```
<!-- Generated SVGOMG Version 2.1.1 -->
<svg viewBox="
<defs id="defs">
<clipPath id="clip1">
<g id="groups_vectorpaths">
<g id="Layer_1">
<g id="Layer_2">
<g id="Layer_3">
<g id="Layer_4">
<g id="Layer_5">
<g id="Layer_6">
<g id="Layer_7">
<g id="Layer_8">
<g id="Layer_9">
<g id="Layer_10">
<g id="Layer_11">
<g id="Layer_12">
<g id="Layer_13">
<g id="Layer_14">
<g id="Layer_15">
<g id="Layer_16">
<g id="Layer_17">
<g id="Layer_18">
<g id="Layer_19">
<g id="Layer_20">
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0" xmlns:dc="http://purl.org/dc/elements/1.0/" xmlns:sy="http://
purl.org/rss/1.0/modules/syndication/" xmlns:slash="http://
purl.org/rss/1.0/modules/slash/" xmlns:wfw="http://wellformedweb.org/CommentAPI/"
xmlns:feedburner="http://rss.netsol.com/" >
<channel>
<title>Ultimate Guitar - All Rock the short square for a song</
title>
<atom:link rel="self" type="application/rss+
xml" />
<description>Ultimate Guitar - All Rock the short square for a song</
description>
<lastBuildDate>Fri, 31 Dec 2021 00:00:00 +0000</
lastBuildDate>
<generator_name_guitar_pro_version_5_or_6_or_7_or_8_or_9_or_10_or_11_or_12_or_
13_or_14_or_15_or_16_or_17_or_18_or_19_or_20_or_21_or_22_or_23_or_24_or_25_or_
26_or_27_or_28_or_29_or_30_or_31_or_32_or_33_or_34_or_35_or_36_or_37_or_38_or_
39_or_40_or_41_or_42_or_43_or_44_or_45_or_46_or_47_or_48_or_49_or_50_or_51_or_
52_or_53_or_54_or_55_or_56_or_57_or_58_or_59_or_60_or_61_or_62_or_63_or_64_or_
65_or_66_or_67_or_68_or_69_or_70_or_71_or_72_or_73_or_74_or_75_or_76_or_77_or_
78_or_79_or_80_or_81_or_82_or_83_or_84_or_85_or_86_or_87_or_88_or_89_or_90_or_
91_or_92_or_93_or_94_or_95_or_96_or_97_or_98_or_99_or_100_or_101_or_102_or_103_
or_104_or_105_or_106_or_107_or_108_or_109_or_110_or_111_or_112_or_113_or_114_or_
115_or_116_or_117_or_118_or_119_or_120_or_121_or_122_or_123_or_124_or_125_or_
126_or_127_or_128_or_129_or_130_or_131_or_132_or_133_or_134_or_135_or_136_or_
137_or_138_or_139_or_140_or_141_or_142_or_143_or_144_or_145_or_146_or_147_or_
148_or_149_or_150_or_151_or_152_or_153_or_154_or_155_or_156_or_157_or_158_or_
159_or_160_or_161_or_162_or_163_or_164_or_165_or_166_or_167_or_168_or_169_or_
170_or_171_or_172_or_173_or_174_or_175_or_176_or_177_or_178_or_179_or_180_or_
181_or_182_or_183_or_184_or_185_or_186_or_187_or_188_or_189_or_190_or_191_or_
192_or_193_or_194_or_195_or_196_or_197_or_198_or_199_or_200_or_201_or_202_or_
203_or_204_or_205_or_206_or_207_or_208_or_209_or_210_or_211_or_212_or_213_or_
214_or_215_or_216_or_217_or_218_or_219_or_220_or_221_or_222_or_223_or_224_or_
225_or_226_or_227_or_228_or_229_or_230_or_231_or_232_or_233_or_234_or_235_or_
236_or_237_or_238_or_239_or_240_or_241_or_242_or_243_or_244_or_245_or_246_or_
247_or_248_or_249_or_250_or_251_or_252_or_253_or_254_or_255_or_256_or_257_or_
258_or_259_or_260_or_261_or_262_or_263_or_264_or_265_or_266_or_267_or_268_or_
269_or_270_or_271_or_272_or_273_or_274_or_275_or_276_or_277_or_278_or_279_or_
280_or_281_or_282_or_283_or_284_or_285_or_286_or_287_or_288_or_289_or_290_or_
291_or_292_or_293_or_294_or_295_or_296_or_297_or_298_or_299_or_300_or_301_or_
302_or_303_or_304_or_305_or_306_or_307_or_308_or_309_or_310_or_311_or_312_or_
313_or_314_or_315_or_316_or_317_or_318_or_319_or_320_or_321_or_322_or_323_or_
324_or_325_or_326_or_327_or_328_or_329_or_330_or_331_or_332_or_333_or_334_or_
335_or_336_or_337_or_338_or_339_or_340_or_341_or_342_or_343_or_344_or_345_or_
346_or_347_or_348_or_349_or_350_or_351_or_352_or_353_or_354_or_355_or_356_or_
357_or_358_or_359_or_360_or_361_or_362_or_363_or_364_or_365_or_366_or_367_or_
368_or_369_or_370_or_371_or_372_or_373_or_374_or_375_or_376_or_377_or_378_or_
379_or_380_or_381_or_382_or_383_or_384_or_385_or_386_or_387_or_388_or_389_or_
390_or_391_or_392_or_393_or_394_or_395_or_396_or_397_or_398_or_399_or_400_or_
401_or_402_or_403_or_404_or_405_or_406_or_407_or_408_or_409_or_410_or_411_or_
412_or_413_or_414_or_415_or_416_or_417_or_418_or_419_or_420_or_421_or_422_or_
423_or_424_or_425_or_426_or_427_or_428_or_429_or_430_or_431_or_432_or_433_or_
434_or_435_or_436_or_437_or_438_or_439_or_440_or_441_or_442_or_443_or_444_or_
445_or_446_or_447_or_448_or_449_or_450_or_451_or_452_or_453_or_454_or_455_or_
456_or_457_or_458_or_459_or_460_or_461_or_462_or_463_or_464_or_465_or_466_or_
467_or_468_or_469_or_470_or_471_or_472_or_473_or_474_or_475_or_476_or_477_or_
478_or_479_or_480_or_481_or_482_or_483_or_484_or_485_or_486_or_487_or_488_or_
489_or_490_or_491_or_492_or_493_or_494_or_495_or_496_or_497_or_498_or_499_or_
500_or_501_or_502_or_503_or_504_or_505_or_506_or_507_or_508_or_509_or_510_or_
511_or_512_or_513_or_514_or_515_or_516_or_517_or_518_or_519_or_520_or_521_or_
522_or_523_or_524_or_525_or_526_or_527_or_528_or_529_or_530_or_531_or_532_or_
533_or_534_or_535_or_536_or_537_or_538_or_539_or_540_or_541_or_542_or_543_or_
544_or_545_or_546_or_547_or_548_or_549_or_550_or_551_or_552_or_553_or_554_or_
555_or_556_or_557_or_558_or_559_or_560_or_561_or_562_or_563_or_564_or_565_or_
566_or_567_or_568_or_569_or_570_or_571_or_572_or_573_or_574_or_575_or_576_or_
577_or_578_or_579_or_580_or_581_or_582_or_583_or_584_or_585_or_586_or_587_or_
588_or_589_or_590_or_591_or_592_or_593_or_594_or_595_or_596_or_597_or_598_or_
599_or_600_or_601_or_602_or_603_or_604_or_605_or_606_or_607_or_608_or_609_or_
610_or_611_or_612_or_613_or_614_or_615_or_616_or_617_or_618_or_619_or_620_or_
621_or_622_or_623_or_624_or_625_or_626_or_627_or_628_or_629_or_630_or_631_or_
632_or_633_or_634_or_635_or_636_or_637_or_638_or_639_or_640_or_641_or_642_or_
643_or_644_or_645_or_646_or_647_or_648_or_649_or_650_or_651_or_652_or_653_or_
