Kerbal Space Program - Bug #11382

Bad resolution calls crash X window manager.

09/14/2016 02:18 AM - psycho_zs

Status:	Unity Bug	**Start date:**	09/14/2016
Severity:	High	**% Done:**	100%
Assignee:			
Category:	Application		
Target version:			
Version:	Build 01473	**Language:**	English (US)
Platform:	Linux	**Mod Related:**	No
Expansion:			

Description

Debian testing 64bit, Openbox.
build id = 01473

This was already touched in now-archived bug about crashes on Linux. I would like to make it a separate issue.

When starting KSP in windowed mode, the window is being rapidly destroyed/recreated at some points. At that times it may randomly jump to weird resolutions, some examples include: 32642x1, 65984x34528, 8006x1.
Depending on resulting rendering area, it may either become unplayable or cause instant out-of-memory situation (usually perceived as long system hang and eventual crash of X server if not treated proactively).

This jump usually happens:

- seconds from the start, before Squad logo
- during loading screen (very rarely, almost never)
- after loading screen, before main menu

Related issues:

Related to Kerbal Space Program - Bug #11955: Game crashes X Windows when sta...	**Unity Bug**	**09/20/2016**
Related to Kerbal Space Program - Bug #13684: Moving the KSP window breaks th...	**Duplicate**	**01/27/2017**
Has duplicate Kerbal Space Program - Bug #14344: KSP randomly overwrites Unit...	**Duplicate**	**03/17/2017**

History

#1 - 09/14/2016 06:02 PM - Stratagerm

- File KSP.log_x_windows_crash added

Update: For more info on this bug, see this thread on reddit

Confirmed also on Ubuntu 14.04.

These windowed mode crashes of the X server to the login prompt started with version 1.1. Prior to 1.1 KSP ran in windowed mode on Linux without issue.

Beginning with version 1.1, starting in windowed mode (without "Full Screen" checked in the graphics options) has caused frequent crashes of the entire X windows desktop.

With 1.2.0.1473 (LinuxPlayer)-pre on Steam, starting it about 10 times there was one X windows crash to login, two times where the Squad logo screen didn't show the progress bar and file names, and seven normal starts.

The crash happens so early there's little in the KSP.log file.

It appears to be timing related. The issue seems limited to windowed mode; a possible workaround is to ensure that Full Screen remains checked.

Note that it affects both standard and custom window sizes. While most of my testing on a 2560 x 1440 display has been at "Custom" size (achieved by selecting 2560 x 1440 resolution and unchecking Full Screen), it has also crashed X windows when set to a standard window size of 1920 x 1200 (a size that easily fits on the screen without resizing).

Note that when it comes up, the KSP window is drawn with the standard three controls: Minimize Window, Restore Window, and Close Window. On mouseover the Restore Window control disappears. This behavior also started with version 1.1.

Steps to reproduce:

1. On Steam, start 1.2.0.1473 (LinuxPlayer)-pre.
2. Uncheck "Full Screen."
3. Restart 10 times (immediately quit upon successful start). It should crash X windows before the tenth restart.

#2 - 09/14/2016 07:32 PM - Stratagerm

"Crashes X windows on startup" is a title which would better describe this bug.

#3 - 09/14/2016 07:58 PM - psycho_zs

The current title is the correct one.

KSP does not crash X. This sequence happens:

1. KSP resizes itself to freakishly large resolution.
2. All memory is sucked up instantly to provide for this huge rendering area (at this point all processes grind to near complete stop, apparent system freeze happens).
3. Kernel OOM killer kills whatever has the highest OOM score, which usually happens to be X server process.

Here is the current iteration of my launcher script for 1.2.2 and 1.2.9 Prerelease. It puts two failsafes in place:

