_cam_position.y, game_cam_position.z};
  cam.rotation = {game_cam_orientation.x, game_cam_orientation.y, game_cam_orientation.z, game_cam_orientation.w};

  ansel::updateCamera(cam);
  // This is where a game would typically perform collision detection
  // and adjust the values requested by player in cam.position

  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_degrees(cam.fov);

  // modify projection matrices by the offset amounts
  offset_game_projection_matrices(cam.projectionOffsetX, cam.projectionOffsetY);
}
```

Projection offset for super resolution tiles

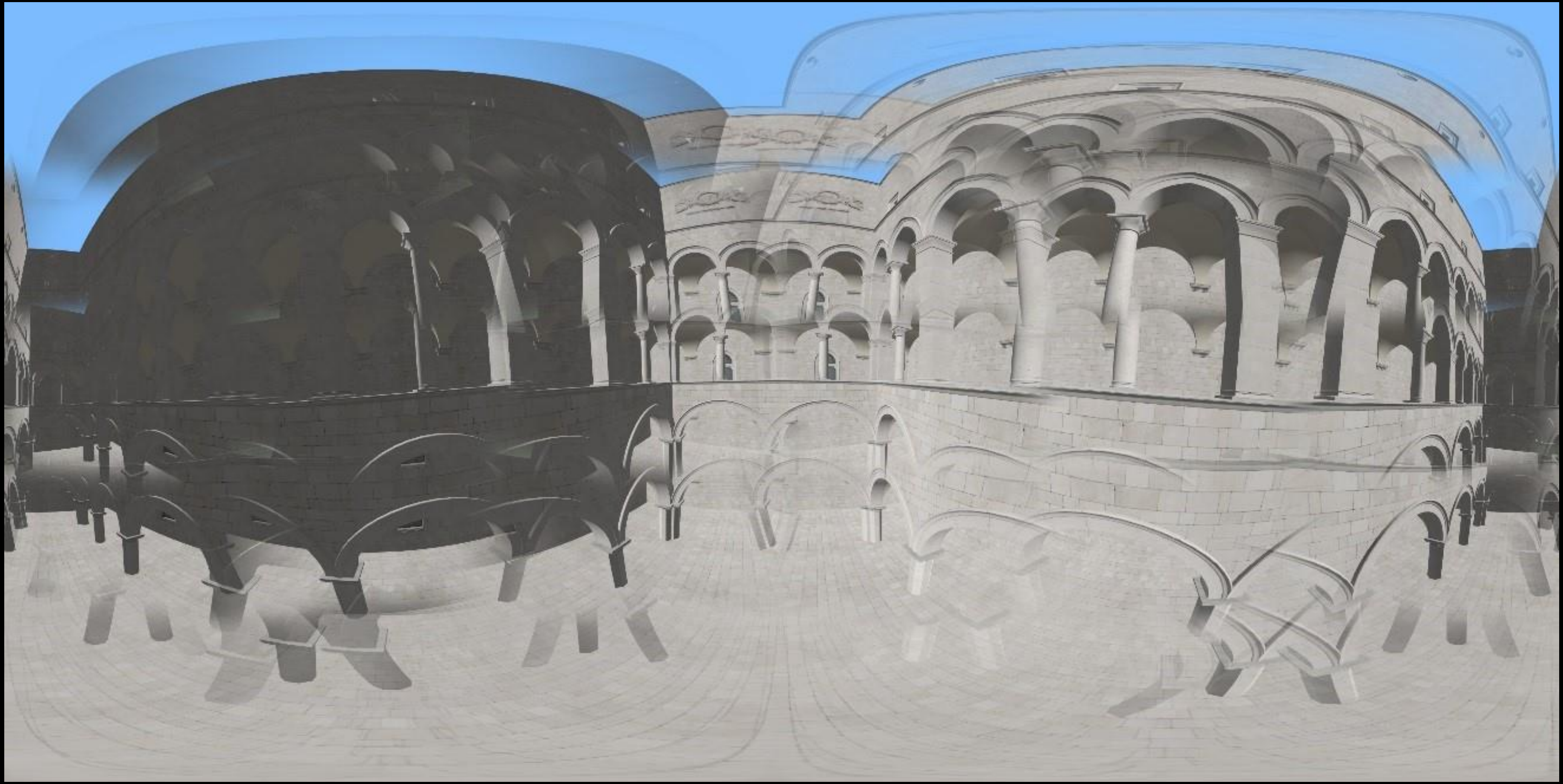
```
void offset_game_projection_matrices(float offsetX, float offsetY)
{
  // In this simple example we only need to modify the projection matrix associated with the game camera.
  // If the game is doing clever things like optimizing reflections or shadows based on projection matrix
  // then those code paths need to take a non-zero projection offset into account.

  // For nostalgia effect this game is using an old classic:
  D3DXMATRIX projection;
  D3DXMatrixPerspectiveFovRH(&projection, g_fov_radians, g_aspect, g_z_near, g_z_far)

  // Apply the offsets directly to the finished product (values are already normalized):
  projection._31 += offsetX;
  projection._32 += offsetY;

  // Update the games projection matrix:
  g_projection_matrix = projection;
}
```

Offset and view angle for Super resolution

Rotation and view angle for 360 photos

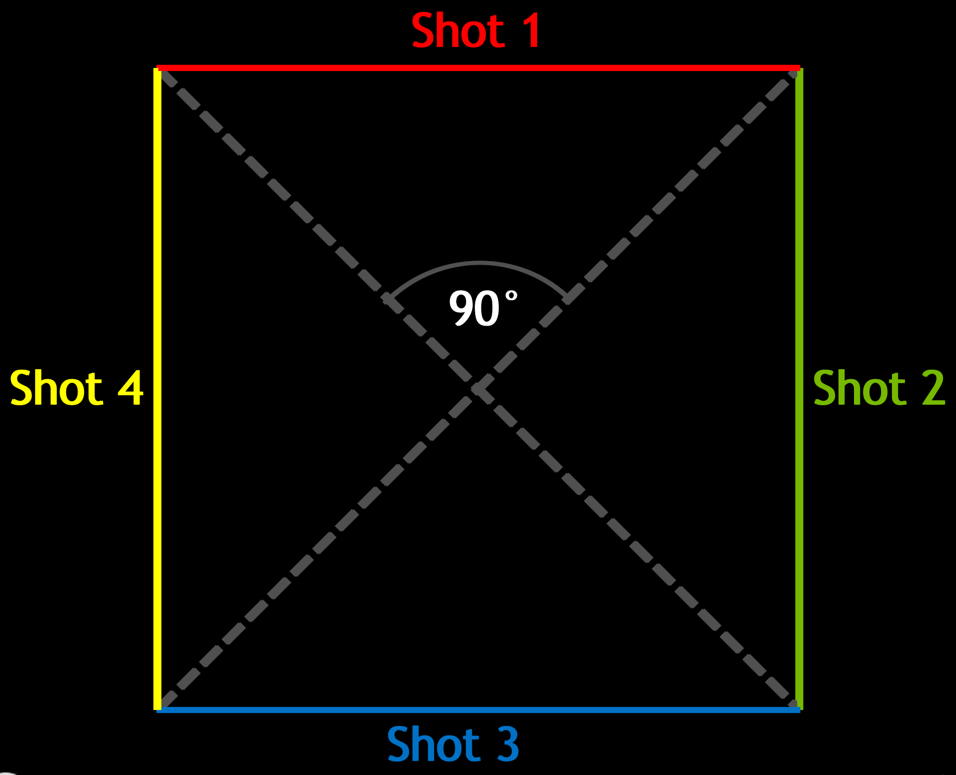


View angle incorrectly handled



Debugging view angle handling

