Live Streaming DerbyNet with OBS

With the COVID-19 pandemic, many organizations find themselves live streaming their "virtual pinewood derby" events. This document provides some hints and tips for live streaming with DerbyNet and Open Broadcaster Software (http://obsproject.com). Many of the suggestions here likely apply to using DerbyNet with other encoders as well.

General Set-Up Considerations

USB Cable Lengths and Power Distribution

Passive USB 2.0 cables are limited to about 5 meters, or 16 feet, in length, according to the USB standard. Longer cables *may* work, but there's no guarantee. A pinewood derby track, on the other hand, might be 40 or 50 feet in length. Especially if you're using multiple USB cameras, it may not be possible to position a single computer within 5 meters of each of them. If that's the case for you, you'll need either powered USB hub(s) or "active" repeater cables, or both.

In particular, note that an active repeater cable itself draws power, in addition to the power requirements of the camera or other device at the end. This may require more total power than a laptop USB port can provide. To avoid an undervoltage condition at the camera or device, you may need to use a powered USB hub in addition an active repeater cable.

If you suspect or anticipate problems along these lines, consider investing in a USB power meter.

Some products and further reading:

http://www.yourcablestore.com/USB-Cable-Length-Limitations-And-How-To-Break-Them_ep_42-1.html

https://www.amazon.com/Monoprice-Extension-Repeater-PlayStation-Keyboard/dp/B009GUTFX8/

https://www.amazon.com/MakerHawk-3-7-30V-Voltage-Multimeter-Voltmeter/dp/B07FMQZVW2/

https://www.aliexpress.com/item/32922333751.html

Bit Rates

The default bit rates set by OBS are pretty low. For better-looking video, check the recommended bit rate for your video platform. E.g., for YouTube, take a look at https://support.google.com/youtube/answer/2853702?hl=en.

Wi-Fi

If you have a choice between connecting over Wi-Fi and connecting over Ethernet, prefer Ethernet. That's not to say that Wi-Fi will give you problems; lots of people stream successfully over Wi-Fi. It's just that a wired connection, if easily available, lets you avoid risk of radio interference or other issues.

Scene Switching with a USB Keypad

OBS can be configured with "hot keys" for scene switching. Assigning hot keys to a cheap USB

keypad lets your emcee do scene switching with minimal instruction and fuss. (And for way less than the cost of an Elgato Stream Deck.)

There's also the OBS Remote app that uses a phone in place of dedicated hardware.

Take Events Directly from derby-timer.jar[1]

derby-timer.jar is capable of connecting to the OBS websocket plugin and transmitting requests in response to heat start[2] and heat finish events. In particular, it's possible to trigger hotkeys to cause scene changes or other actions.

Communication with an OBS websocket requires an -obs-uri command line argument. If the websocket is protected by password, an -obs-password command line argument will also be needed. To trigger a hotkey for heat finish, add the -obs-finish argument with the name of the hotkey, e.g.,

```
java -jar derby-timer.jar \
      -obs-uri ws://192.168.0.12:4444 \
      -obs-password secret \
      -obs-finish OBS_KEY_NUM2
```

The -obs-start and -obs-finish arguments each accept two forms:

- The argument can be the name of a hotkey, from the list of names found at https://github.com/obsproject/obs-studio/blob/master/libobs/obs-hotkeys.h. (Key modifiers cannot be included in this form.)
- If the argument starts with an '@' character, the rest of the argument is the path to a file that should contain a full websocket request, as described at https://github.com/Palakis/obs-websocket/blob/4.x-current/docs/generated/protocol.md.

Some DerbyNet Easter Eggs

Kiosk With Specified Content

Kiosk content is normally controlled by the race coordinator, from the Kiosk Dashboard page. For broadcast, you may wish to have inset views of specific kiosk pages. In OBS, use a "Browser" view for this, and add a 'page' query parameter to the URL, to set a specific page. The argument gives the relative path to the kiosk page file, normally "kiosks/*the-page*.kiosk".

For example, for a hosted DerbyNet instance, the URL for an inset of the "Now Racing" page would look like:

https://hosting.jeffpiazza.org/derbynet/2020/ma/demo/kiosk.php?page=kiosks/now-racing.kiosk

On-Deck Display for the Audience

Normally, the on-deck page or kiosk is focused on the heat that will *follow* the heat currently staging.

---

1 Many users report a preference to leaving replay in the hands of the emcee rather than going to the trouble of configuring derby-timer.jar and OBS websockets. Accordingly, the functionality described in this section may be removed in a future release. If you use it and find it valuable, speak up!