1. Starts temporary geometry guard daemon that kills KSP if one of its sides goes more than 3x of desired resolution (requires xwininfo from x11-utils).
2. Bumps OOM score of KSP process to put it first on OOM death row (in case geometry guard couldn't kill it in time).
3. Also it removes window constraints and restores correct window size every 3 seconds, so it can recover from non-fatal size jumps (if the game itself can continue). (Requires xprop from x11-utils).

```
#!/bin/sh

# http://forum.kerbalspaceprogram.com/index.php?/profile/137644-psycho_zs/
# v201703.2
# WTFPL

BINARY=KSP.x86_64

WIN_WIDTH=1366
WIN_HEIGHT=768

GEOM_GUARD_TIME="180 seconds"

ALLOW_WIDTH_MAX=$(( $WIN_WIDTH * 3 ))
ALLOW_HEIGHT_MAX=$(( $WIN_HEIGHT * 3 ))

GUARD=1
[ ! -x "$(which xwininfo)" ] && { GUARD=0 ; printf "\e[31mWarning: xwininfo not found or is not an executable!
\n         Geometry guard will not work!\e[39m\n" >&2 ; }
[ ! -x "$(which xprop)" ] && { GUARD=0 ; printf "\e[31mWarning: xprop not found or is not an executable!\n
     Geometry guard will not work!\e[39m\n" >&2 ; }

# change dir to script location
cd "$(dirname "$(readlink -f "$0")")"

printf "Still can not learn localized fraction delimiters? setting LC_ALL=C\n"
LANG=C
LC_ALL=C
export LC_ALL
export LANG
unset LANGUAGE

printf "Fixing weird permissions from game archive.\n"
find . -type d -exec chmod 755 "{}" \;
find . -type f -exec chmod 644 "{}" \;
chmod 755 KSP.x86  KSP.x86_64  Launcher.x86  Launcher.x86_64 "${BINARY}" "$(basename "$0")"

# trap for child process cleanup
trap 'kill $KSP_PID 2> /dev/null' EXIT HUP INT TERM

GEOMKILL="2"
COUNT="0"

until [ "$GEOMKILL" = "0" ]
do
    [ "$GEOMKILL" = "1" ] && printf "\nHere we go again...\n"
    printf "Clearing unity3d config dir\n"
    [ -d "${XDG_CONFIG_HOME:-$HOME/.config}/unity3d/" ] && rm -r "${XDG_CONFIG_HOME:-$HOME/.config}/unity3d/"

    GEOMKILL="0"
    COUNT=$(( $COUNT + 1 ))

    printf "\n\e[32mLaunching KSP, attempt ${COUNT}... Hold on to your helmets and pray the Kraken!\e[39m\n\n"

    ./"${BINARY}" -screen-fullscreen 0 -popupwindow -screen-width $WIN_WIDTH -screen-height $WIN_HEIGHT &
    KSP_PID=$!

    printf "\e[32mLaunched with PID ${KSP_PID}.\e[39m\n\n"

    printf "\e[33mBumping KSP OOM killer score in case it goes berserk and chews up all the memory at once.\nCurrent OOM score: $(cat /proc/${KSP_PID}/oom_score)\e[39m\n"
    echo 15 > "/proc/${KSP_PID}/oom_adj"
    printf "\e[33mNew OOM score: $(cat /proc/${KSP_PID}/oom_score)\e[39m\n"

    if [ "$GUARD" = "1" ]
    then

        printf "Starting window geometry repairer...\n"

        while kill -0 "$KSP_PID" 2> /dev/null
        do
            # drop potentially corrupted window hints including size and increment restrictions
            # and force window size
            xprop -name "Kerbal Space Program" -remove WM_NORMAL_HINTS 2>/dev/null
            wmctrl -Fr "Kerbal Space Program" -e "0,-1,-1,${WIN_WIDTH},${WIN_HEIGHT}" 2>/dev/null
            # do this once in 3 seconds, we're not rushing anywhere
            sleep 3
        done &

        printf "Starting window geometry guard...\n"
        STOPDATE="$(date +%s -d "$GEOM_GUARD_TIME")"

        while kill -0 "$KSP_PID" 2> /dev/null && [ "$(date +%s)" -lt "$STOPDATE" ]
        do
            # this should be fast, only one subprocess per iteration, the rest is shell
            GEOM="$(xwininfo -name "Kerbal Space Program" 2>/dev/null)"
            HEIGHT="${GEOM##*Height:}"
            HEIGHT="${GEOM##*[[:space:]]}"
            HEIGHT="${HEIGHT%%[^0-9]*}"
            WIDTH="${GEOM##*Width:}"
            WIDTH="${GEOM##*[[:space:]]}"
            WIDTH="${WIDTH%%[^0-9]*}"

            # Check if geometry goes too far up.
            if [ "${WIDTH:-0}" -gt "$ALLOW_WIDTH_MAX" -o "${HEIGHT:-0}" -gt "$ALLOW_HEIGHT_MAX" ]
            then
                kill -9 "$KSP_PID"
                GEOMKILL=1
                printf "\n\e[31mWindow geometry is freaky (${WIDTH:-NA}x${HEIGHT:-NA}), killed KSP!\e[39m\n\n" >&2
                sleep 1
                break
            else
                GEOMKILL=0
            fi
            [ "$WIDTH" != "$PREVWIDTH" -o "$HEIGHT" != "$PREVHEIGHT" ] && printf "\e[33mWindow geometry changed from ${PREVWIDTH:-NA}x${PREVHEIGHT:-NA} to ${WIDTH:-NA}x${HEIGHT:-NA}\e[39m\n"
            PREVWIDTH=$WIDTH
            PREVHEIGHT=$HEIGHT
        done
        [ "$GEOMKILL" = "0" ] && printf "\e[33mWindow geometry guard is (hopefully) no longer needed.\e[39m\n"

    else
        printf "\e[31mWindow geometry guard is missing its tools!\nYou're on your own with OOM.\e[39m\n" >&2
    fi
done

wait
```

########################

Older version for historic reasons:

```
#!/bin/bash

WINWIDTH=1366
WINHEIGHT=768
BINARY=KSP.x86_64

# change dir to script location
cd "$(dirname "$(readlink -f "$0")")"

printf "Still can not learn locales while reading settings? setting LC_ALL=C\n"
LANG=C
LC_ALL=C
export LC_ALL
export LANG
unset LANGUAGE

printf "Fixing weird permissions from game archive.\n"
find . -type f -exec chmod 644 "{}" \;
find . -type d -exec chmod 755 "{}" \;
chmod 755 KSP.x86  KSP.x86_64  Launcher.x86  Launcher.x86_64 "${BINARY}" "$(basename "$0")"

{
    [ -x "$(which xwininfo)" ] || { printf "xwininfo not found, aborting window geometry guard!\n" >&2 ; exit 1 ; }
    STOPDATE="$(date +%s -d "180 seconds")"
    printf "Starting window geometry guard...\n"
    while [ "$(date +%s)" -lt "$STOPDATE" ] > /dev/null
    do
        WIDTH="$(xwininfo -name "Kerbal Space Program" 2>/dev/null | grep -o 'Width:[[:space:]]*[0-9]\+' | grep -o '[0-9]\+')"
        HEIGHT="$(xwininfo -name "Kerbal Space Program" 2>/dev/null | grep -o 'Height:[[:space:]]*[0-9]\+' | grep -o '[0-9]\+')"
        [ "${WIDTH:-0}" -gt "$(( $WINWIDTH * 2 ))" ] && { printf "Window geometry is freaky (${WIDTH}x${HEIGHT}), killing KSP!\n" >&2 ; killall -9 "${BINARY}" ; exit 0 ; }
        [ "${HEIGHT:-0}" -gt "$(( $WINHEIGHT * 2 ))" ] && { printf "Window geometry is freaky (${WIDTH}x${HEIGHT}), killing KSP!\n" >&2 ; killall -9 "${BINARY}" ; exit 0 ; }
        [ "$WIDTH" != "$PREVWIDTH" -o "$HEIGHT" != "$PREVHEIGHT" ] && printf "Window geometry changed from ${PREVWIDTH:-NA}x${PREVHEIGHT:-NA} to ${WIDTH}x${HEIGHT}\n"
        PREVWIDTH=$WIDTH
        PREVHEIGHT=$HEIGHT
        sleep 0.2
        pidof "${BINARY}" >/dev/null || break
    done
    printf "Window geometry guard no longer needed.\n"
} &

{
    STOPDATE="$(date +%s -d "2 seconds")"
    until pidof "${BINARY}" > /dev/null || [ "$(date +%s)" -ge "$STOPDATE" ]
    do
        printf "Waiting for \"${BINARY}\" process to appear...\n"
        sleep .1
    done
    if pidof "${BINARY}" > /dev/null
    then
        printf "Bumping KSP OOM killer score in case it goes berserk and chews up all the memory at once.\nCurrent OOM score: $(cat /proc/$(pgrep "${BINARY}")/oom_score)\n"
        echo 15 > "/proc/$(pgrep "${BINARY}")/oom_adj"
        printf "New OOM score: $(cat /proc/$(pgrep "${BINARY}")/oom_score)\n"
    fi
} &

printf "Launching. Hold on to yer helmets and pray the Kraken!\n"
exec ./"${BINARY}" -screen-fullscreen 0 -screen-width $WINWIDTH -screen-height $WINHEIGHT
```

If geometry guard makes it in time, KSP would be killed instantly. If freeze happens before that, OOM killer would do it, X and everything else will survive.