- Set Field of View in Ansel UI to 90 degrees
- Then yaw the camera (mouse drag left/right or joystick left/right) and take a panorama covering a full circle => 4 shots



1	2	3	4	1

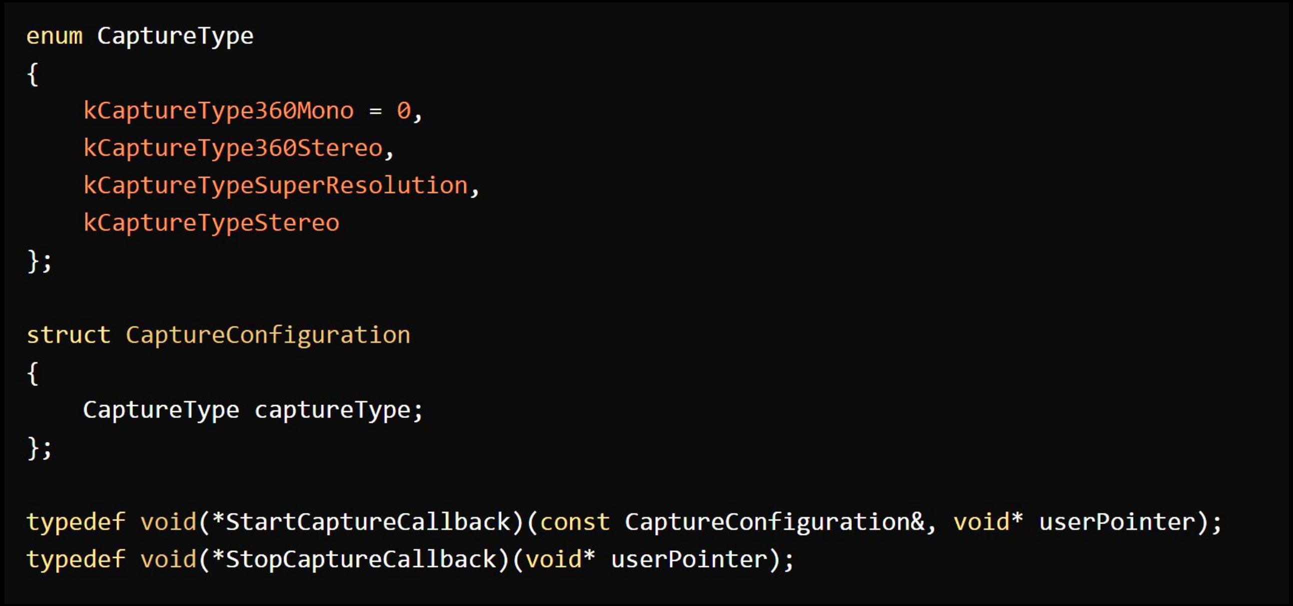
The shots should line up when placed like this

When the pictures don’t line up

- Verify that correct field of view type for the game was specified in `ansel::Configuration::fovType`
- Verify that degrees (not radians) are being used when interfacing with Ansel
- Verify that game’s projection matrix is using field of view and *frame buffer* aspect ratio for x & y pixel scaling and *nothing else*
  - If the game is using some additional pixel scaling to save on fill rate etc this can mean that the field of view is not what Ansel expects. This needs to be accounted for in conversion to/from `ansel::Camera::fov`.

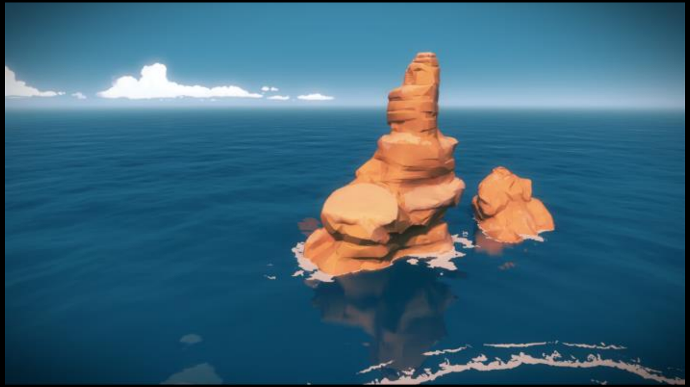
Post-effects and multipart shots

- 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

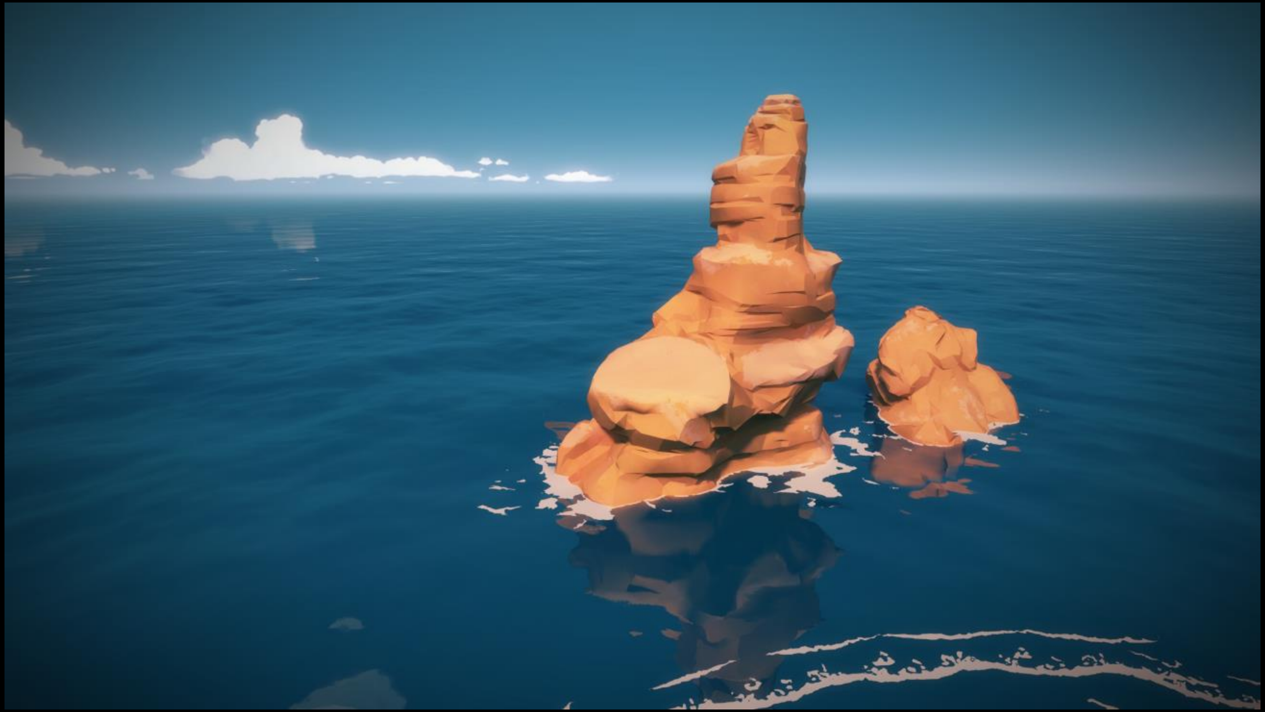


```
enum CaptureType
{
    kCaptureType360Mono = 0,
    kCaptureType360Stereo,
    kCaptureTypeSuperResolution,
    kCaptureTypeStereo
};

struct CaptureConfiguration
{
    CaptureType captureType;
};

typedef void(*StartCaptureCallback)(const CaptureConfiguration&, void* userPointer);
typedef void(*StopCaptureCallback)(void* userPointer);
```

Handling vignette correctly



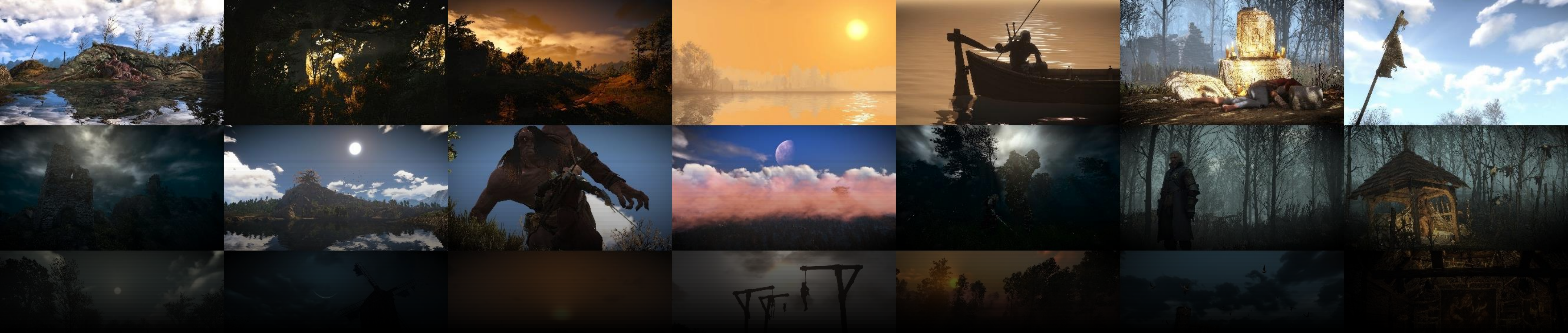
Regular shot (vignette active)



Super resolution shot (vignette disabled)
+ vignette applied by user via Ansel filters