2 Many timers can only signal the end of a heat, and are not capable of signaling the start of a heat. Obviously, heat start events can only be sent if derby-timer.jar can detect the heat start.

(This helps the wrangler get an early start gathering the cars for that next heat, to minimize the time between heats.) If you instead intend to show the on-deck page directly to the audience, add the 'focus' query parameter to the URL. E.g.

https://my-local-server.local/derbynet/kiosk.php?page=kiosks/ondeck.kiosk&focus=current

The 'focus' query parameter can take the values 'current', 'previous', or 'next', to direct the page to scroll to the current, previous, or next heat entry, respectively. (Note that the row for the previous heat doesn't show any car photos, but current and next heat rows do.)

Note that the kiosk version of the on-deck page automatically scrolls so that the current focused heat is vertically centered in the window. For an inset, you may wish to adjust the width and height so that just one row is visible; that single row will always show the current heat.

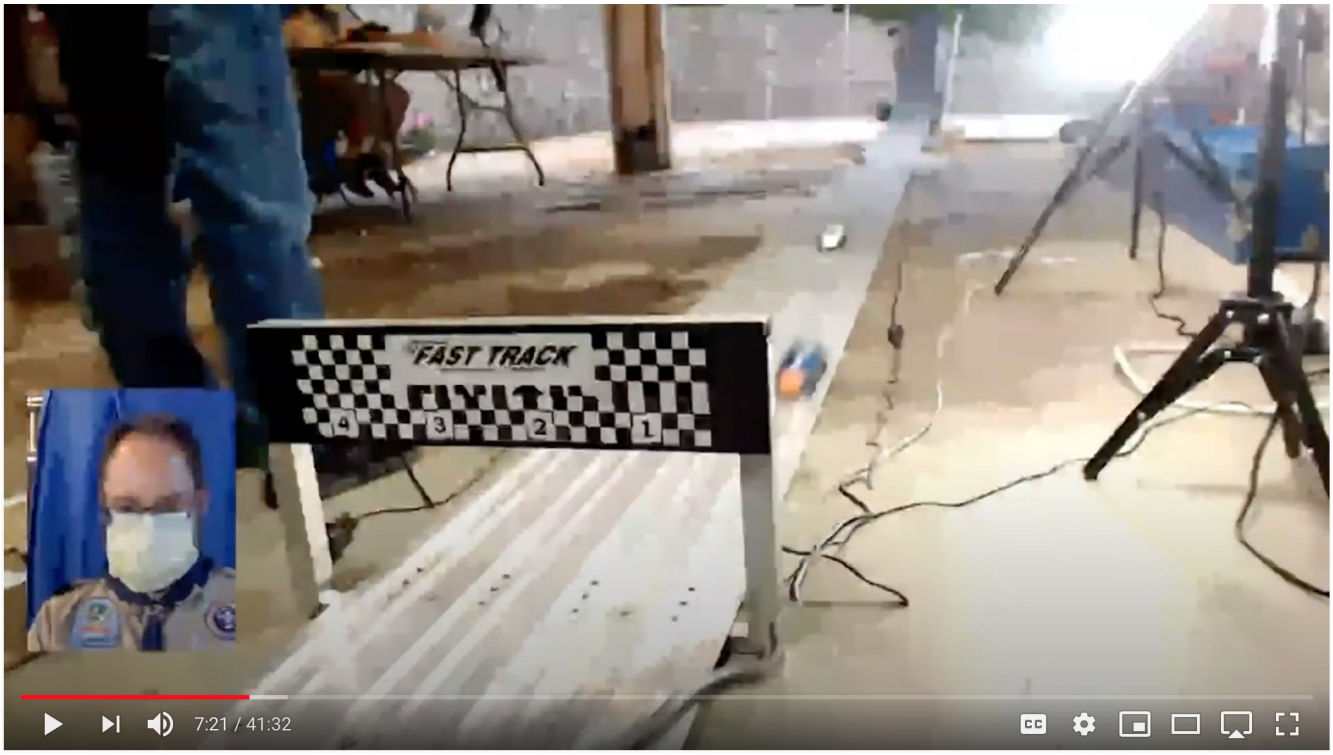
Online Balloting for Awards

One way to help your audience to stay engaged for your event is to ask them to vote on who should receive some of the non-speed awards, like "Best Patriotic Theme," "Coolest Car," or "Best in Show." Because your audience will all be online, you can just give them a URL to visit and register their preference. See the "Online Balloting with DerbyNet" document for more details.