#4 - 09/14/2016 10:33 PM - psycho_zs

...The bug also can be triggered by applying video settings in-game, even if resolution setting was not changed.

I do not know much about starting process, but why window is being recreated two times (or possibly more in rapid succession) during launch? Is it internal Unity's thing or can it be tamed? It also has an inconvenient side-effect of jumping into current workspace (apparently, because it is a new window every time) after being launched on another workspace.

#5 - 09/15/2016 11:11 AM - Stratagerm

Still present in 1.2.0.1479 (LinuxPlayer)-pre.

The OOM-Killer abort which kills X Windows may depend on the window manager being run. I'm running Ubuntu 14.04.5 LTS using a Gnome-Flashback/Metacity session. Most Ubuntu users are using the default Unity desktop (not to be confused with the Unity game engine). Other window managers may be more resistant to responding to requests for outrageously-sized windows, which is why not all Linux users experience the X Windows termination.

From /var/log/syslog at the time of termination:

```
Sep 14 15:55:21 linuxbox gnome-session[6405]: WARNING: Application 'metacity.desktop' killed by signal 6
Sep 14 15:55:21 linuxbox gnome-session[6405]: WARNING: App 'metacity.desktop' respawning too quickly
Sep 14 15:55:21 linuxbox gnome-session[6405]: CRITICAL: We failed, but the fail whale is dead. Sorry....
```

#6 - 09/15/2016 12:52 PM - Stratagerm

psycho_zs, your launcher script likely won't work for metacity, which isn't being killed by OOM Killer. OOM Killer sends SIGTERM and SIGKILL. Metacity is being killed with signal 6, which is SIGABRT.

It's likely killing itself by calling abort(3) from the function meta_bug in util.c (metacity-2.34.13). The huge window request likely causes metacity to abort thinking it has a bug. gnome-session restarts it, metacity again sees the huge window request and aborts. Finally gnome-session gives up on it (gsm_app_restart in gsm-app.c) and ends up at gsm_fail_whale_dialog_we_failed (gsm-fail-whale.c). No OOM Killer involved.

Correction: Metacity's not calling function meta_bug, so it's not aborting itself. It's calling g_log_set_handler to set log_handler to handle gnome errors which calls meta_warning which prints the "Native children" message (see post below) and calls meta_print_backtrace which prints the backtrace. The SIGABRT is likely coming from glibc or another library. But still no OOM Killer involved.

This KSP bug thus acts differently depending on which window manager is used. What window manager are you using? Probably not metacity.

#7 - 09/15/2016 01:23 PM - psycho_zs

Openbox.

So, geometry guard fails for metacity? xwininfo should still work when there is no window manager running.

If gnome-session uses near-zero or zero timeouts for restarting it, there may be not enough time...

#8 - 09/15/2016 02:03 PM - psycho_zs

I've made some changes to the script. Can you try it again, saving the output?:

```
launcher.sh 2>&1 | tee -a /tmp/ksplauncher.log
```

#9 - 09/15/2016 03:25 PM - Stratagerm

Didn't work. Failed on the first run.

Hadn't tried it before now, but had figured out that OOM Killer didn't pertain to Metacity. And as you say, Metacity likely aborts too quickly.

For log see http://pastebin.com/S2nLubxg

I modded the script slightly, removing the permission changing stuff. And adjusted for my window size.

Not too concerned with a script workaround for this bug, as the workaround is to run Full Screen (which users of compositing window managers tend to do). The right thing is for KSP, Metacity, and Openbox to be fixed.

#10 - 09/15/2016 04:06 PM - psycho_zs

As a last ditch you can remove "sleep 0.2" in geometry guard cycle. Tighter cycle may catch the moment of wrong resolution in between metacity restart attempts.

The other possibility is that with metacity case it is not the size of the window, but something else causing it to crash.

#11 - 09/15/2016 04:40 PM - Stratagerm

[Numerous edits to this entry.]

Confirming the Metacity self-abort.

Restarting Metacity to produce a log file, and then getting the KSP bug:

```
> env METACITY_VERBOSE=1 METACITY_USE_LOGFILE=1 metacity --replace
Opened log file /tmp/metacity-23095-debug-log-EIVJNY
Segmentation fault (core dumped)
>
```

The log file shows this message from the callback to log_handler in Metacity just before it's aborted by glibc:

```
Window manager warning: Log level 16: Native children wider or taller than 65535 pixels are not supported
```

The "Native children" string is from _gdk_x11_window_move_resize_child in gdk/x11/gdkgeometry-x11.c from Ubuntu package libgtk-3-0.

That string shows up in a post to a KSP forum thread from July 22 titled KSP Unloadable on Linux in a /var/log/syslog excerpt from someone running gnome-shell (which was terminated).
http://forum.kerbalspaceprogram.com/index.php?/topic/144250-ksp-unloadable-on-linux/#comment-2686710

Note also these lines from earlier in the Metacity log file:

```
GEOMETRY: Window 0x500000e (Kerbal Spa) sets base size 61517240 x 0
GEOMETRY: Window 0x500000e (Kerbal Spa) sets min size 2490 x 1363
GEOMETRY: Window 0x500000e (Kerbal Spa) sets max size 2490 x 1363
GEOMETRY: Window 0x500000e (Kerbal Spa) sets resize width inc: 1914952540 height inc: 32637
```

There's the confirmation that KSP/Unity 5 is ~~requesting windows~~ passing hints with garbage sizes.

