// Copyright 2013 The Flutter Authors. All rights reserved.

// Use of this source code is governed by a BSD-style license that can be

// found in the LICENSE file.

#include "include/flutter/flutter\_view\_controller.h"

#include <algorithm>

#include <iostream>

namespace flutter {

FlutterViewController::FlutterViewController(int width,

int height,

const DartProject& project) {

engine\_ = std::make\_shared<FlutterEngine>(project);

controller\_ = FlutterDesktopViewControllerCreate(width, height,

engine\_->RelinquishEngine());

if (!controller\_) {

std::cerr << "Failed to create view controller." << std::endl;

return;

}

view\_ = std::make\_unique<FlutterView>(

FlutterDesktopViewControllerGetView(controller\_));

}

FlutterViewController::~FlutterViewController() {

if (controller\_) {

FlutterDesktopViewControllerDestroy(controller\_);

}

}

FlutterViewId FlutterViewController::view\_id() const {

auto view\_id = FlutterDesktopViewControllerGetViewId(controller\_);

return static\_cast<FlutterViewId>(view\_id);

}

void FlutterViewController::ForceRedraw() {

FlutterDesktopViewControllerForceRedraw(controller\_);

}

std::optional<LRESULT> FlutterViewController::HandleTopLevelWindowProc(

HWND hwnd,

UINT message,

WPARAM wparam,

LPARAM lparam) {

LRESULT result;

bool handled = FlutterDesktopViewControllerHandleTopLevelWindowProc(

controller\_, hwnd, message, wparam, lparam, &result);

return handled ? result : std::optional<LRESULT>(std::nullopt);

}

} // namespace flutter