Scene Suggestions

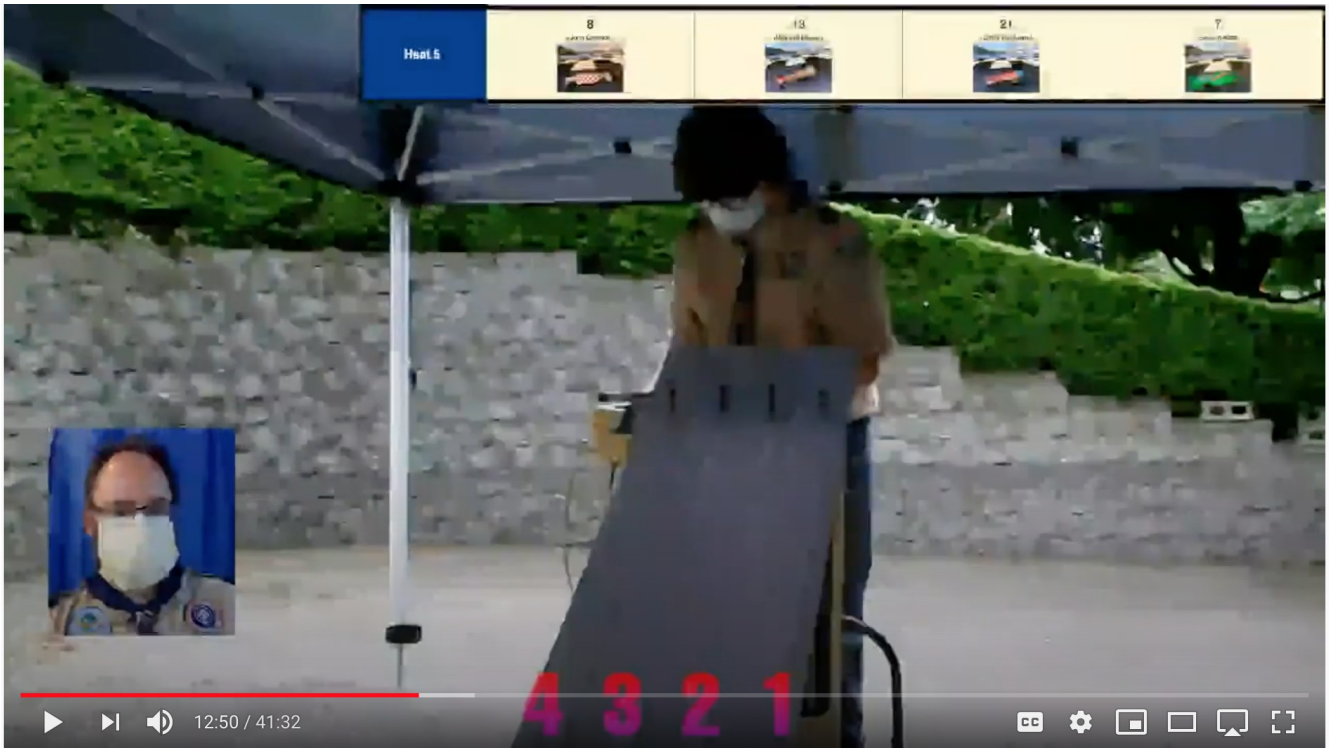
Track View Scene

While a heat is running, your audience would most like to see a view of the track. Place a camera near the finish line, aimed to capture as much of the track as possible. You may wish to add an inset of the "Now Racing" page to remind the audience what cars they're seeing, and/or an inset of the you or your emcee



Starting Gate Scene

If you have a second camera, consider a view of just the start of the track, so the audience can see the cars as they're being staged. You might also add an inset of the "On Deck" (see 'focus_current', above) or "Now Racing" page. If your track lanes don't read left to right in this view, consider adding some overlaid lane labels.