Based on the above GEOMETRY lines from the Metacity log, and looking at the code in src/core/window-props.c, it's likely that the Unity 5 Linux player is making a bad call to gdk_window_set_geometry_hints (or gtk_window_set_geometry_hints). Specifcally, its likely that the GDK_HINT_BASE_SIZE, GDK_HINT_RESIZE_INC, and possibly the GDK_HINT_ASPECT flags are set in the GdkWindowHints geom_mask argument even though the GdkGeometry struct has garbage values in those fields.

https://developer.gnome.org/gdk3/stable/gdk3-Windows.html#gdk-window-set-geometry-hints
https://developer.gnome.org/gdk3/stable/gdk3-Windows.html#GdkWindowHints
https://developer.gnome.org/gdk3/stable/gdk3-Windows.html#GdkGeometry

Note that the documentation for struct GdkGeometry says "gdk_window_set_geometry_hints() expects the hints to be fully valid already and simply passes them to the window manager."

#12 - 09/16/2016 01:03 AM - Stratagerm

I've patched Metacity file src/core/window-props.c to add more sanity checks to function meta_set_normal_hints. This successfully prevents Metacity from aborting.

Looking at the Metacity logs, I'm seeing some strange behavior. The GdkWindowHints geom_mask argument itself is inconsistent. Often GDK_HINT_RESIZE_INC is set even though the width_inc and height_inc fields in the struct GdkGeometry are invalid. But sometimes GDK_HINT_RESIZE_INC isn't set.

There's something else funny going on but I can't yet put a finger on it.

#13 - 09/16/2016 12:17 PM - Stratagerm

#11565, Cinnamon DE crashes on game start, is a duplicate of this bug.

That entry is priority High, which I think more accurately represents the severity of the effect on the user's computer from this bug.

#15 - 09/16/2016 03:40 PM - sal_vager

- *Subject changed from Random resolution jumps at startup to Bad resolution calls crash X window manager.*
- *Category changed from Graphics to Unity3D*
- *Status changed from New to Need More Info*
- *Severity changed from Low to High*

I agree that this should be set higher, as the game does crash, however I don't see a means for Squad to address this as all graphics calls are supposed to be handled by Unity.

The only resolutions that should be requested are the values in the prefs file (by Unity) and the values in settings.cfg (after KSP starts) so where are these other numbers coming from?

I've made a Unity project at 2560x1440 windowed mode, does this also cause the issue?

https://www.dropbox.com/s/cgmckrjj1x5ql48/Roll%20A%20Ball.7z?dl=0

#16 - 09/16/2016 04:35 PM - psycho_zs

On first launch it did jump to 38416x32761 size and kinda hung without any memory hogging. Second time worked fine.

(Openbox here).

So theoretically it can also jump to something >65535 which is fatal for metacity.

...yep, several attempts later it jumped to 17856x65535 when I moved the window a bit.

#17 - 09/16/2016 05:31 PM - sal_vager

- Status changed from Need More Info to Updated

- % Done changed from 0 to 10

That's what I hoped you wouldn't say :(

Okay there is no KSP code in that test, it's literally the tutorial here https://unity3d.com/learn/tutorials/projects/roll-ball-tutorial

Squad will have to report this issue to Unity and see if they fix it, and there's *a lot* of Linux player crash reports on their tracker already.

#18 - 09/16/2016 10:53 PM - waterlubber

I'm having this issue as well, I can't game to run at all even with psycho_zs's wonderful script.

I've had *some* luck with 1.1.2 and changing around the resolution values in settings.cfg and unity's copy (located in ~/.config/unity3d/Squad), so you may be able to mess around with those and get it to work.

Is it possible for Squad to create a workaround or are all linux players just totally boned for now?

#19 - 09/16/2016 10:58 PM - Stratagerm

psycho_zs wrote:

> So theoretically it can also jump to something >65535 which is fatal for metacity.

That error doesn't kill Metacity; my earlier analysis was wrong and I corrected that post. But see my long post below for exactly what happens.

waterlubber wrote:

> I'm having this issue as well, I can't game to run at all even with psycho_zs's wonderful script.

It doesn't work for all Window managers. Just Openbox so far.

> Is it possible for Squad to create a workaround or are all linux players just totally boned for now?

Run Full screen. The problem is with running in windowed mode using a window manager with insufficient sanity checking. **Important: which window manager do you use?**

#20 - 09/16/2016 11:01 PM - Stratagerm

sal_vager wrote:

> I don't see a means for Squad to address this as all graphics calls are supposed to be handled by Unity.

I agreeâ this is a bug in the Unity game engine.

sal_vager wrote:

> The only resolutions that should be requested are the values in the prefs file (by Unity) and the values in settings.cfg (after KSP starts) so where are these other numbers coming from?

To understand this bug you have to understand how calls to gdk_window_set_geometry_hints work. Please review the links to the documentation of the gdk-window-set-geometry-hints function, struct GdkGeometry, and enum GdkWindowHints.

What's happening is that the GdkWindowHints argument to gdk_window_set_geometry_hints has flags (bits) set that shouldn't be, so that the window manager is reading uninitialized values out of the struct GdkGeometry it gets.

Here are some examples from a log produced by my testing of Metacity before I patched it. The logs are extremely verbose and thus heavily excerpted here.

In this case there are four times Metacity gets a struct GdkGeometry from the Unity game engine.

The first time, just the Window's minimum size is set, to a default resonable value:

```
GEOMETRY: Updating WM_NORMAL_HINTS for 0x5000002 (Kerbal Spa)
GEOMETRY: Window 0x5000002 (Kerbal Spa) sets min size 100 x 100
GEOMETRY: Window 0x5000002 (Kerbal Spa) doesn't set gravity, using NW
WINDOW_OPS: Window 0x5000002 (Kerbal Spa) fullscreen = 0 not resizable, maximizable = 1 fullscreenable = 1 min
 size 100x100 max size 2147483647x2147483647
```

Those 2147483647 values? That's Metacity setting the unspecified max values to G_MAXINT; totally normal. This shows that max values can be huge without harming anything.