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



THE WITCHER 3: WILD HUNT

200,000 ANSEL WORKS OF ART

WATCH DOGS 2

“ *It's a joy to witness what our players can create with Ansel and how easily it allows for high-quality, professional results* ”

FLORIN SANDA, UBISOFT PRODUCER

Photo by Treebsquire

WAR THUNDER

> *“When you see that Nvidia Ansel is added to the game. Life is complete.”*

MOTOR_STORM, WAR THUNDER GAMER

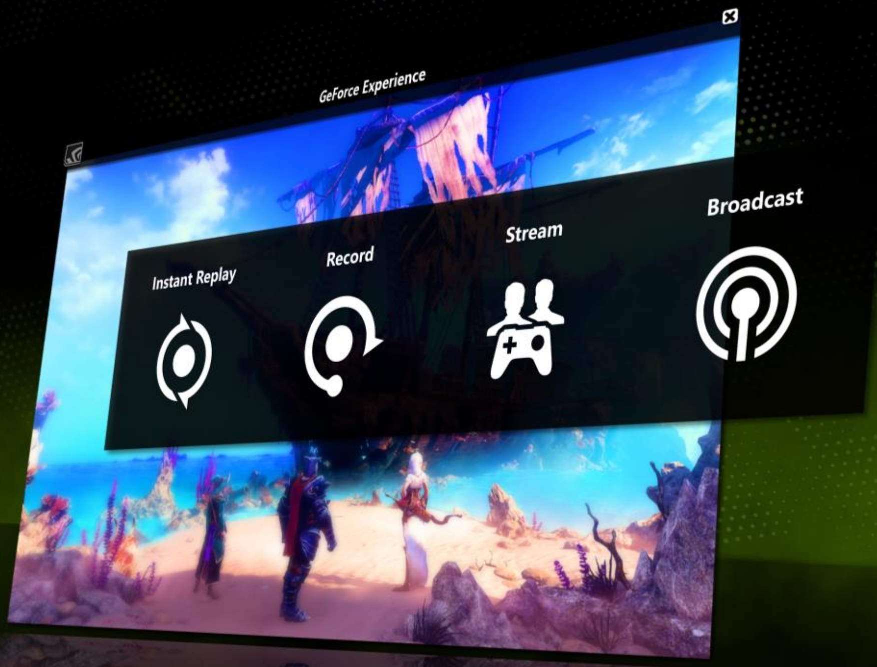
Photo by Vachtar

Capturing video and screenshots from game highlights without lifting a finger!

Malcolm Parmar, 2017-03-02

SHADOWPLAY

CAPTURE YOUR BEST GAMING MOMENTS



facebook YouTube Google imgur

200M

videos per year

2x

year over year growth

Boss Key’s experiments with ShadowPlay

“In games today it’s all about bragging rights and owning your moment.

Years ago we used ShadowPlay to a very unique way to capture players awesome gravity-defying clips and share them on social media.

Since, we've been collaborating with NVIDIA to turn this into a major feature and are excited about ShadowPlay Highlights, coming to LawBreakers first.”

CLIFF BLESZINSKI

CEO, Founder / Boss Key Productions

Desired functionality

Convenience

- I’m busy playing; my fingers and brain aren’t free to hit the record hot-key
- Can you just do it automatically for me?
- Plus, if I’m going to share something, I want it to be dead simple

Auto-curation

- I can’t waste time looking for the good stuff in a two hour recording
- Can you just record the cool moments and show them to me?

So we designed a highlights feature together...