654_or_655_or_656_or_657_or_658_or_659_or_660_or_661_or_662_or_663_or_664_or_
665_or_666_or_667_or_668_or_669_or_670_or_671_or_672_or_673_or_674_or_675_or_
676_or_677_or_678_or_679_or_680_or_681_or_682_or_683_or_684_or_685_or_686_or_
687_or_688_or_689_or_690_or_691_or_692_or_693_or_694_or_695_or_696_or_697_or_
698_or_699_or_700_or_701_or_702_or_703_or_704_or_705_or_706_or_707_or_708_or_
709_or_710_or_711_or_712_or_713_or_714_or_715_or_716_or_717_or_718_or_719_or_
720_or_721_or_722_or_723_or_724_or_725_or_726_or_727_or_728_or_729_or_730_or_
731_or_732_or_733_or_734_or_735_or_736_or_737_or_738_or_739_or_740_or_741_or_
742_or_743_or_744_or_745_or_746_or_747_or_748_or_749_or_750_or_751_or_752_or_
753_or_754_or_755_or_756_or_757_or_758_or_759_or_760_or_761_or_762_or_763_or_
764_or_765_or_766_or_767_or_768_or_769_or_770_or_771_or_772_or_773_or_774_or_
775_or_776_or_777_or_778_or_779_or_780_or_781_or_782_or_783_or_784_or_785_or_
786_or_787_or_788_or_789_or_790_or_791_or_792_or_793_or_794_or_795_or_796_or_
797_or_798_or_799_or_800_or_801_or_802_or_803_or_804_or_805_or_806_or_807_or_
808_or_809_or_810_or_811_or_812_or_813_or_814_or_815_or_816_or_817_or_818_or_
819_or_820_or_821_or_822_or_823_or_824_or_825_or_826_or_827_or_828_or_829_or_
830_or_831_or_832_or_833_or_834_or_835_or_836_or_837_or_838_or_839_or_840_or_
841_or_842_or_843_or_844_or_845_or_846_or_847_or_848_or_849_or_850_or_851_or_
852_or_853_or_854_or_855_or_856_or_857_or_858_or_859_or_860_or_861_or_862_or_
863_or_864_or_865_or_866_or_867_or_868_or_869_or_870_or_871_or_872_or_873_or_
874_or_875_or_876_or_877_or_878_or_879_or_880_or_881_or_882_or_883_or_884_or_
885_or_886_or_887_or_888_or_889_or_890_or_891_or_892_or_893_or_894_or_895_or_
896_or_897_or_898_or_899_or_900_or_901_or_902_or_903_or_904_or_905_or_906_or_
907_or_908_or_909_or_910_or_911_or_912_or_913_or_914_or_915_or_916_or_917_or_
918_or_919_or_920_or_921_or_922_or_923_or_924_or_925_or_926_or_927_or_928_or_
929_or_930_or_931_or_932_or_933_or_934_or_935_or_936_or_937_or_938_or_939_or_
940_or_941_or_942_or_943_or_944_or_945_or_946_or_947_or_948_or_949_or_950_or_
951_or_952_or_953_or_954_or_955_or_956_or_957_or_958_or_959_or_960_or_961_or_
962_or_963_or_964_or_965_or_966_or_967_or_968_or_969_or_970_or_971_or_972_or_
973_or_974_or_975_or_976_or_977_or_978_or_979_or_980_or_981_or_982_or_983_or_
984_or_985_or_986_or_987_or_988_or_989_or_990_or_991_or_992_or_993_or_994_or_
995_or_996_or_997_or_998_or_999_or_1000_or_1001_or_1002_or_1003_or_1004_or_
1005_or_1006_or_1007_or_1008_or_1009_or_1010_or_1011_or_1012_or_1013_or_1014_
or_1015_or_1016_or_1017_or_1018_or_1019_or_1020_or_1021_or_1022_or_1023_or_
1024_or_1025_or_1026_or_1027_or_1028_or_1029_or_1030_or_1031_or_1032_or_1033_
or_1034_or_1035_or_1036_or_1037_or_1038_or_1039_or_1040_or_1041_or_1042_or_
1043_or_1044_or_1045_or_1046_or_1047_or_1048_or_1049_or_1050_or_1051_or_1052_
or_1053_or_1054_or_1055_or_1056_or_1057_or_1058_or_1059_or_1060_or_1061_or_
1062_or_1063_or_1064_or_1065_or_1066_or_1067_or_1068_or_1069_or_1070_or_1071_
or_1072_or_1073_or_1074_or_1075_or_1076_or_1077_or_1078_or_1079_or_1080_or_
1081_or_1082_or_1083_or_1084_or_1085_or_1086_or_1087_or_1088_or_1089_or_1090_
or_1091_or_1092_or_1093_or_1094_or_1095_or_1096_or_1097_or_1098_or_1099_or_
1100_or_1101_or_1102_or_1103_or_1104_or_1105_or_1106_or_1107_or_1108_or_1109_
or_1110_or_1111_or_1112_or_1113_or_1114_or_1115_or_1116_or_1117_or_1118_or_
1119_or_1120_or_1121_or_1122_or_1123_or_1124_or_1125_or_1126_or_1127_or_1128_
or_1129_or_1130_or_1131_or_1132_or_1133_or_1134_or_1135_or_1136_or_1137_or_
1138_or_1139_or_1140_or_1141_or_1142_or_1143_or_1144_or_1145_or_1146_or_1147_
or_1148_or_1149_or_1150_or_1151_or_1152_or_1153_or_1154_or_1155_or_1156_or_
1157_or_1158_or_1159_or_1160_or_1161_or_1162_or_1163_or_1164_or_1165_or_1166_
or_1167_or_1168_or_1169_or_1170_or_1171_or_1172_or_1173_or_1174_or_1175_or_
1176_or_1177_or_1178_or_1179_or_1180_or_1181_or_1182_or_1183_or_1184_or_1185_
or_1186_or_1187_or_1188_or_1189_or_1190_or_1191_or_1192_or_1193_or_1194_or_
1195_or_1196_or_1197_or_1198_or_1199_or_1200_or_1201_or_1202_or_1203_or_1204_
or_1205_or_1206_or_1207_or_1208_or_1209_or_1210_or_1211_or_1212_or_1213_or_
1214_or_1215_or_1216_or_1217_or_1218_or_1219_or_1220_or_1221_or_1222_or_1223_
or_1224_or_1225_or_1226_or_1227_or_1228_or_1229_or_1230_or_1231_or_1232_or_
1233_or_1234_or_1235_or_1236_or_1237_or_1238_or_1239_or_1240_or_1241_or_1242_
or_1243_or_1244_or_1245_or_1246_or_1247_or_1248_or_1249_or_1250_or_1251_or_
1252_or_1253_or_1254_or_1255_or_1256_or_1257_or_1258_or_1259_or_1260_or_1261_
or_1262_or_1263_or_1264_or_1265_or_1266_or_1267_or_1268_or_1269_or_1270_or_
1271_or_1272_or_1273_or_1274_or_1275_or_1276_or_1277_or_1278_or_1279_or_1280_
or_1281_or_1282_or_1283_or_1284_or_1285_or_1286_or_1287_or_1288_or_1289_or_
1290_or_1291_or_1292_or_1293_or_1294_or_1295_or_1296_or_1297_or_1298_or_1299_
or_1300_or_1301_or_1302_or_1303_or_1304_or_1305_or_1306_or_1307_or_1308_or_
1309_or_1310_or_1311_or_1312_or_1313_or_1314_or_1315_or_1316_or_1317_or_1318_
or_1319_or_1320_or_1321_or_1322_or_1323_or_1324_or_1325_or_1326_or_1327_or_
1328_or_1329_or_1330_or_1331_or_1332_or_1333_or_1334_or_1335_or_1336_or_1337_
or_1338_or_1339_or_1340_or_1341_or_1342_or_1343_or_1344_or_1345_or_1346_or_
1347_or_1348_or_1349_or_1350_or_1351_or_1352_or_1353_or_1354_or_1355_or_1356_
or_1357_or_1358_or_1359_or_1360_or_1361_or_1362_or_1363_or_1364_or_1365_or_
1366_or_1367_or_1368_or_1369_or_1370_or_1371_or_1372_or_1373_or_1374_or_1375_
or_1376_or_1377_or_1378_or_1379_or_1380_or_1381_or_1382_or_1383_or_1384_or_
```

```
/*!