The second time it's setting the min and max window sizes to the values saved by the Unity game engine in the prefs file. Normal and correct:

```
GEOMETRY: Updating WM_NORMAL_HINTS for 0x5000002 (Kerbal Spa)
GEOMETRY: Window 0x5000002 (Kerbal Spa) sets min size 2490 x 1363
GEOMETRY: Window 0x5000002 (Kerbal Spa) sets max size 2490 x 1363
GEOMETRY: Window 0x5000002 (Kerbal Spa) doesn't set gravity, using NW
WINDOW_OPS: Window 0x5000002 (Kerbal Spa) fullscreen = 0 not resizable, maximizable = 0 fullscreenable = 0 min
 size 2490x1363 max size 2490x1363
```

Now the fun begins. The third time it passes some garbage values for other fields in the struct GdkGeometry because of the flags in the GdkWindowHints argument (which specify which GdkGeometry fields are valid):

```
GEOMETRY: Updating WM_NORMAL_HINTS for 0x5000002 (Kerbal Spa)
GEOMETRY: Window 0x5000002 (Kerbal Spa) sets min size 2490 x 1363
GEOMETRY: Window 0x5000002 (Kerbal Spa) sets max size 2490 x 1363
GEOMETRY: Window 0x5000002 (Kerbal Spa) sets resize width inc: 1914952540 height inc: 32637
GEOMETRY: Window 0x5000002 (Kerbal Spa) sets min_aspect: 1833344640/32637 max_aspect: 15878738/0
GEOMETRY: Window 0x5000002 (Kerbal Spa) sets gravity 0
GEOMETRY: Window 0x5000002 (Kerbal Spa) sets min aspect ratio larger than largest aspect ratio possible given
min/max size constraints; disabling min aspect ratio constraint.
GEOMETRY: XSizeHints: USPosition now unset
WINDOW_OPS: Window 0x5000002 (Kerbal Spa) fullscreen = 0 not resizable, maximizable = 0 fullscreenable = 0 min
 size 2490x1363 max size 2490x1363
```

The width_inc, height_inc, min_aspect, and max_aspect fields all contain uninitialzed values, but it turns out not to hurt anything. Note that Metacity's sanity checking triggers on the aspect ratio values, but not the size increment values.

The fourth time the stuff really hits the fan. This time the GdkWindowHints argument has the GDK_HINT_BASE_SIZE bit set, so Metacity uses the uninitialized values in the base_width and base_height fields of the struct GdkGeometry it gets:

```
GEOMETRY: Updating WM_NORMAL_HINTS for 0x500000e (Kerbal Spa)
GEOMETRY: Window 0x500000e (Kerbal Spa) sets base size 61517240 x 0
GEOMETRY: Window 0x500000e (Kerbal Spa) sets min size 2490 x 1363
GEOMETRY: Window 0x500000e (Kerbal Spa) sets max size 2490 x 1363
GEOMETRY: Window 0x500000e (Kerbal Spa) sets resize width inc: 1914952540 height inc: 32637
GEOMETRY: Window 0x500000e (Kerbal Spa) doesn't set gravity, using NW
GEOMETRY: Window 0x500000e (Kerbal Spa) has width_inc (1914952540) that does not evenly divide min_width - bas
e_width (2490 - 61517240); thus effective min_width is really 1976469780
GEOMETRY: Window 0x500000e (Kerbal Spa) has width_inc (1914952540) that does not evenly divide max_width - bas
e_width (2490 - 61517240); thus effective max_width is really 61517240
GEOMETRY: Window 0x500000e (Kerbal Spa) has height_inc (32637) that does not evenly divide min_height - base_h
eight (1363 - 0); thus effective min_height is really 32637
GEOMETRY: Window 0x500000e (Kerbal Spa) has height_inc (32637) that does not evenly divide max_height - base_h
eight (1363 - 0); thus effective max_height is really 0
GEOMETRY: Window 0x500000e (Kerbal Spa) sets max width 61517240 less than min width 1976469780, disabling resi
ze
GEOMETRY: Window 0x500000e (Kerbal Spa) sets max height 0 less than min height 32637, disabling resize
WINDOW_OPS: Window 0x500000e (Kerbal Spa) fullscreen = 0 not resizable, maximizable = 0 fullscreenable = 0 min
 size 1976469780x32637 max size 1976469780x32637
```

Because of insufficient sanity checking Metacity is now using a garbage value for the window size.

Why did metacity change the min and max size values from the correct ones that were requested?

This is what window managers do. They move and resize windows to fit them all on the screen. The size increment hints are good for things like terminals so they're resized in units corresponding to character sizes instead of pixels. The extra lines show Metacity doing its arithmetic to adjust the min and max sizes modulo the increment hints. But since the hints it received are uninitialized garbage, its adjustment of the max and min window sizes is nonsensical and bad things ensue.

Some number of log lines later, Metacity is trying to act on the outrageously huge window:

```
GEOMETRY: Calculated frame size 1976469782x32666
GEOMETRY: Syncing frame geometry 68,24 1976469782x32666 (SE: 1976469850,32690)
SHAPES: Frame 0x42004b5 has shaped corners
Window manager warning: Log level 16: Native children wider or taller than 65535 pixels are not supported
Window manager:   metacity() [0x4358ca]
Window manager:   /lib/x86_64-linux-gnu/libglib-2.0.so.0(g_logv+0x1b1) [0x7fefbea12ae1]
Window manager:   /lib/x86_64-linux-gnu/libglib-2.0.so.0(g_log+0x82) [0x7fefbea12d72]
Window manager:   /usr/lib/x86_64-linux-gnu/libgdk-x11-2.0.so.0(+0x5b9ea) [0x7fefbfe929ea]
Window manager:   /usr/lib/x86_64-linux-gnu/libgdk-x11-2.0.so.0(+0x67873) [0x7fefbfe9e873]
Window manager:   /usr/lib/x86_64-linux-gnu/libgdk-x11-2.0.so.0(+0x4535e) [0x7fefbfe7c35e]
Window manager:   metacity() [0x449ed0]
Window manager:   metacity() [0x424bfe]
Window manager:   metacity() [0x43ba8e]
Window manager:   metacity() [0x43c320]
Window manager:   metacity() [0x43c398]
Window manager:   /lib/x86_64-linux-gnu/libglib-2.0.so.0(g_main_context_dispatch+0x135) [0x7fefbea0bce5]
Window manager:   /lib/x86_64-linux-gnu/libglib-2.0.so.0(+0x49048) [0x7fefbea0c048]
Window manager:   /lib/x86_64-linux-gnu/libglib-2.0.so.0(g_main_loop_run+0x6a) [0x7fefbea0c30a]
Window manager:   metacity() [0x40faf0]
Window manager:   /lib/x86_64-linux-gnu/libc.so.6(__libc_start_main+0xf5) [0x7fefbc75af45]
Window manager:   metacity() [0x40fecb]
GEOMETRY: Syncing new client geometry 69,52 1976469780x32637, border: false pos: false size: true
GEOMETRY: New size/position 69,52 1976469780x32637 (user 69,52 2490x1363)
```

Somewhere in the process of resizing the new window to the outrageously huge size, function _gdk_x11_window_move_resize_child in file gdkgeometry-x11.c gets called, which does a sanity check that ends up with Metacity getting a callback to function log_handler which prints the "Window manager warning" line and then prints a backtrace.