Leveraging GeForce Experience’s Recording tech

- When something interesting happens, the game tells GeForce Experience to save a specific portion of gameplay as video or screenshot

Leveraging GeForce Experience’s Overlay

- After a session, game can tell GeForce Experience to display a summary of highlights for that session for the user to review and potentially share

...and made it non-invasive

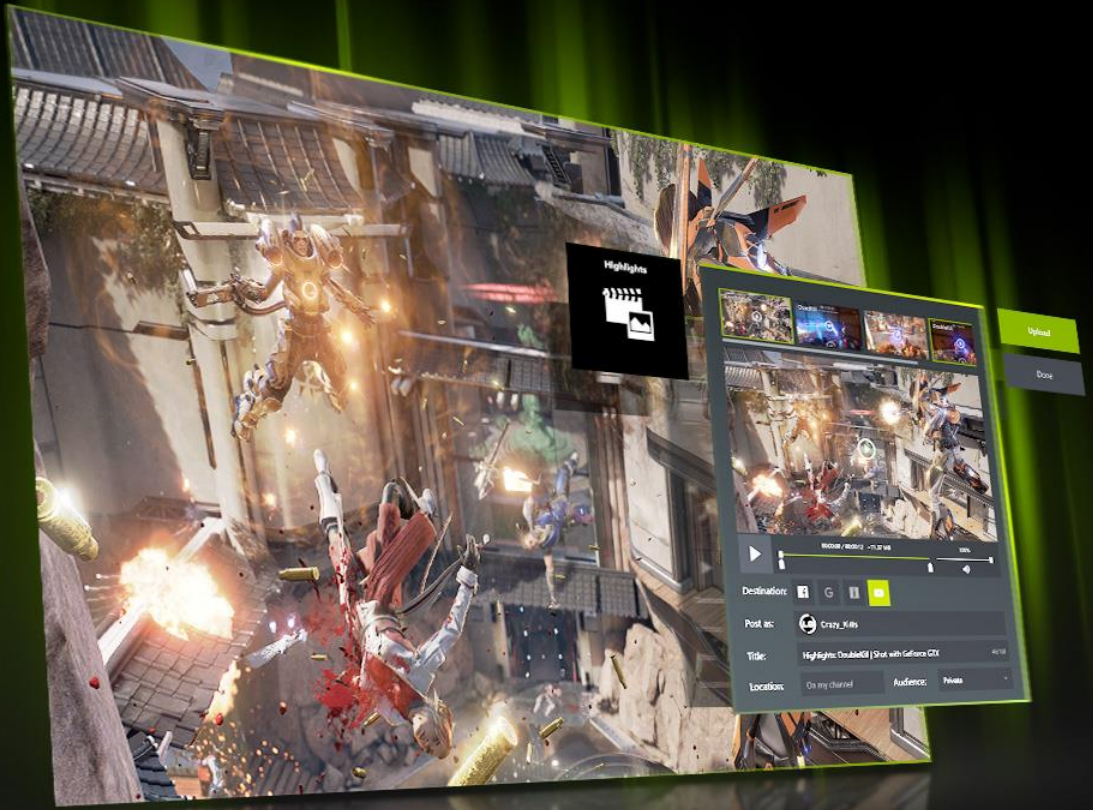
User can

- Control the type of highlights recorded per-game
- Review highlights *after the game* session or from the Gallery
- Elect to enter or skip summary via *game UI*
- Specify the amount of disk space devoted to highlights

Capture is low impact

- No game FPS drops
- Minimal system resource use

SHADOWPLAY HIGHLIGHTS



Destination:

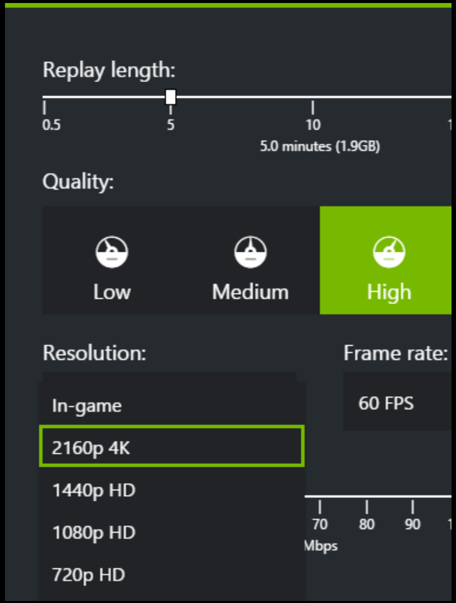
Post as: Crazy_Kills

Title: Highlights: DoubleKill | Shot with GeForce GTX

Location: On my channel

Audience: Private

CONFIGURATION (USER)



Replay length:

0.5 · 5 · 10

5.0 minutes (1.9GB)

Quality: Low · Medium · **High**