Kiosk Scenes

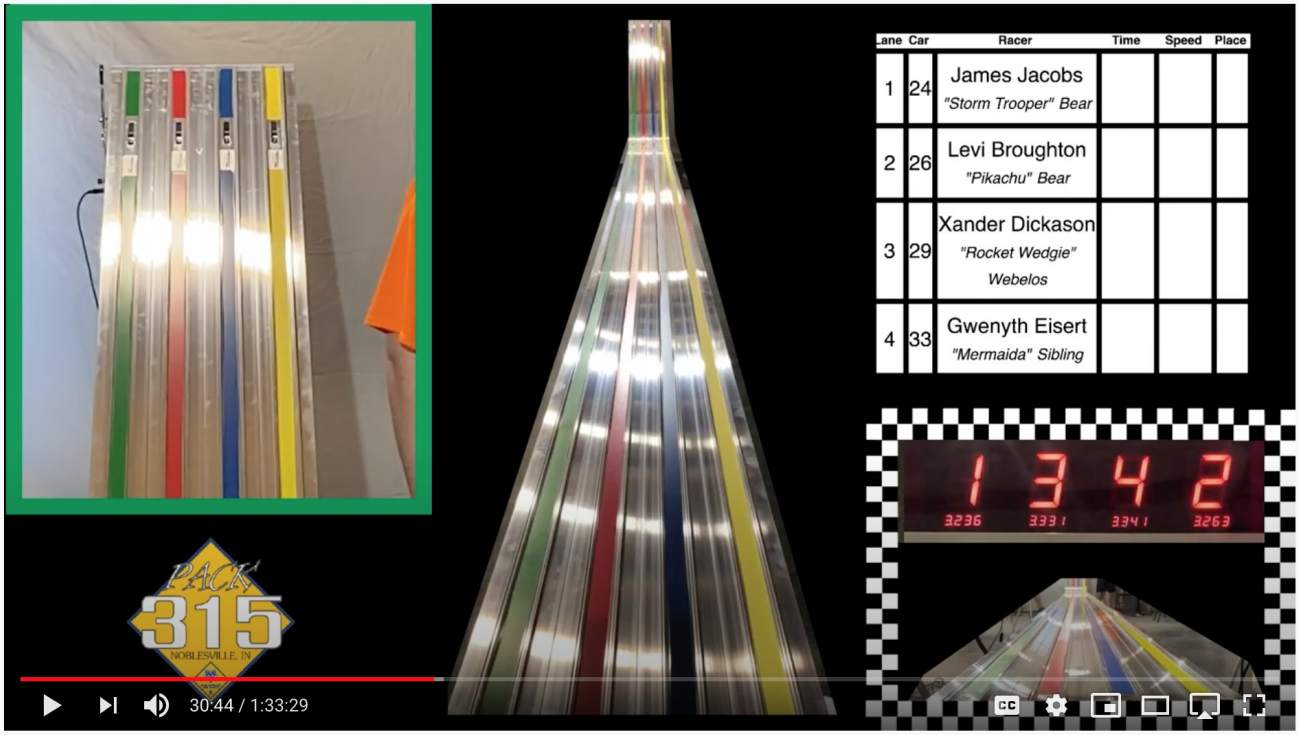
Set a scene that's primarily or exclusively a kiosk browser. In fact, it can be useful to have two such scenes, one having an inset of the emcee and one focusing just on the kiosk display.

Note that an OBS "Browser" source won't work if you need to interact with the page as part of the set up. In particular, a replay kiosk requires interaction from the user, so in OBS it must be set up as either a "Window Capture" or "Display Capture" source in OBS. However, see the note below about replay kiosk performance.

Note further that OBS on MacOS has a bug which causes flicking of "Window Capture" sources. To avoid the flickering, the only real option is to use "Display Capture," which may be difficult unless you can attach a second display to your Mac.

"All in One" Scenes

Here's an example of using an OBS image mask and combining several video sources into one scene.



Pack 315 Official Pinewood Derby

Single-Computer Performance Concerns with Replay

DerbyNet's replay kiosk feature provides "instant replay" capability in a kiosk. It's a real audience-pleaser, but it's also computationally demanding.

OBS, and likely any other software stream encoder, also requires some significant computing horsepower.

My personal experience has been that my quad-core laptop just wasn't able to meet the challenge of running both OBS and a replay kiosk at the same time. YouTube would initially report "excellent" stream quality, but would then warn that too little video was being received. Also, Chrome would warn (in the console) that many frames were taking too long to be captured, and the actual replay would be jerky and nearly unwatchable.

You may find you're better off using the instant-replay features built into OBS directly, especially the "Replay Source" plug-in (https://obsproject.com/forum/resources/replay-source.686/), rather than doubly tax the CPU by trying to leverage a browser window running DerbyNet replay[3]. The "Replay Source" plug-in has an instructional video that accompanies it (be sure to turn on subtitles, or you're left to guess what it means).

3 Effectively, this is asking OBS to re-encode video as it's being decoded by the replay kiosk, all on the same machine. Your CPU and GPU will thank you for eliminating the middleman.