* jqueryui.com | Copyright 2012 jquery.com
* Dual licensed under the MIT or GPL Version 2 licenses.
*/

/*!
* jquery.ui.core.min.js
*/

/*!
* jquery.ui.widget.min.js
*/

/*!
* jquery.ui.mouse.min.js
*/

/*!
* jquery.ui.draggable.min.js
*/

/*!
* jquery.ui.resizable.min.js
*/

/*!
* jquery.ui.selectable.min.js
*/

/*!
* jquery.ui.sortable.min.js
*/

/*!
* jquery.ui.accordion.min.js
*/

/*!
* jquery.ui.autocomplete.min.js
*/

/*!
* jquery.ui.button.min.js
*/

/*!
* jquery.ui.dialog.min.js
*/

/*!
* jquery.ui.slider.min.js
*/

/*!
* jquery.ui.tabs.min.js
*/

/*!
* jquery.ui.datepicker.min.js
*/

/*!
* jquery.ui.progressbar.min.js
*/

/*!
* jquery.ui.effects.core.min.js
*/

/*!
* jquery.ui.effects.blind.min.js
*/

/*!
* jquery.ui.effects.bounce.min.js
*/

/*!
* jquery.ui.effects.clip.min.js
*/

/*!
* jquery.ui.effects.drop.min.js
*/

/*!
* jquery.ui.effects.explode.min.js
*/

/*!
* jquery.ui.effects.fade.min.js
*/

/*!
* jquery.ui.effects.fold.min.js
*/

/*!
* jquery.ui.effects.highlight.min.js
*/

/*!
* jquery.ui.effects.pulsate.min.js
*/

/*!
* jquery.ui.effects.scale.min.js
*/

/*!
* jquery.ui.effects.shake.min.js
*/

/*!
* jquery.ui.effects.slide.min.js
*/

/*!
* jquery.ui.effects.transfer.min.js
*/

/*!
* jquery.ui.position.min.js
*/

/*!
* jquery.ui.dialog.extend.min.js
*/

/*!
* jquery.ui.tooltip.min.js
*/

/*!
* jquery.ui.menu.min.js
*/

/*!
* jquery.ui.spinner.min.js
*/

/*!
* jquery.ui.tabs.ajax.min.js
*/

/*!
* jquery.ui.tabs.length.min.js
*/

/*!
* jquery.ui.tabs.scrollable.min.js
*/

/*!
* jquery.ui.tabs.sortable.min.js
*/

/*!
* jquery.ui.tabs.closable.min.js
*/

/*!
* jquery.ui.tabs.rotate.min.js
*/

/*!
* jquery.ui.tabs.hash.min.js
*/

/*!
* jquery.ui.tabs.bookmarkable.min.js
*/

/*!
* jquery.ui.tabs.deletable.min.js
*/

/*!
* jquery.ui.tabs.navigation.min.js
*/

/*!
* jquery.ui.tabs.paging.min.js
*/

/*!
* jquery.ui.tabs.persist.min.js
*/

/*!
* jquery.ui.tabs.spinner.min.js
*/

/*!
* jquery.ui.tabs.state.min.js
*/

/*!
* jquery.ui.tabs.template.min.js
*/

/*!
* jquery.ui.tabs.vertical.min.js
*/

/*!
* jquery.ui.tabs.overflow.min.js
*/

/*!
* jquery.ui.tabs.draggable.min.js
*/

/*!
* jquery.ui.tabs.dropdown.min.js
*/

/*!
* jquery.ui.tabs.forceheight.min.js
*/

/*!
* jquery.ui.tabs.hover.min.js
*/

/*!
* jquery.ui.tabs.mouseover.min.js
*/

/*!
* jquery.ui.tabs.preloading.min.js
*/

/*!
* jquery.ui.tabs.remote.min.js
*/

/*!
* jquery.ui.tabs.scroll.min.js
*/

/*!
* jquery.ui.tabs.selectable.min.js
*/

/*!
* jquery.ui.tabs.show.min.js
*/

/*!
* jquery.ui.tabs.slide.min.js
*/

/*!
* jquery.ui.tabs.slideshow.min.js
*/

/*!
* jquery.ui.tabs.sortable.min.js
*/

/*!
* jquery.ui.tabs.tooltip.min.js
*/

/*!
* jquery.ui.tabs.transition.min.js
*/

/*!
* jquery.ui.tabs.vertical.min.js
*/

/*!
* jquery.ui.tabs.width.min.js
*/

/*!
* jquery.ui.tabs.zoom.min.js
*/

/*!
* jquery.ui.tabs.zoomin.min.js
*/

/*!
* jquery.ui.tabs.zoomout.min.js
*/

/*!
* jquery.ui.tabs.zoomtoggle.min.js
*/

/*!
* jquery.ui.tabs.zoomtoggleout
```

```
// get the input data space //
var inputDataSpace = d3.select("#input_tarpaulins")

// find the input data space bounds
var inputDataSpaceBounds = inputDataSpace.node().getBBox();

// hide the material groups in the input activity //
d3.select("#input_tarpaulins").transition().duration(500).
  attr("opacity", 0);

// find the groups of material items //
var inputMaterialGroups = d3.select("#input_tarpaulins").
  selectAll("g");

// create the material groups in the input activity //
var inputMaterialGroupsEnter = inputMaterialGroups.enter().
  append("g").
  attr("id", function(d) { return "input_tarpaulins_" + d.id; }).
  attr("opacity", 0);

// find the input material groups //
var inputMaterialGroupsUpdate = inputMaterialGroups.
  transition().duration(500).
  attr("opacity", 1);

// create the input material groups //
var inputMaterialGroupsEnterTransition = inputMaterialGroupsEnter.
  transition().duration(500).
  attr("opacity", 1);

// create the input material groups //
var inputMaterialGroupsExit = inputMaterialGroups.exit().
  transition().duration(500).
  attr("opacity", 0).
  remove();

// find the input material groups //
var inputMaterialGroupsAll = d3.select("#input_tarpaulins").
  selectAll("g");

// find the input material groups //
var inputMaterialGroupsAllTransition = inputMaterialGroupsAll.
  transition().duration(500).
  attr("opacity", 1);

// find the input material groups //
var inputMaterialGroupsAllExit = inputMaterialGroupsAll.exit().
  transition().duration(500).
  attr("opacity", 0).
  remove();

// find the input material groups //
var inputMaterialGroupsAllEnter = inputMaterialGroupsAll.enter().
  append("g").
  attr("id", function(d) { return "input_tarpaulins_" + d.id; }).
  attr("opacity", 0);

// find the input material groups //
var inputMaterialGroupsAllEnterTransition = inputMaterialGroupsAllEnter.