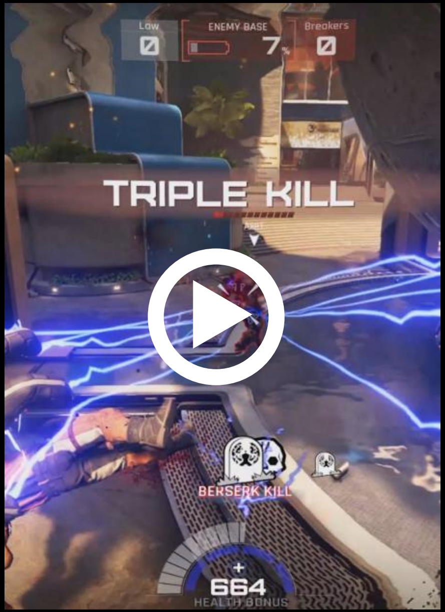
Resolution:
- In-game
- **2160p 4K**
- 1440p HD
- 1080p HD
- 720p HD

Frame rate: 60 FPS

70 · 80 · 90 Mbps

Video settings
Highlights settings
Notifications settings

VIDEO (GAME)

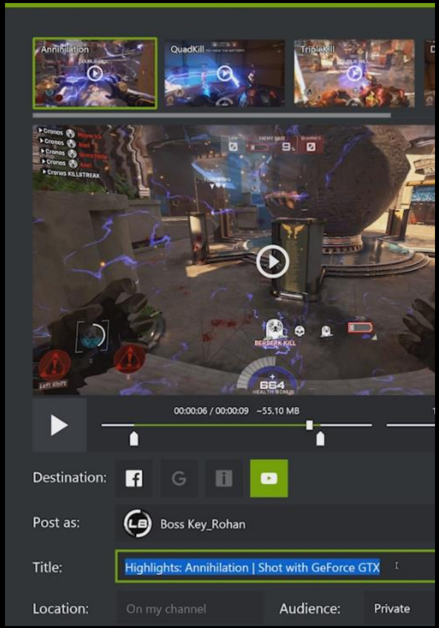


4K 60 FPS H.264
no impact to gameplay

SCREENSHOTS (GAME)

4K PNG images
no impact to gameplay

REVIEW (USER)



00:00:06 / 00:00:09 ~55.10 MB

Destination:

Post as: Boss Key_Rohan

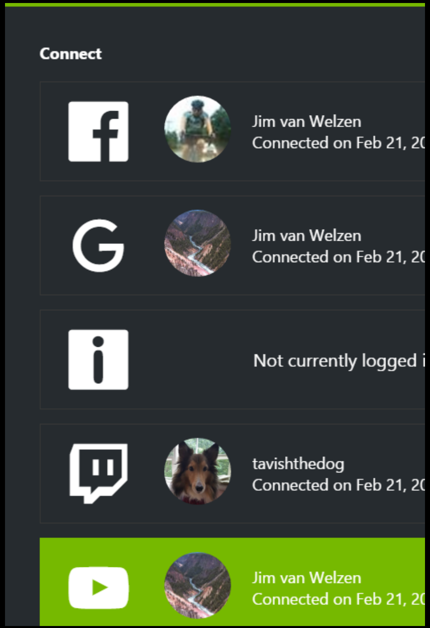
Title: Highlights: Annihilation | Shot with GeForce GTX

Location: On my channel

Audience: Private

In-game overlay
Review, trim, upload

SHARE (USER)



Connect

Jim van Welzen
Connected on Feb 21, 20

Jim van Welzen
Connected on Feb 21, 20

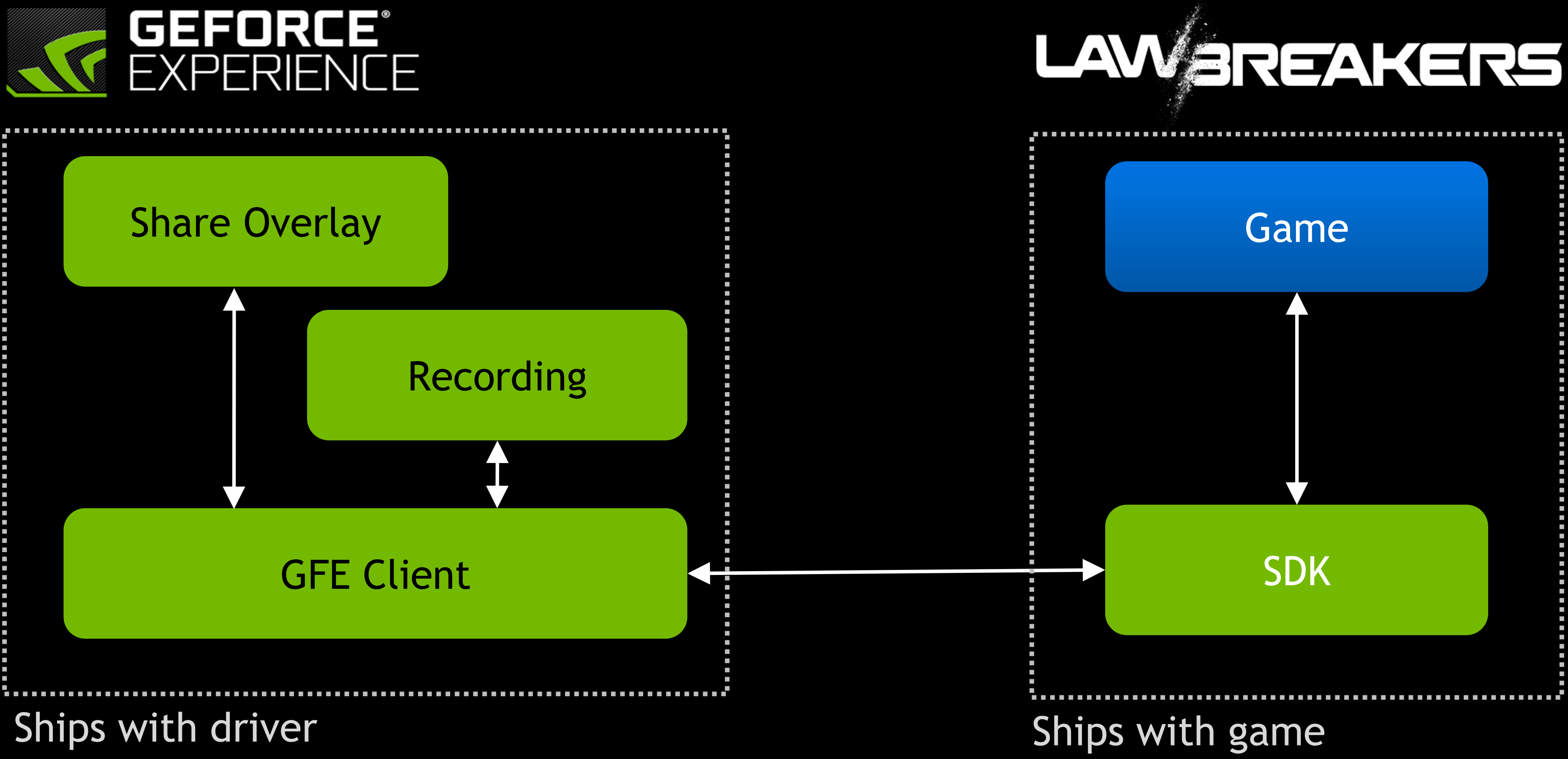
Not currently logged i

tavishthedog
Connected on Feb 21, 20

Jim van Welzen
Connected on Feb 21, 20

Facebook
YouTube / Google+
Imgur
more coming...

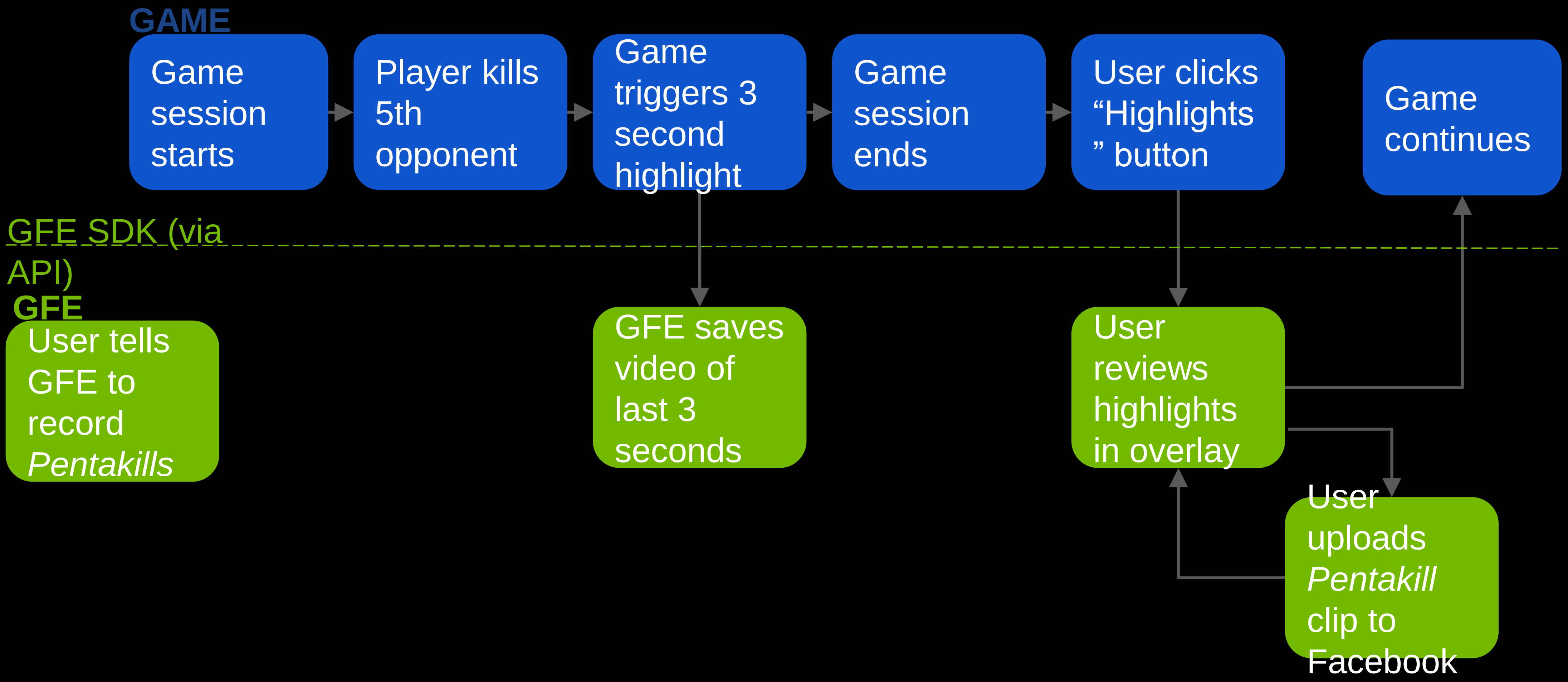
ShadowPlay Highlights Architecture



API

NVGSDK_**Create**	// Construct the main SDK interface.