In tests running Metacity in a terminal window it keeps going, vainly trying to handle the gargantuan window, but only five more lines show up in the log file after the above excerpt when it gets a SIGSEGV and dumps core. The backtrace shows that it was in glibc's memcpy function, specifically __memcpy_sse2_unaligned. The call to memcpy came from libgtk-x11.

On Ubuntu 14.04 with gnome-session-flashback and stock Metacity managed by gnome-session, you're doomed. You'll soon be looking at a login prompt.

```
Sep 14 15:55:21 linuxbox gnome-session[6405]: WARNING: Application 'metacity.desktop' killed by signal 6
Sep 14 15:55:21 linuxbox gnome-session[6405]: WARNING: App 'metacity.desktop' respawning too quickly
Sep 14 15:55:21 linuxbox gnome-session[6405]: CRITICAL: We failed, but the fail whale is dead. Sorry....
```

Not sure why Metacity running under gnome-session gets a SIGABRT instead of a SIGSEGV, but in about a dozen tests under gnome-session that was always the case.

As I said in a prior post, the various window managers react differently in response to the uninitialized values in the struct GdkGeometry passed by Unity 5. So far it appears that some versions of Cinnamon, Gnome-shell, Metacity, and Openbox are vulnerable. Compiz appears to be robust.

The question is why Unity 5 is inappropriately setting bits in the GdkWindowHints argument associated with the struct GdkGeometry it passes to the window managers.

#21 - 09/18/2016 01:37 PM - waterlubber

> Run Full screen. The problem is with running in windowed mode using a window manager with insufficient sanity checking. **Important: which window manager do you use?**

I'm using Openbox 3.5.2
I'm also having trouble getting the game to run fullscreen, even if I edit the settings.cfg and the second one in ~/.config/unity3d

#22 - 09/18/2016 01:43 PM - waterlubber

EDIT:

I got it working with a modified version of psycho's script, most of the changes are in the execution statement!
http://pastebin.com/6Scugme7

#23 - 09/18/2016 04:13 PM - psycho_zs

New iteration of my script.
Down with exec, it now puts KSP in background, clearly getting its PID, avoids potential race conditions, traps signals for cleaning up its children.

And the most shiny new feature - it launches KSP in cycle until it works, Kerbal style! (cycle is repeated only if KSP was killed by geometry guard).
I've just tired to launch it manually 10 times in a row. Now you can launch once and go drink some coffee until you hear the main menu music! :)

(This is for Openbox users of course)

```
#!/bin/bash

WINWIDTH=1366
WINHEIGHT=768
BINARY=KSP.x86_64

# change dir to script location
cd "$(dirname "$(readlink -f "$0")")"

printf "Still can not learn locales while reading settings? setting LC_ALL=C\n"
LANG=C
LC_ALL=C
export LC_ALL
export LANG
unset LANGUAGE

printf "Fixing weird permissions from game archive.\n"
find . -type f -exec chmod 644 "{}" \;
find . -type d -exec chmod 755 "{}" \;
chmod 755 KSP.x86  KSP.x86_64  Launcher.x86  Launcher.x86_64 "${BINARY}" "$(basename "$0")"

trap 'kill $(jobs -p) &>/dev/null' EXIT
trap 'kill $(jobs -p) &>/dev/null' SIGHUP
trap 'kill $(jobs -p) &>/dev/null' SIGINT

GEOMKILL="-1"

until [ "$GEOMKILL" = "0" ]
do
    [ "$GEOMKILL" = "1" ] && printf "\n\nHere we go again...\n\n"
    GEOMKILL="0"

    printf "Launching KSP... Hold on to yourr helmets and pray the Kraken!\n"
    ./"${BINARY}" -screen-fullscreen 0 -screen-width $WINWIDTH -screen-height $WINHEIGHT &
    KSPPID=$!
    printf "Launched with PID ${KSPPID}.\n"

    printf "Bumping KSP OOM killer score in case it goes berserk and chews up all the memory at once.\nCurrent OOM score: $(cat /proc/${KSPPID}/oom_score)\n"
    echo 15 > "/proc/${KSPPID}/oom_adj"
    printf "New OOM score: $(cat /proc/${KSPPID}/oom_score)\n"

    if [ -x "$(which xwininfo)" ]
    then
        printf "Starting window geometry guard...\n"
        STOPDATE="$(date +%s -d "180 seconds")"
        while [ "$(date +%s)" -lt "$STOPDATE" ] && kill -0 "$KSPPID" 2> /dev/null
        do
            WIDTH="$(xwininfo -name "Kerbal Space Program" 2>/dev/null | grep -o 'Width:[[:space:]]*[0-9]\+' | grep -o '[0-9]\+')"
            HEIGHT="$(xwininfo -name "Kerbal Space Program" 2>/dev/null | grep -o 'Height:[[:space:]]*[0-9]\+' | grep -o '[0-9]\+')"
            if [ "${WIDTH:-0}" -gt "$(( $WINWIDTH * 2 ))" -o "${HEIGHT:-0}" -gt "$(( $WINHEIGHT * 2 ))" ]
            then
                printf "Window geometry is freaky (${WIDTH}x${HEIGHT}), killing KSP!\n" >&2
                kill -9 "$KSPPID"
                GEOMKILL=1
                break
            else
                GEOMKILL=0
            fi
            [ "$WIDTH" != "$PREVWIDTH" -o "$HEIGHT" != "$PREVHEIGHT" ] && printf "Window geometry changed from ${PREVWIDTH:-NA}x${PREVHEIGHT:-NA} to ${WIDTH}x${HEIGHT}\n"
            PREVWIDTH=$WIDTH
            PREVHEIGHT=$HEIGHT
            sleep 0.1
        done
        [ "$GEOMKILL" = "0" ] && printf "Window geometry guard no longer needed.\n"
    else
        printf "xwininfo not found, aborting window geometry guard!\n" >&2
    fi
done
```

```
wait
```

#24 - 09/19/2016 10:58 AM - sal_vager

It's not just Openbox that's affected though, I've had reports of this on Gnome, Unity(DE), KDE, Cinnamon and Mate, the only one that isn't doing this is xfce which uses xfwm.