  transition().duration(500).
  attr("opacity", 1);

// find the input material groups //
var inputMaterialGroupsAllExitTransition = inputMaterialGroupsAll.exit().
  transition().duration(500).
  attr("opacity", 0).
  remove();

// find the input material groups //
var inputMaterialGroupsAllEnterTransitionTransition = inputMaterialGroupsAllEnter.
  transition().duration(500).
  attr("opacity", 1);

// find the input material groups //
var inputMaterialGroupsAllExitTransitionTransition = inputMaterialGroupsAll.exit().
  transition().duration(500).
  attr("opacity", 0).
  remove();

// find the input material groups //
var inputMaterialGroupsAllEnterTransitionTransitionTransition = inputMaterialGroupsAllEnter.
  transition().duration(500).
  attr("opacity", 1);

// find the input material groups //
var inputMaterialGroupsAllExitTransitionTransitionTransition = inputMaterialGroupsAll.exit().
  transition().duration(500).
  attr("opacity", 0).
  remove();

// find the input material groups //
var inputMaterialGroupsAllEnterTransitionTransitionTransitionTransition = inputMaterialGroupsAllEnter.
  transition().duration(500).
  attr("opacity", 1);

// find the input material groups //
var inputMaterialGroupsAllExitTransitionTransitionTransitionTransition = inputMaterialGroupsAll.exit().
  transition().duration(500).
  attr("opacity", 0).
  remove();

// find the input material groups //
var inputMaterialGroupsAllEnterTransitionTransitionTransitionTransitionTransition = inputMaterialGroupsAllEnter.
  transition().duration(500).
  attr("opacity", 1);

// find the input material groups //
var inputMaterialGroupsAllExitTransitionTransitionTransitionTransitionTransition = inputMaterialGroupsAll.exit().
  transition().duration(500).
  attr("opacity", 0).
  remove();

// find the input material groups //
var inputMaterialGroupsAllEnterTransitionTransitionTransitionTransitionTransitionTransition = inputMaterialGroupsAllEnter.
  transition().duration(500).
  attr("opacity", 1);

// find the input material groups //
var inputMaterialGroupsAllExitTransitionTransitionTransitionTransitionTransitionTransition = inputMaterialGroupsAll.exit().
  transition().duration(500).
  attr("opacity", 0).
  remove();

// find the input material groups //
var inputMaterialGroupsAllEnterTransitionTransitionTransitionTransitionTransitionTransitionTransition = inputMaterialGroupsAllEnter.
  transition().duration(500).
  attr("opacity", 1);

// find the input material groups //
var inputMaterialGroupsAllExitTransitionTransitionTransitionTransitionTransitionTransitionTransition = inputMaterialGroupsAll.exit().
  transition().duration(500).
  attr("opacity", 0).
  remove();

// find the input material groups //
var inputMaterialGroupsAllEnterTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransition = inputMaterialGroupsAllEnter.
  transition().duration(500).
  attr("opacity", 1);

// find the input material groups //
var inputMaterialGroupsAllExitTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransition = inputMaterialGroupsAll.exit().
  transition().duration(500).
  attr("opacity", 0).
  remove();

// find the input material groups //
var inputMaterialGroupsAllEnterTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransition = inputMaterialGroupsAllEnter.
  transition().duration(500).
  attr("opacity", 1);

// find the input material groups //
var inputMaterialGroupsAllExitTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransition = inputMaterialGroupsAll.exit().
  transition().duration(500).
  attr("opacity", 0).
  remove();

// find the input material groups //
var inputMaterialGroupsAllEnterTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransition = inputMaterialGroupsAllEnter.
  transition().duration(500).
  attr("opacity", 1);

// find the input material groups //
var inputMaterialGroupsAllExitTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransition = inputMaterialGroupsAll.exit().
  transition().duration(500).
  attr("opacity", 0).
  remove();

// find the input material groups //
var inputMaterialGroupsAllEnterTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransition = inputMaterialGroupsAllEnter.
  transition().duration(500).
  attr("opacity", 1);

// find the input material groups //
var inputMaterialGroupsAllExitTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransition = inputMaterialGroupsAll.exit().
  transition().duration(500).
  attr("opacity", 0).
  remove();

// find the input material groups //
var inputMaterialGroupsAllEnterTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransition = inputMaterialGroupsAllEnter.
  transition().duration(500).
  attr("opacity", 1);

// find the input material groups //
var inputMaterialGroupsAllExitTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransition = inputMaterialGroupsAll.exit().
  transition().duration(500).
  attr("opacity", 0).
  remove();

// find the input material groups //
var inputMaterialGroupsAllEnterTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransition = inputMaterialGroupsAllEnter.
  transition().duration(500).
  attr("opacity", 1);

// find the input material groups //
var inputMaterialGroupsAllExitTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransition = inputMaterialGroupsAll.exit().
  transition().duration(500).
  attr("opacity", 0).
  remove();

// find the input material groups //
var inputMaterialGroupsAllEnterTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransition = inputMaterialGroupsAllEnter.
  transition().duration(500).
  attr("opacity", 1);

// find the input material groups //
var inputMaterialGroupsAllExitTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransition = inputMaterialGroupsAll.exit().
  transition().duration(500).
  attr("opacity", 0).
  remove();

// find the input material groups //
var inputMaterialGroupsAllEnterTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransition = inputMaterialGroupsAllEnter.
  transition().duration(500).
  attr("opacity", 1);

// find the input material groups //
var inputMaterialGroupsAllExitTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransition = inputMaterialGroupsAll.exit().
  transition().duration(500).
  attr("opacity", 0).
  remove();

// find the input material groups //
var inputMaterialGroupsAllEnterTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransition = inputMaterialGroupsAllEnter.
  transition().duration(500).
  attr("opacity", 1);

// find the input material groups //
var inputMaterialGroupsAllExitTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransition = inputMaterialGroupsAll.exit().
  transition().duration(500).
  attr("opacity", 0).
  remove();

// find the input material groups //
var inputMaterialGroupsAllEnterTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransition = inputMaterialGroupsAllEnter.
  transition().duration(500).
  attr("opacity", 1);

// find the input material groups //
var inputMaterialGroupsAllExitTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransition = inputMaterialGroupsAll.exit().
  transition().duration(500).
  attr("opacity", 0).
  remove();

// find the input material groups //
var inputMaterialGroupsAllEnterTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransition = inputMaterialGroupsAllEnter.
  transition().duration(500).
  attr("opacity", 1);

// find the input material groups //
var inputMaterialGroupsAllExitTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransition = inputMaterialGroupsAll.exit().
  transition().duration(500).
  attr("opacity", 0).
  remove();

// find the input material groups //
var inputMaterialGroupsAllEnterTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransition = inputMaterialGroupsAllEnter.
  transition().duration(500).
  attr("opacity", 1);

// find the input material groups //
var inputMaterialGroupsAllExitTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransitionTransition = inputMaterialGroupsAll.exit().
  transition().duration(500).
  attr("opacity", 0).