NVGSDK_Highlights_**Configure**	// Provide a list of possible highlight types to GFE
NVGSDK_Highlights_**StartSession**	// Begin a session which groups several highlights together
NVGSDK_Highlights_**SetScreenshotHighlight**	// Captures a screenshot highlight of given type for current session
NVGSDK_Highlights_**SetVideoHighlight**	// Captures a video highlight of given type for current session
NVGSDK_Highlights_**StopSession**	// Stop a session which groups several moments together
NVGSDK_Highlights_**OpenSessionSummary**	// Ask GFE to display summary for all highlights in the last session
NVGSDK_**Release**	// Release the main SDK interface

Example ShadowPlay Highlights Flow



GAME

Game session starts

Player kills 5th opponent

Game triggers 3 second highlight

Game session ends

User clicks “Highlights” button

Game continues

GFE SDK (via API)

GFE

User tells GFE to record *Pentakills*

GFE saves video of last 3 seconds

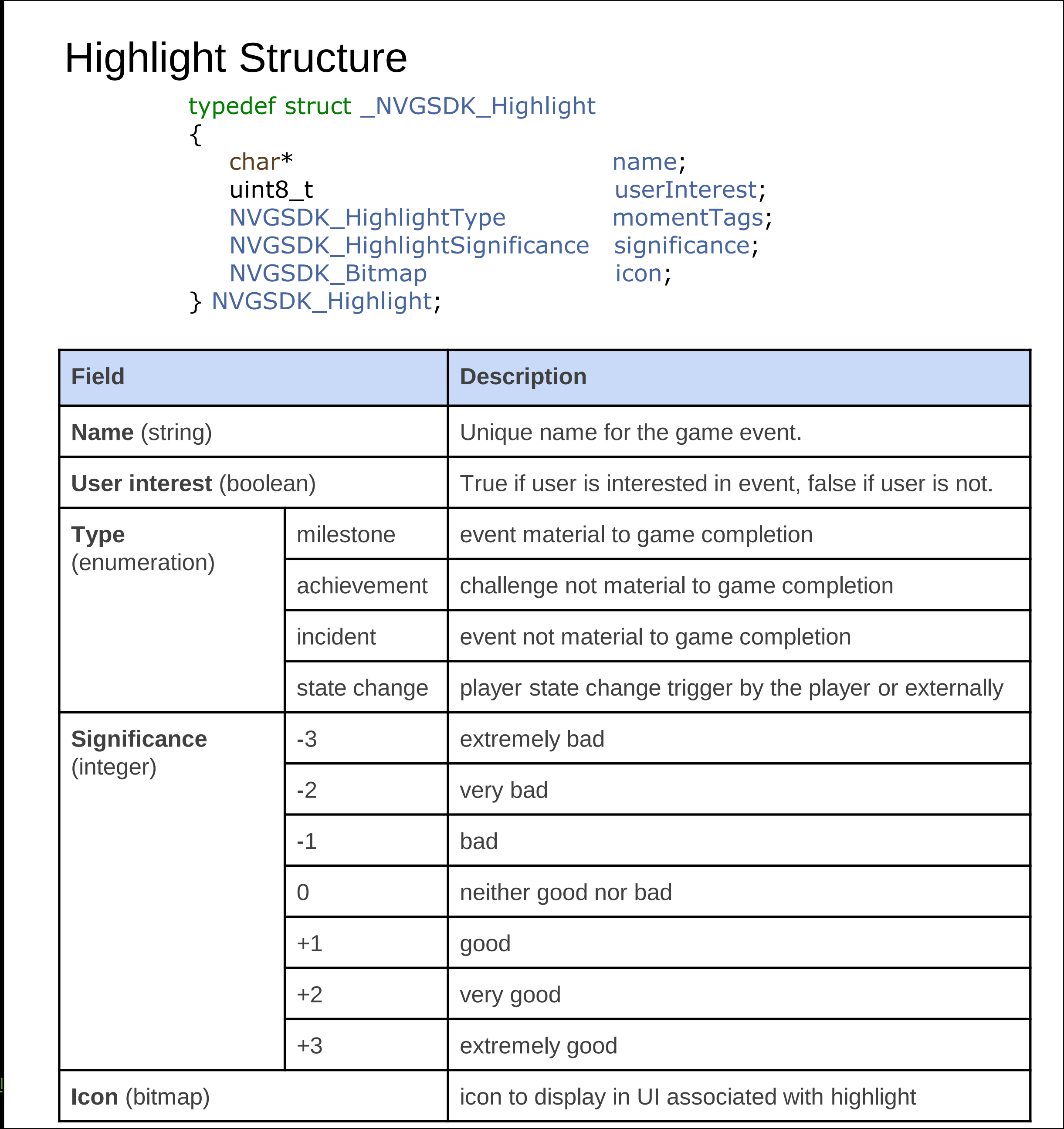
User reviews highlights in overlay

User uploads *Pentakill* clip to Facebook

Highlights Table

- Linchpin of coordination
- Table of type NVGSDK_Highlight
- Defines each game's possible events

- Game passes table to GFE on Configure
- GFE presents to user (by name) for opt in/out
- GFE only captures opted-in highlights
- Metadata available for filtering in summary



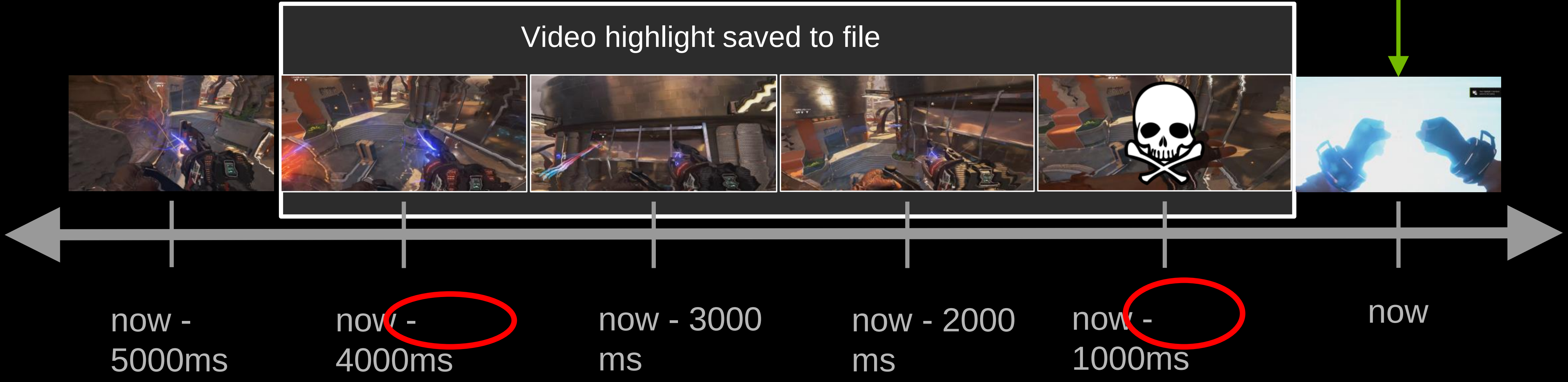
Highlight Structure

```
typedef struct _NVGSDK_Highlight
{
    char*                           name;
    uint8_t                         userInterest;
    NVGSDK_HighlightType            momentTags;
    NVGSDK_HighlightSignificance    significance;
    NVGSDK_Bitmap                   icon;
} NVGSDK_Highlight;
```

Field		Description
Name (string)		Unique name for the game event.