None of the Unity3D tracker issues where Linux crashes have any details, they just state "Unity crashes on Linux".

Edit, Can't find a report button on their tracker, so I crashed the Unity editor (not hard to do) and posted Strategerms info in the crash report box.

#25 - 09/21/2016 08:01 PM - sal_vager

- Related to Bug #11955: Game crashes X Windows when starting added

#26 - 09/22/2016 12:29 PM - AmenRa

I attempted to use psycho_zs script in Gnome Ubuntu 16.04.01 LTS.
I still have the problem of KSP crashing occasionally at startup however the script does appear to prevent the X Windows desktop from terminating.

#27 - 10/07/2016 10:18 AM - psycho_zs

Is that just me or the last couple of builds seem less affected?

#28 - 10/08/2016 04:18 AM - psycho_zs

1569 did not freak out once, 1574 did it 10 times right from the start. Maybe just a coincidence though.

#29 - 10/11/2016 09:59 PM - waterlubber

psycho_zs wrote:

> New iteration of my script.
> Down with exec, it now puts KSP in background, clearly getting its PID, avoids potential race conditions, traps signals for cleaning up its children.
>
> And the most shiny new feature - it launches KSP in cycle until it works, Kerbal style! (cycle is repeated only if KSP was killed by geometry guard). I've just tired to launch it manually 10 times in a row. Now you can launch once and go drink some coffee until you hear the main menu music! :)
>
> (This is for Openbox users of course)
>
> [...]

X server just crashed for me using this script.

I did make one or two modifications, mainly adding -force-gfx-direct and -force-glcore to the exec statement (otherwise it would open and immediately crash, not get killed by the geometry guard). I am using Openbox 3.5.2 and Debian 8 jessie, as well as the KSP release build.

It also threw a segfault in dmesg:
[82217.052097] KSP.x86_64[18564]: segfault at 7f5c7fab7ac8 ip 0000000000f2dd3d sp 00007ffc4f122e10 error 4 in KSP.x86_64[400000+1858000]

I am using fglrx drivers, OpenGL version is 4.5.13399, graphics card is R9 270X.
CPU is overclocked in BIOS but shouldn't be an issue. (It's quite stable)

#30 - 10/11/2016 10:05 PM - waterlubber

I did something very stupid, and very brash, and did get it to work!

Based on a thing I saw in dmesg, I changed oom_adj to oom_score_adj, and then ran the game as root. It worked on the second try.

#31 - 10/12/2016 05:50 AM - Tallone55

The problem gotten even worse upon the full release of KSP 1.2. In the pre-release as late as yesterday (October-10th) the game was working as previously described above, but with the full update release, KSP has ceased running properly **ever** in windowed mode, despite working (inconsistently) before. Fullscreen doesn't cause the bug, so the game is still playable, but **any** attempt to launch the game in windowed mode will cause your DE to crash.

(I realize this is a Unity bug, but I figured I'd post an update on the bug tracker I actually have access to.)

#32 - 10/12/2016 06:21 AM - psycho_zs

Release build launches properly only after 10-15th iteration. Worse than previous.
Also builds of the last week never freaked out on transition between loading screen and main menu (which happened for earlier builds).

Is it absolutely 100% impossible for KSP code to somehow interfere with this behavior?

Can I please get a download link for build 1569? Should've keep it :(I want to test a wild guess.

#33 - 10/15/2016 10:16 PM - waterlubber

I have 1500, 1548 and release. Not sure about the legality for sending KSP versions around although we're all pretty sure you have it

#34 - 12/01/2016 04:52 PM - sal_vager

- Project changed from KSP Pre-Release to Kerbal Space Program

- Category changed from Unity3D to 368

Moving for visibility.

#35 - 01/18/2017 02:33 PM - agoode

See
https://bugzilla.gnome.org/show_bug.cgi?id=765655

#36 - 01/19/2017 10:23 PM - ringerc

You can work around this by running a window manager other than gnome-shell

XFCE, for example, works perfectly.

It's definitely a KSP/Unity bug, but gnome-shell is also insufficiently defensive and crashes out when it should kill the problem client or ignore the request.

#39 - 03/17/2017 01:31 AM - Ruedii

Fixed as of the newest release (seems fixed with the fixes for the resolution in the config.)

Can someone else test on a libgnome based window manager. I tested on KDE that was previously subject to what seemed like this bug.

Remember, delete the settings.cfg and prefs files to verify this on a clean slate.

#40 - 03/17/2017 07:36 AM - psycho_zs

Unfortunately, I can not confirm fix. (
1.2.9 En, windowed mode, Openbox+Compton. It took somewhere about 50 tries to launch it. It's either became worse, or it is just a statistical fluctuation.