  remove;
```

```
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECK=@@UNIQUE_CHECK, UNIQUE_CHECK=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

/*!40000 DROP TABLE IF EXISTS `ad_ad_tag`;
/*!40101 SET @saved_cmd_line = 'mysqldump --port=3306 --host=127.0.0.1 --user=root --password=* --ignore-table=mysqldump.slow_log --ignore-table=mysqldump.general_log --ignore-table=mysqldump.gtid_executed --ignore-table=mysqldump.innodb_index_stats --ignore-table=mysqldump.innodb_table_stats --single-transaction --quick --lock-tables=false --set-gtid-purged=OFF --hex-blob --triggers --routines --events --ignore-table=mysql.ndb_binlog_index --ignore-table=mysql.slow_log --ignore-table=mysql.general_log --ignore-table=mysql.gtid_executed --ignore-table=mysql.innodb_index_stats --ignore-table=mysql.innodb_table_stats --ignore-table=mysql.server_cost --ignore-table=mysql.engine_cost --ignore-table=mysql.proxies_priv --ignore-table=mysql.time_zone --ignore-table=mysql.time_zone_name --ignore-table=mysql.time_zone_transition --ignore-table=mysql.time_zone_transition_type --ignore-table=mysql.help_topic --ignore-table=mysql.help_category --ignore-table=mysql.help_relation --ignore-table=mysql.help_keyword --ignore-table=mysql.plugin --ignore-table=mysql.servers --ignore-table=mysql.tables_priv --ignore-table=mysql.columns_priv --ignore-table=mysql.procs_priv --ignore-table=mysql.proxies_priv
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0" xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:sy="http://purl.org/rss/1.0/modules/syndication/" xmlns:slash="http://
<channel>
<title>Ars Technica - All content</title>
<description>The latest from Ars Technica</description>
<lastBuildDate>Wed, 01 Jan 2025 00:00:00 +0000</lastBuildDate>
<language>en-us</language>
<sy:updatePeriod>hourly</sy:updatePeriod>
<sy:updateFrequency>1</sy:updateFrequency>
<generator>https://wordpress.com/feed/?v=6.8.3-wpcomsh-1</generator>
<copyright>Copyright 2025 Ars Technica, LLC</copyright>
<image>
<title>Ars Technica - All content</title>
<link>https://arstechnica.com/feed/?type=all&page=1</link>

</image>
<item>
<guid isPermaLink="false">https://arstechnica.com/?p=2000000</guid>
<link>https://arstechnica.com/gadgets/2025/01/example-article-one/</link>
<pubDate>Wed, 01 Jan 2025 00:00:00 +0000</pubDate>
<dc:creator>Example Author</dc:creator>
<category>Gadgets</category>
<category>Hardware</category>
<category>Reviews</category>
<category>Example Tag</category>
<category>Another Tag</category>
<category>Third Tag</category>
<category>Fourth Tag</category>
<category>Fifth Tag</category>
<category>Sixth Tag</category>
<category>Seventh Tag</category>
<category>Eighth Tag</category>
<category>Ninth Tag</category>
<category>Tenth Tag</category>
<category>Eleventh Tag</category>
<category>Twelfth Tag</category>
<category>Thirteenth Tag</category>
<category>Fourteenth Tag</category>
<category>Fifteenth Tag</category>
<category>Sixteenth Tag</category>
<category>Seventeenth Tag</category>
<category>Eighteenth Tag</category>
<category>Nineteenth Tag</category>
<category>Twentieth Tag</category>
<category>Twenty-first Tag</category>
<category>Twenty-second Tag</category>
<category>Twenty-third Tag</category>
<category>Twenty-fourth Tag</category>
<category>Twenty-fifth Tag</category>
<category>Twenty-sixth Tag</category>
<category>Twenty-seventh Tag</category>
<category>Twenty-eighth Tag</category>
<category>Twenty-ninth Tag</category>
<category>Thirty-first Tag</category>
<category>Thirty-second Tag</category>
<category>Thirty-third Tag</category>
<category>Thirty-fourth Tag</category>
<category>Thirty-fifth Tag</category>
<category>Thirty-sixth Tag</category>
<category>Thirty-seventh Tag</category>
<category>Thirty-eighth Tag</category>
<category>Thirty-ninth Tag</category>
<category>Forty-first Tag</category>
<category>Forty-second Tag</category>
<category>Forty-third Tag</category>
<category>Forty-fourth Tag</category>
<category>Forty-fifth Tag</category>
<category>Forty-sixth Tag</category>
<category>Forty-seventh Tag</category>
<category>Forty-eighth Tag</category>
<category>Forty-ninth Tag</category>
<category>Fifty-first Tag</category>
<category>Fifty-second Tag</category>
<category>Fifty-third Tag</category>
<category>Fifty-fourth Tag</category>
<category>Fifty-fifth Tag</category>
<category>Fifty-sixth Tag</category>
<category>Fifty-seventh Tag</category>
<category>Fifty-eighth Tag</category>
<category>Fifty-ninth Tag</category>
<category>Sixty-first Tag</category>
<category>Sixty-second Tag</category>
<category>Sixty-third Tag</category>
<category>Sixty-fourth Tag</category>
<category>Sixty-fifth Tag</category>
<category>Sixty-sixth Tag</category>
<category>Sixty-seventh Tag</category>
<category>Sixty-eighth Tag</category>
<category>Sixty-ninth Tag</category>
<category>Seventy-first Tag</category>
<category>Seventy-second Tag</category>
<category>Seventy-third Tag</category>
<category>Seventy-fourth Tag</category>
<category>Seventy-fifth Tag</category>
<category>Seventy-sixth Tag</category>
<category>Seventy-seventh Tag</category>
<category>Seventy-eighth Tag</category>
<category>Seventy-ninth Tag</category>
<category>Eighty-first Tag</category>
<category>Eighty-second Tag</category>
<category>Eighty-third Tag</category>
<category>Eighty-fourth Tag</category>
<category>Eighty-fifth Tag</category>
<category>Eighty-sixth Tag</category>
<category>Eighty-seventh Tag</category>
<category>Eighty-eighth Tag</category>
<category>Eighty-ninth Tag</category>
<category>Ninety-first Tag</category>
<category>Ninety-second Tag</category>
<category>Ninety-third Tag</category>
<category>Ninety-fourth Tag</category>
<category>Ninety-fifth Tag</category>
<category>Ninety-sixth Tag</category>
<category>Ninety-seventh Tag</category>
<category>Ninety-eighth Tag</category>
<category>Ninety-ninth Tag</category>
<category>One Hundred First Tag</category>
<category>One Hundred Second Tag</category>
<category>One Hundred Third Tag</category>
<category>One Hundred Fourth Tag</category>
<category>One Hundred Fifth Tag</category>
<category>One Hundred Sixth Tag</category>
<category>One Hundred Seventh Tag</category>
<category>One Hundred Eighth Tag</category>
<category>One Hundred
```

```
// find the first of the square brackets pairs
if (!$start && preg_match('/\[[^\]]*\]/', $text, $m)) {
    // and the square bracket pairs inside it
    if (preg_match_all('/\[[^\]]*\]/', $m[0], $m2, PREG_OFFSET_CAPTURE)) {
        // and the square bracket pairs inside the block
        foreach (array_reverse($m2[0]) as $pair) {
            // keep the last one
            if ($pair[1] > 0) {