User interest (boolean)		True if user is interested in event, false if user is not.
Type (enumeration)	milestone	event material to game completion
	achievement	challenge not material to game completion
	incident	event not material to game completion
	state change	player state change trigger by the player or externally
Significance (integer)	-3	extremely bad
	-2	very bad
	-1	bad
	0	neither good nor bad
	+1	good
	+2	very good
	+3	extremely good
Icon (bitmap)		icon to display in UI associated with highlight

Setting a Highlight

```
NVGSDK_Highlights_SetVideoHighlight

    ( hSDK, “5v5Fight”, {“kill”, startDelta = -4000, endDelta = -1000});
```



Video highlight saved to file

now - 5000ms

now - 4000ms

now - 3000 ms

now - 2000 ms

now - 1000ms

now

Putting it all together:
Annihilation Highlight captured in a game session

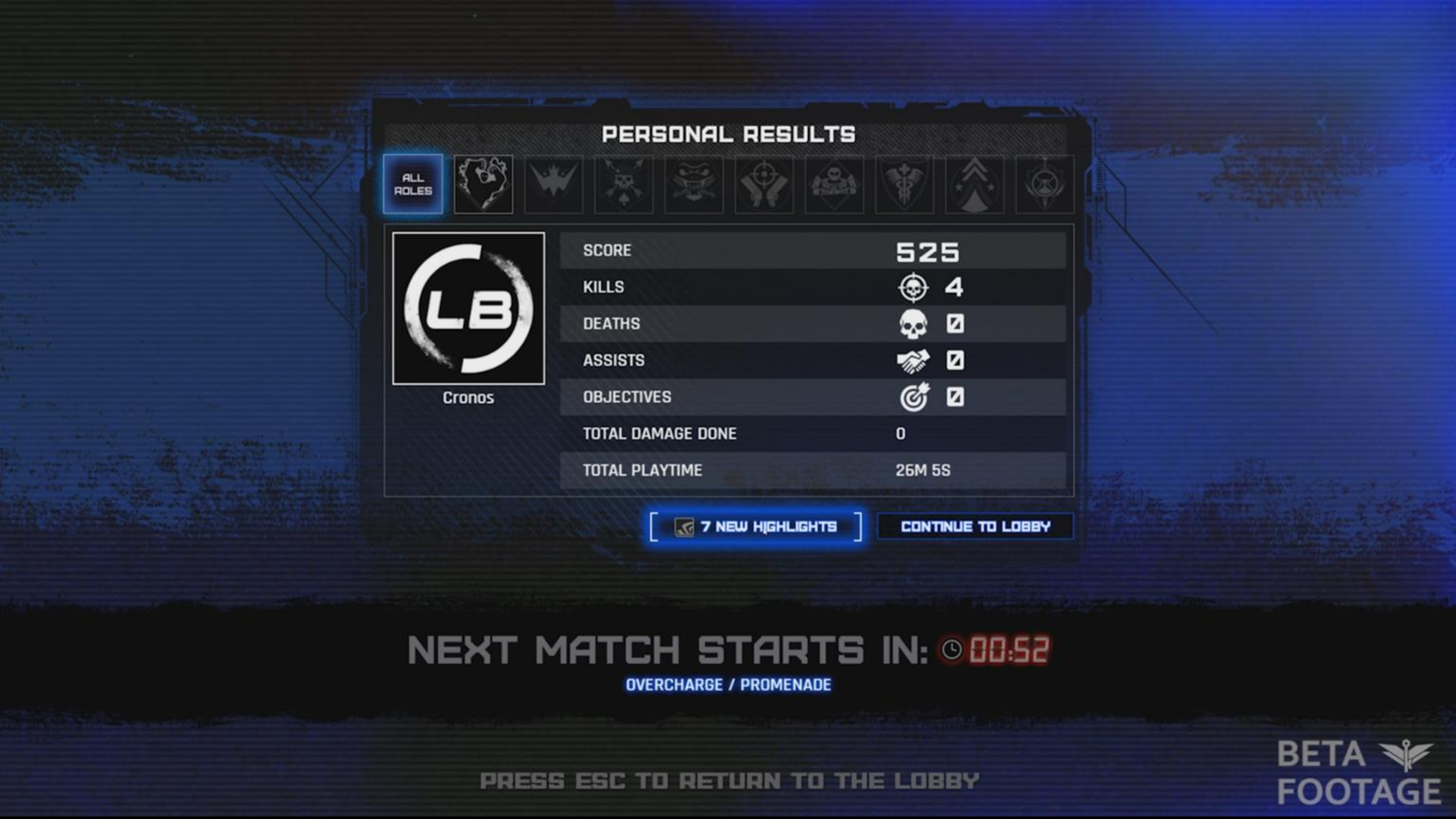
LAWBREAKERS

GRAVITY - DEFYING - COMBAT

Cronos Maverick
Cronos Axel
Cronos Bomchelle
Cronos Axe1
Cronos KILLSTREAK
Low
0
ENEMY BASE
8%
Breakers
0
PROMENADE NORTH
BETA v 128110
ANNIHILATION
QUAD KILL
Q
E
Left Shift
664
HEALTH BONUS
BETA FOOTAGE

Cronos Maverick
Cronos Axel
Cronos Bomchelle
Cronos Axe1
Cronos KILLSTREAK
Law
0
ENEMY BASE
9%
Breakers
0
Highlight saved to Gallery!
34m
BERSERK KILL
Left Shift
E
664
BETA FOOTAGE

Highlight saved to Gallery!