#41 - 03/17/2017 10:51 AM - psycho_zs

Refreshed version of my launcher script, now in POSIX, with colored messages and attempt counter.

```
#!/bin/sh

WIN_WIDTH=1366
WIN_HEIGHT=768
BINARY=KSP.x86_64
GEOM_GUARD_TIME="180 seconds"

[ ! -x "$(which xwininfo)" ] && printf "\e[31mWarning: xwininfo not found or is not an executable!\n         G
eometry guard will not work!\e[39m\n" >&2

# change dir to script location
cd "$(dirname "$(readlink -f "$0")")"

printf "Still can not learn localized fraction delimiters? setting LC_ALL=C\n"
LANG=C
LC_ALL=C
export LC_ALL
export LANG
unset LANGUAGE

printf "Fixing weird permissions from game archive.\n"
find . -type d -exec chmod 755 "{}" \;
find . -type f -exec chmod 644 "{}" \;
chmod 755 KSP.x86  KSP.x86_64  Launcher.x86  Launcher.x86_64 "${BINARY}" "$(basename "$0")"

# trap for child process cleanup
trap 'kill $KSPPID 2> /dev/null ; exit' EXIT HUP INT TERM
```

```
GEOMKILL="2"
COUNT="0"

until [ "$GEOMKILL" = "0" ]
do
    [ "$GEOMKILL" = "1" ] && printf "\nHere we go again...\n"

    printf "Clearing unity3d config dir\n"
    [ -d "${XDG_CONFIG_HOME:-$HOME/.config}/unity3d/" ] && rm -r "${XDG_CONFIG_HOME:-$HOME/.config}/unity3d/"

    GEOMKILL="0"
    COUNT=$(( $COUNT + 1 ))

    printf "\n\e[32mLaunching KSP, attempt ${COUNT}... Hold on to your helmets and pray the Kraken!\e[39m\n\n"


    ## multiple variants, some work better than others.
    #./"${BINARY}" -screen-fullscreen 0 -screen-width $WIN_WIDTH -screen-height $WIN_HEIGHT &
    ./"${BINARY}" -screen-fullscreen 0 -popupwindow -screen-width $WIN_WIDTH -screen-height $WIN_HEIGHT &
    #./"${BINARY}" -popupwindow -screen-fullscreen &
    KSPPID=$!

    printf "\e[32mLaunched with PID ${KSPPID}.\e[39m\n\n"

    printf "\e[33mBumping KSP OOM killer score in case it goes berserk and chews up all the memory at once.\nC
urrent OOM score: $(cat /proc/${KSPPID}/oom_score)\e[39m\n"
    echo 15 > "/proc/${KSPPID}/oom_adj"
    printf "\e[33mNew OOM score: $(cat /proc/${KSPPID}/oom_score)\e[39m\n"

    if [ -x "$(which xwininfo)" ]
    then
        printf "Starting window geometry guard...\n"
        STOPDATE="$(date +%s -d "$GEOM_GUARD_TIME")"
        while [ "$(date +%s)" -lt "$STOPDATE" ] && kill -0 "$KSPPID" 2> /dev/null
        do
            GEOM="$(xwininfo -name "Kerbal Space Program" 2>/dev/null)"
            HEIGHT="${GEOM##*Height:}"
            HEIGHT="${GEOM##*[[:space:]]}"
            HEIGHT="${HEIGHT%%[^0-9]*}"
            WIDTH="${GEOM##*Width:}"
            WIDTH="${GEOM##*[[:space:]]}"
            WIDTH="${WIDTH%%[^0-9]*}"

            if [ "${WIDTH:-0}" -gt "$(( $WIN_WIDTH * 2 ))" -o "${HEIGHT:-0}" -gt "$(( $WIN_HEIGHT * 2 ))" -o "
${WIDTH:-10000}" -lt "$(( $WIN_WIDTH / 2 ))" -o "${HEIGHT:-10000}" -lt "$(( $WIN_HEIGHT / 2 ))" ]
            then
                kill -9 "$KSPPID"
                GEOMKILL=1
                printf "\n\e[31mWindow geometry is freaky (${WIDTH:-NA}x${HEIGHT:-NA}), killed KSP!\e[39m\n\n"
 >&2
                sleep 1
                break
            else
                GEOMKILL=0
            fi
            [ "$WIDTH" != "$PREVWIDTH" -o "$HEIGHT" != "$PREVHEIGHT" ] && printf "\e[33mWindow geometry change
d from ${PREVWIDTH:-NA}x${PREVHEIGHT:-NA} to ${WIDTH:-NA}x${HEIGHT:-NA}\e[39m\n"
            PREVWIDTH=$WIDTH
            PREVHEIGHT=$HEIGHT
        done
        [ "$GEOMKILL" = "0" ] && printf "\e[33mWindow geometry guard is (hopefully) no longer needed.\e[39m\n"

    else
        printf "\e[31mxwininfo not found, aborting window geometry guard!\nYou're on your own with OOM.\e[39m\
n" >&2
    fi
done

wait
```

#42 - 03/17/2017 11:20 AM - AmenRa

I can confirm KSP 1.22.1622 x64 is working well in window mode on Ubuntu 16.04.02 x64 running Gnome Shell 3.18.5. However, I don't know if I should attribute the stability to KSP updates or to the Ubuntu upgrade from 16.04 -> 16.04.02 (or both?). Ubuntu 16.04.02 had significant changes to

the hardware stack. You can read about it here:
http://www.omgubuntu.co.uk/2017/02/download-ubuntu-16-04-2-lts

#43 - 03/20/2017 03:41 PM - sal_vager

- Related to Bug #13684: Moving the KSP window breaks the resolution added

#44 - 03/21/2017 07:24 AM - Ruedii

This is now fixed as far as I can tell.

#45 - 03/21/2017 09:33 AM - psycho_zs

@Ruedii
Which window manager are you running?

#46 - 03/21/2017 10:41 AM - sal_vager

Update, unity have responded to another report of this issue, and have taken steps in newer Unity versions that might solve this, but are as of yet untested.

https://forum.unity3d.com/threads/screen-setresolution-ignores-window-dimensions-on-linux.457606/

#47 - 03/21/2017 11:01 AM - Squelch

- Status changed from Updated to Unity Bug

- % Done changed from 10 to 100

Tests conducted with empty unity projects, and with KSP reveal that the window size can randomly be set to bizarre values. This affects some window managers more than others, and can either crash the application, or even the whole WM. Unity technologies appear to have acknowledge the problem, and it is fixed upstream with a change to using SDL libraries.

We are exploring a workaround in the meantime.

#48 - 03/21/2017 11:14 AM - Squelch

- Has duplicate Bug #14344: KSP randomly overwrites Unity resolution with 100 added

#49 - 03/27/2017 12:39 PM - psycho_zs

Tried latest build. It is fixed now. Unity player's size restriction is completely turned off now (like it was before 1.1). This bug seems to be confined to restriction mechanism, so it is gone too. As a bonus, window is resizeable again.
I tried moving and resizing the window like mad for a whole minute, no casualties.

Kudos to sal_vager !

Files

KSP.log	196 KB	09/14/2016	psycho_zs
KSP.log_x_windows_crash	536 Bytes	09/14/2016	Stratagerm