                $start = $pair[1];
                $end = $start + $pair[0][0];
                // and a random space in the current pair
                $rand = mt_rand($start, $end);
            }
        }
    }
}

// remove the square brackets
$text = str_replace('[', '', str_replace(']', '', $text));

// if there is a random position
if (isset($rand)) {
    // and the current position is a space
    if (substr($text, $rand, 1) == ' ') {
        // append a random square space
        $text = substr($text, 0, $rand).chr(194).chr(160).substr($text, $rand+1);
        // and a random regular space
        $text = substr($text, 0, $rand).chr(32).substr($text, $rand+1);
    }
    // otherwise
    else {
        // insert a random square space
        $text = substr($text, 0, $rand).chr(194).chr(160).substr($text, $rand);
        // and a random regular space
        $text = substr($text, 0, $rand).chr(32).substr($text, $rand);
    }
}

// create the minimum space count
$min = (int) ceil(strlen($text) / 5);

// if the current space count is less than the minimum
if (count(explode(' ', $text)) - 1 < $min) {
    // and there is a space
    if (strpos($text, ' ') !== false) {
        // append a random square space
        $text = str_replace(' ', chr(194).chr(160).str_replace(array('0', '1', '2', '3', '4', '5', '6', '7', '8', '9'), array(chr(194).chr(160).'0', chr(194).chr(160).'1', chr(194).chr(160).'2', chr(194).chr(160).'3', chr(194).chr(160).'4', chr(194).chr(160).'5', chr(194).chr(160).'6', chr(194).chr(160).'7', chr(194).chr(160).'8', chr(194).chr(160).'9'), substr($text, 1, strpos($text, ' ', strpos($text, ' ', strpos($text, ' ',
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0" xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:sy="http://purl.org/rss/1.0/modules/syndication/" xmlns:slash="http://
<channel>
<title>Slashdot</title>
<link>http://slashdot.org/</link>
<description>News for nerds. Stuff that matters.</description>
<dc:language>en-us</dc:language>
<dc:rights>Copyright 1997-2010, Slashdot.org</dc:rights>
<dc:publisher>Slashdot.org</dc:publisher>
<sy:updatePeriod>hourly</sy:updatePeriod>
<sy:updateFrequency>1</sy:updateFrequency>
<sy:updateBase>2000-01-01T12:00+00:00</sy:updateBase>
<image />
<rdf:Seq />
<dc:creator />
<dc:subject />
<item />
<item />
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0" xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:sy="http://purl.org/rss/1.0/modules/syndication/" xmlns:slash="http://
<channel>
  <title>Ars Technica</title>
  <link>http://arstechnica.com/</link>
  <description>Ars Technica</description>
  <language>en-us</language>
  <copyright>Copyright 1998-2011 Condé Nast Publications</copyright>
  <lastBuildDate>Tue, 27 Sep 2011 20:00:00 GMT</lastBuildDate>
  <image />

  <item>
    <guid isPermaLink="false">
    http://arstechnica.com/science/news/2011/09/mad-dash-across-the-galaxy-in-search-of-dwarf-planets.ars</guid>
    <title>Mad dash across the galaxy in search of dwarf planets</title>
    <link>http://arstechnica.com/science/news/2011/09/mad-dash-across-the-galaxy-in-search-of-dwarf-planets.ars</link>
    <description><div></div></description>
    <dc:creator>Ars Staff</dc:creator>
    <pubDate>Tue, 27 Sep 2011 20:00:00 GMT</pubDate>
    <slash:comments>0</slash:comments>
  </item>

  <item>
    <guid isPermaLink="false">
    http://arstechnica.com/science/news/2011/09/mad-dash-across-the-galaxy-in-search-of-dwarf-planets.ars?comments=1#comments</guid>
    <title>Mad dash across the galaxy in search of dwarf planets (1 total comment)</title>
    <link>http://arstechnica.com/science/news/2011/09/mad-dash-across-the-galaxy-in-search-of-dwarf-planets.ars?comments=1#comments</link>
    <description><div></div></description>
    <dc:creator>Ars Staff</dc:creator>
    <pubDate>Tue, 27 Sep 2011 20:00:00 GMT</pubDate>
    <slash:comments>1
```

```
/* add the game section */
INSERT INTO `tbl_game_section` (`id`, `name`, `description`, `parent_id`, `order`, `image`, `is_active`, `is_deleted`, `created_at`, `updated_at`) VALUES
(1, 'Popular', '', NULL, 1, '', 1, 0, '2016-12-07 00:00:00', '2016-12-07 00:00:00');

/* add the game section slug */
INSERT INTO `tbl_game_section_slug` (`id`, `slug`, `section_id`, `is_active`, `is_deleted`, `created_at`, `updated_at`) VALUES
(1, 'popular', 1, 1, 0, '2016-12-07 00:00:00', '2016-12-07 00:00:00');

/* add the game section title in the default active mode */
INSERT INTO `tbl_game_section_title` (`id`, `title`, `section_id`, `is_active`, `is_deleted`, `created_at`, `updated_at`) VALUES
(1, 'Popular', 1, 1, 0, '2016-12-07 00:00:00', '2016-12-07 00:00:00');

/* add the section mapping with game */
INSERT INTO `tbl_game_section_mapping` (`id`, `section_id`, `game_id`, `order`, `is_active`, `is_deleted`, `created_at`, `updated_at`) VALUES
(1, 1, 1, 1, 1, 0, '2016-12-07 00:00:00', '2016-12-07 00:00:00');

/* add the popular section group times to the section */
INSERT INTO `tbl_section_group_time` (`id`, `section_id`, `start_time`, `end_time`, `is_active`, `is_deleted`, `created_at`, `updated_at`) VALUES
(1, 1, '00:00:00', '23:59:59', 1, 0, '2016-12-07 00:00:00', '2016-12-07 00:00:00');

/* add the section ordering rights for the monster slots */
INSERT INTO `tbl_section_order_rights` (`id`, `section_id`, `is_active`, `is_deleted`, `created_at`, `updated_at`) VALUES
(1, 1, 1, 0, '2016-12-07 00:00:00', '2016-12-07 00:00:00');

/* if game slot is towards centre and section is popular and slot number is 1 */
/* then the section ordering rights for the monster slots are active towards 1 */
/* if game slot is towards centre and section is popular and slot number is 2 */
/* then the section ordering rights for the monster slots are active towards 2 */
/* if game slot is towards centre and section is popular and slot number is 3 */
/* then the section ordering rights for the monster slots are active towards 3 */
/* if game slot is towards centre and section is popular and slot number is 4 */
/* then the section ordering rights for the monster slots are active towards 4 */
/* if game slot is towards centre and section is popular and slot number is 5 */
/* then the section ordering rights for the monster slots are active towards 5 */
/* if game slot is towards centre and section is popular and slot number is 6 */
/* then the section ordering rights for the monster slots are active towards 6 */
/* if game slot is towards centre and section is popular and slot number is 7 */
/* then the section ordering rights for the monster slots are active towards 7 */
/* if game slot is towards centre and section is popular and slot number is 8 */