VICTORY
BETA
FOOTAGE



PERSONAL RESULTS

ALL ROLES

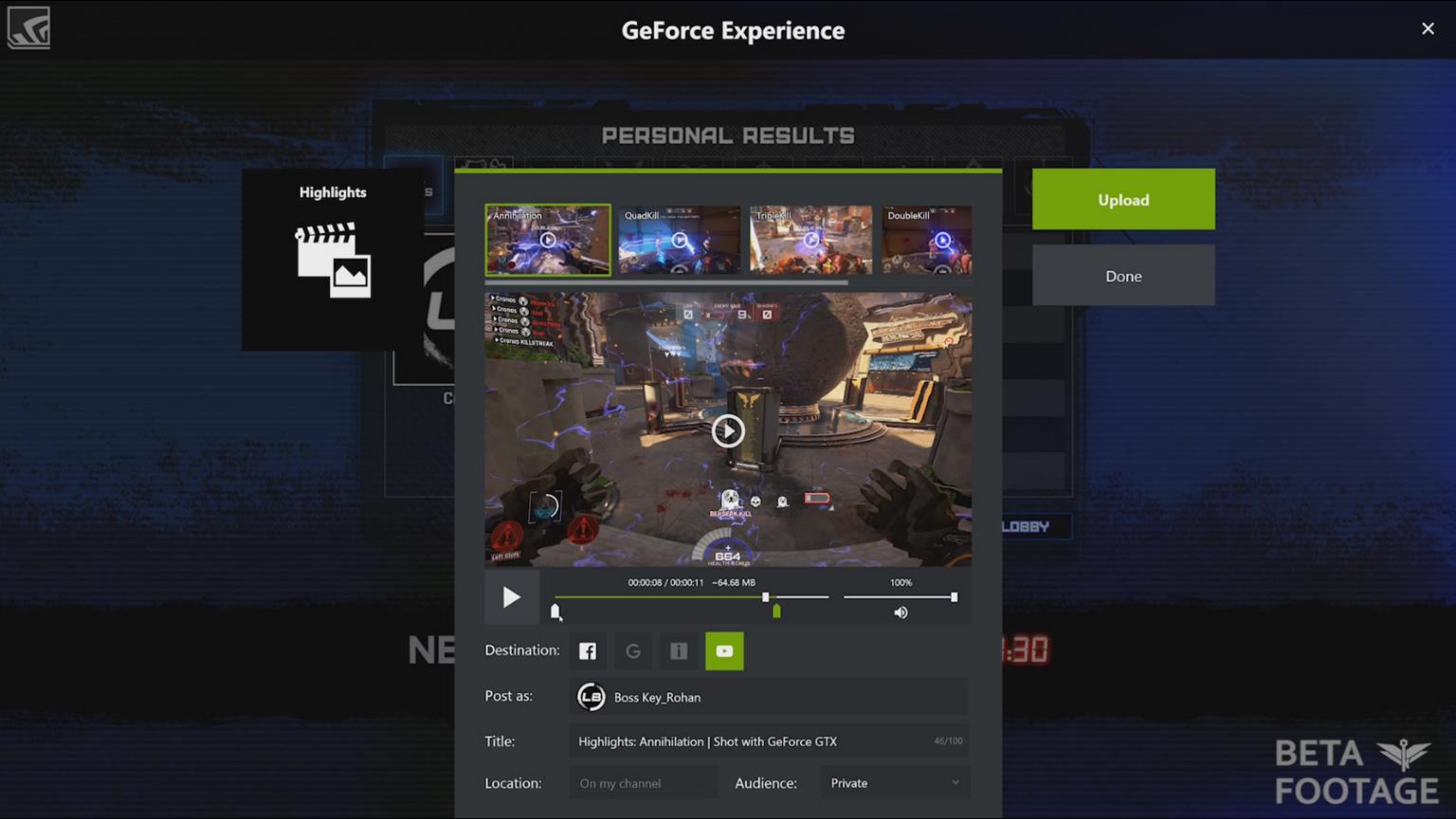
Cronos

SCORE	525
KILLS	4
DEATHS	0
ASSISTS	0
OBJECTIVES	0
TOTAL DAMAGE DONE	0
TOTAL PLAYTIME	26M 5S

NEXT MATCH STARTS IN: 00:52

OVERCHARGE / PROMENADE

PRESS ESC TO RETURN TO THE LOBBY



00:00:08 / 00:00:11 ~64.68 MB

100%

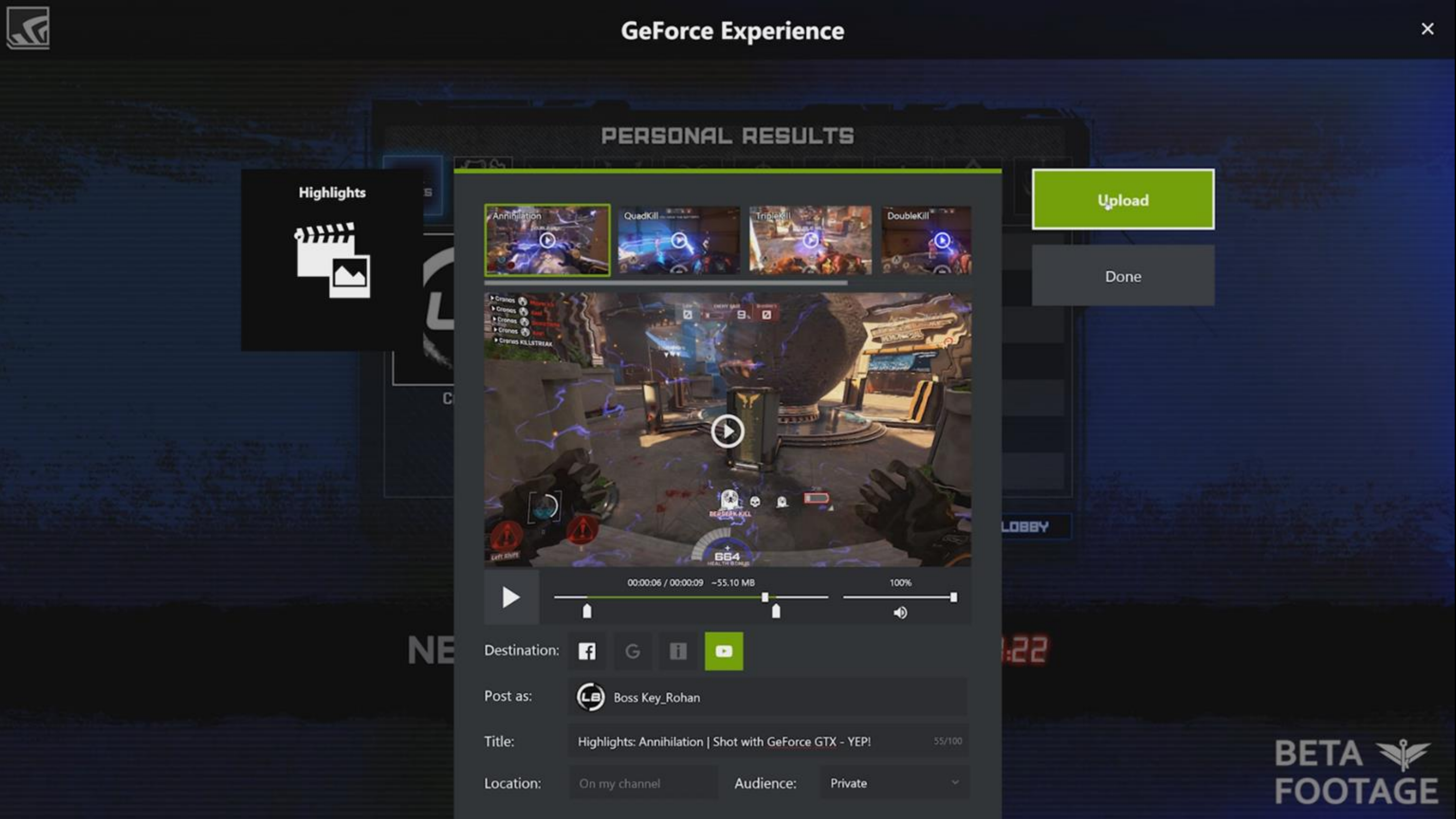
Destination:

Post as: Boss Key_Rohan

Title: Highlights: Annihilation | Shot with GeForce GTX 46/100

Location: On my channel

Audience: Private



00:00:06 / 00:00:09 ~55.10 MB

100%

Destination:

Post as: Boss Key_Rohan

Title: Highlights: Annihilation | Shot with GeForce GTX - YEP! 55/100

Location: On my channel

Audience: Private

Why you should integrate ShadowPlay Highlights

- Capture your players' best gaming moments automatically
- Frictionless sharing to social media
- No game modification required
- Optionally add UI element for access to highlights within game
- Technology works equally well with single- and multiplayer games
- Rolling your own solution is a large investment
- Sign up for SDK: https://developer.nvidia.com/shadowplay-highlights