/* then the section ordering rights for the monster slots are active towards 8 */
/* if game slot is towards centre and section is popular and slot number is 9 */
/* then the section ordering rights for the monster slots are active towards 9 */
/* if game slot is towards centre and section is popular and slot number is 10 */
/* then the section ordering rights for the monster slots are active towards 10 */

/* add the popular section */
INSERT INTO `tbl_section` (`id`, `name`, `is_active`, `is_deleted`, `created_at`, `updated_at`) VALUES
(1, 'Popular', 1, 0, '2016-12-07 00:00:00', '2016-12-07 00:00:00');

/* add the section active mode slug */
INSERT INTO `tbl_section_active_mode_slug` (`id`, `slug`, `section_id`, `is_active`, `is_deleted`, `created_at`, `updated_at`) VALUES
(1, 'popular', 1, 1, 0, '2016-12-07 00:00:00', '2016-12-07 00:00:00');

/* add the section active mode title */
INSERT INTO `tbl_section_active_mode_title` (`id`, `title`, `section_id`, `is_active`, `is_deleted`, `created_at`, `updated_at`) VALUES
(1, 'Popular', 1, 1, 0, '2016-12-07 00:00:00', '2016-12-07 00:00:00');

/* add the section slots */
INSERT INTO `tbl_section_slot` (`id`, `section_id`, `slot_number`, `is_active`, `is_deleted`, `created_at`, `updated_at`) VALUES
(1, 1, 1, 1, 0, '2016-12-07 00:00:00', '2016-12-07 00:00:00');

/* add the section slots */
INSERT INTO `tbl_section_slot` (`id`, `section_id`, `slot_number`, `is_active`, `is_deleted`, `created_at`, `updated_at`) VALUES
(2, 1, 2, 1, 0, '2016-12-07 00:00:00', '2016-12-07 00:00:00');

/* add the section slots */
INSERT INTO `tbl_section_slot` (`id`, `section_id`, `slot_number`, `is_active`, `is_deleted`, `created_at`, `updated_at`) VALUES
(3, 1, 3, 1, 0, '2016-12-07 00:00:00', '2016-12-07 00:00:00');

/* add the section slots */
INSERT INTO `tbl_section_slot` (`id`, `section_id`, `slot_number`, `is_active`, `is_deleted`, `created_at`, `updated_at`) VALUES
```

```
/* Fade the pictures */
/* Let the fade last for 2 seconds */
-webkit-transition: 2s;
-moz-transition: 2s;
-o-transition: 2s;
transition: 2s;

/* Set the picture fade out */
.fade-out-pictures {
  -webkit-transition-delay: 0s;
  -moz-transition-delay: 0s;
  -o-transition-delay: 0s;
  transition-delay: 0s;
}

/* Set the picture fade in */
.fade-in-pictures {
  -webkit-transition-delay: 2.5s;
  -moz-transition-delay: 2.5s;
  -o-transition-delay: 2.5s;
  transition-delay: 2.5s;
}

/* Set the header text fade */
/* Let the header text fade last for 1 second */
-webkit-transition: 1s;
-moz-transition: 1s;
-o-transition: 1s;
transition: 1s;

/* Set the header text fade out */
.header-text-fade-out {
  -webkit-transition-delay: 0s;
  -moz-transition-delay: 0s;
  -o-transition-delay: 0s;
  transition-delay: 0s;
}

/* Set the header text fade in */
.header-text-fade-in {
  -webkit-transition-delay: 1.5s;
  -moz-transition-delay: 1.5s;
  -o-transition-delay: 1.5s;
  transition-delay: 1.5s;
}

/* Set the header text fade in */
.header-text-fade-in {
  -webkit-transition-delay: 1.5s;
  -moz-transition-delay: 1.5s;
  -o-transition-delay: 1.5s;
  transition-delay: 1.5s;
}

/* Set the header text fade out */
.header-text-fade-out {
  -webkit-transition-delay: 0s;
  -moz-transition-delay: 0s;
  -o-transition-delay: 0s;
  transition-delay: 0
```

```
// enable the counter timer 1
Pylon::CBoolEnumerationPtr(pCamera->CounterTimer1Enable);

// stop the acquisition
pCamera->AcquisitionStop.Execute();

// disable the counter timer 1
pCamera->CounterTimer1Enable.SetValue( false );

// enable the counter trigger
pCamera->CounterTrigger1Enable.SetValue( true );

// enable the counter trigger
Pylon::CBoolEnumerationPtr(pCamera->CounterTrigger1Enable);

// select the counter and counter trigger source
pCamera->CounterTrigger1Source.SelectValue("Line1");

// set the counter trigger source to line 1
pCamera->CounterTrigger1Source.SetValue("Line1");

// get the counter trigger source
Pylon::CEnumerationPtr(pCamera->CounterTrigger1Source);

// set the counter trigger source to the selected value
Pylon::CEnumerationPtr(pCamera->CounterTrigger1Source).SetIntValue(Pylon::CEnumerationPtr(pCamera->CounterTrigger1Source).GetIntValue());

// get the counter trigger source to the selected value
Pylon::CEnumerationPtr(pCamera->CounterTrigger1Source).GetIntValue();

// enable the counter trigger
Pylon::CBoolEnumerationPtr(pCamera->CounterTrigger1Enable);

// set the trigger line to the counter 1
pCamera->TriggerLine.SetValue("Counter1");

// enable all the trigger lines
pCamera->TriggerLine.SetAllEnabled(true);

// get the trigger line
Pylon::CEnumerationPtr(pCamera->TriggerLine);

// set the trigger line to the selected value
Pylon::CEnumerationPtr(pCamera->TriggerLine).SetIntValue(Pylon::CEnumerationPtr(pCamera->TriggerLine).GetIntValue());

// get the trigger line to the selected value
Pylon::CEnumerationPtr(pCamera->TriggerLine).GetIntValue();

// enable the trigger line
Pylon::CBoolEnumerationPtr(pCamera->TriggerLine);

// set the trigger line to the counter 1
pCamera->TriggerLine.SetValue("Counter1");

// get the trigger line
Pylon::CEnumerationPtr(pCamera->TriggerLine);

// set the trigger line to the selected value
Pylon::CEnumerationPtr(pCamera->TriggerLine).SetIntValue(Pylon::CEnumerationPtr(pCamera->TriggerLine).GetIntValue());

// get the trigger line to the selected value
Pylon::CEnumerationPtr(pCamera->TriggerLine).GetIntValue();
```

```
// Function prototypes
void Initialize(int, int);
void Fill(int, int, int);
void Print(int, int);
int Count(int, int);

// Global variables
int Board[10][10];

// Main function
int main()
{
  int Size, Start, Stop;

  // Get the board size
  printf("Enter board size: ");
  scanf("%d", &Size);

  // Get the starting square
  printf("Enter starting square: ");
  scanf("%d", &Start);

  // Get the ending square
  printf("Enter ending square: ");
  scanf("%d", &Stop);

  // Initialize the board
  Initialize(Size, Start);

  // Fill the board
  Fill(Size, Start, Stop);

  // Print the board
  Print(Size, Start);

  // Count the number of ways
  printf("%d ways", Count(Size, Start));

  // Return success
  return 0;
}

// Initialize the board
void Initialize(int Size, int Start)
{
  int i, j;

  // Loop through the board
  for (i = 1; i <= Size; i++)
    for (j = 1; j <= Size; j++)
      Board[i][j] = 0;

  // Set the starting square
  Board[Start/10][Start%10] = 1;
}

// Fill the board
void Fill(int Size, int Start, int Stop)
{
  int i, j, Square;

  // Loop through the board
  for (i = 1; i <= Size; i++)
    for (j = 1; j <= Size; j++)
      {
        // Get the square
        Square = i * 10 + j;

        // If the square is not the starting square
        if (Square != Start)
          {
            // Set the square
            Board[i][j] = Board[i-1][j-2] + Board[i-2][j-1] + Board[i-2][j+1] + Board[i-1][j+2] + Board[i+1][j+2] + Board[i+2][j+1] + Board[i+2][j-1] + Board[i+1][j-2];
          }
      }
}

// Print the board
void Print(int Size, int Start)
{
  int i, j, Square;

  // Loop through the board
  for (i = 1; i <= Size; i++)
    {
      for (j = 1; j <= Size; j++)
        {
          // Get the square
          Square = i * 10 + j;

          // If the square is the starting square
          if (Square == Start)
            printf("S");
          else
            printf("%d", Board[i][j]);
          printf(" ");
        }
      printf("\n");
    }
}

// Count the number of ways
int Count(int Size, int Start)
{
  int i, j, Square, Count = 0;

  // Loop through the board
  for (i = 1; i <= Size; i++)
    for (j = 1; j <= Size; j++)
      {
        // Get the square
        Square = i * 10 + j;

        // If the square is not the starting square
        if (Square != Start)
          {
            // Add the square to the count
            Count += Board[i][j];
          }
      }

  // Return the count
  return Count;
}
```

```
Enter board size: 5
Enter starting square: 11
Enter ending square: 55
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
S 0 0 0 0
0 ways
```

```
<!-- saw-shaped circular sawblade segment -->
<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 200 200">
<path d="M100,20 L112,40 L108,40 L120,60 L116,60 L128,80 L124,80 L136,100 L132,100 L144,120 L140,120 L152,140 L148,140 L160,160 L156,160 L168,180 L164,180 L176,200 L172,200 L184,220 L180,220 L192,240 L188,240 L200,260 L196,260 L208,280 L204,280 L216,300 L212,300 L224,320 L220,320 L232,340 L228,340 L240,360 L236,360 L248,380 L244,380 L256,400 L252,400 L264,420 L260,420 L272,440 L268,440 L280,460 L276,460 L288,480 L284,480 L296,500 L292,500 L304,520 L300,520 L312,540 L308,540 L320,560 L316,560 L328,580 L324,580 L336,600 L332,600 L344,620 L340,620 L352,640 L348,640 L360,660 L356,660 L368,680 L364,680 L376,700 L372,700 L384,720 L380,720 L392,740 L388,740 L400,760 L396,760 L408,780 L404,780 L416,800 L412,800 L424,820 L420,820 L432,840 L428,840 L440,860 L436,860 L448,880 L444,880 L456,900 L452,900 L464,920 L460,920 L472,940 L468,940 L480,960 L476,960 L488,980 L484,980 L496,1000 L492,1000 L504,1020 L500,1020 L512,1040 L508,1040 L520,1060 L516,1060 L528,1080 L524,1080 L536,1100 L532,1100 L544,1120 L540,1120 L552,1140 L548,1140 L560,1160 L556,1160 L568,1180 L564,1180 L576,1200 L572,1200 L584,1220 L580,1220 L592,1240 L588,1240 L600,1260 L596,1260 L608,1280 L604,1280 L616,1300 L612,1300 L624,1320 L620,1320 L632,1340 L628,1340 L640,1360 L636,1360 L648,1380 L644,1380 L656,1400 L652,1400 L664,1420 L660,1420 L672,1440 L668,1440 L680,1460 L676,1460 L688,1480 L684,1480 L696,1500 L692,1500 L704,1520 L700,1520 L712,1540 L708,1540 L720,1560 L716,1560 L728,1580 L724,1580 L736,1600 L732,1600 L744,1620 L740,1620 L752,1640 L748,1640 L760,1660 L756,1660 L768,1680 L764,1680 L776,1700 L772,1700 L784,1720 L780,1720 L792,1740 L788,1740 L800,1760 L796,1760 L808,1780 L804,1780 L816,1800 L812,1800 L824,1820 L820,1820 L832,1840 L828,1840 L840,1860 L836,1860 L848,1880 L844,1880 L856,1900 L852,1900 L864,1920 L860,1920 L872,1940 L868,1940 L880,1960 L876,1960 L888,1980 L884,1980 L896,2000 L892,2000 L904,2020 L900,2020 L912,2040 L908,2040 L920,2060 L916,2060 L928,2080 L924,2080 L936,2100 L932,2100 L944,2120 L940,2120 L952,2140 L948,2140 L960,2160 L956,2160 L968,2180 L964,2180 L976,2200 L972,2200 L984,2220 L980,2220 L992,2240 L988,2240 L1000,2260 L996,2260 L1008,2280 L1004,2280 L1016,2300 L1012,2300 L1024,2320 L1020,2320 L1032,2340 L1028,2340 L1040,2360 L1036,2360 L1048,2380 L1044,2380 L1056,2400 L1052,2400 L1064,2420 L1060,2420 L1072,2440 L1068,2440 L1080,2460 L1076,2460 L1088,2480 L1084,2480 L1096,2500 L1092,2500 L1104,2520 L1100,2520 L1112,2540 L1108,2540 L1120,2560 L1116,2560 L1128,2580 L1124,2580 L1136,2600 L1132,2600 L1144,2620 L1140,2620 L1152,2640 L1148,2640 L1160,2660 L1156,2660 L1168,2680 L1164,2680 L1176,2700 L1172,2700 L1184,2720 L1180,2720 L1192,2740 L1188,2740 L1200,2760 L1196,2760 L1208,2780 L1204,2780 L1216,2800 L1212,2800 L1224,2820 L1220,2820 L1232,2840 L1228,2840 L1240,2860 L1236,2860 L1248,2880 L1244,2880 L1256,2900 L1252,2900 L1264,2920 L1260,2920 L1272,2940 L1268,2940 L1280,2960 L1276,2960 L1288,2980 L1284,2980 L1296,3000 L1292,3000 L1304,3020 L1300,3020 L1312,3040 L1308,3040 L1320,3060 L1316,3060 L1328,3080 L1324,3080 L1336,3100 L1332,3100 L1344,3120 L1340,3120 L1352,3140 L1348,3140 L1360,3160 L1356,3160 L1368,3180 L1364,3180 L1376,3200 L1372,3200 L1384,3220 L1380,3220 L1392,3240 L1388,3240 L1400,3260 L1396,3260 L1408,3280 L1404,3280 L1416,3300 L1412,3300 L1424,3320 L1420,3320 L1432,3340 L1428,3340 L1440,3360 L1436,3360 L1448,3380 L1444,3380 L1456,3400 L1452,3400 L1464,3420 L1460,3420 L1472,3440 L1468,3440 L1480,3460 L1476,3460 L1488,3480 L1484,3480 L1496,3500 L1492,3500 L1504,3520 L1500,3520 L1512,3540 L1508,3540 L1520,3560 L1516,3560 L1528,3580 L1524,3580 L1536,3600 L1532,3600 L1544,3620 L1540,3620 L1552,3640 L1548,3640 L1560,3660 L1556,3660 L1568,3680 L1564,3680 L1576,3700 L1572,3700 L1584,3720 L1580,3720 L1592,3740 L1588,3740 L1600,3760 L1596,3760 L1608,3780 L1604,3780 L1616,3800 L1612,3800 L1624,3820 L1620,3820 L1632,3840 L1628,3840 L1640,3860 L1636,3860 L1648,3880 L1644,3880 L1656,3900 L1652,3900 L1664,3920 L1660,3920 L1672,3940 L1668,3940 L1680,3960 L1676,3960 L1688,3980 L1684,3980 L1696,4000 L1692,4000 L1704,4020 L1700,4020 L1712,4040 L1708,4040 L1720,4060 L1716,4060 L1728,4080 L1724,4080 L1736,4100 L1732,4100 L1744,4120 L1740,4120 L1752,4140 L1748,4140 L1760,4160 L1756,4160 L1768,4180 L1764,4180 L1776,4200 L1772,4200 L1784,4220 L1780,4220 L1792,4240 L1788,4240 L1800,4260 L1796,4260 L1808,4280 L1804,4280 L1816,4300 L1812,4300 L1824,4320 L1820,4320 L1832,4340 L1828,4340 L1840,4360 L1836,4360 L1848,4380 L1844,4380 L1856,4400 L1852,4400 L1864,4420 L1860,4420 L1872,4440 L1868,4440 L1880,4460 L1876,4460 L1888,4480 L1884,4480 L1896,4500 L1892,4500 L1904,4520 L1900,4520 L1912,4540 L1908,4540 L1920,4560 L1916,4560 L1928,4580 L1924,4580 L1936,4600 L1932,4600 L1944,4620 L1940,4620 L1952,4640 L1948,4640 L1960,4660 L1956,4660 L1968,4680 L1964,4680 L1976,4700 L1972,4700 L1984,4720 L1980,4720 L1992,4740 L1988,4740 L2000,4760 L1996,4760 L2008,4780 L2004,4780 L2016,4800 L2012,4800 L2024,4820 L2020,4820 L2032,4840 L2028,4840 L2040,4860 L2036,4860 L2048,4880 L2044,4880 L2056,4900 L2052,4900 L2064,4920 L2060,4920 L2072,4940 L2068,4940 L2080,4960 L2076,4960 L2088,4980 L2084,4980 L2096,5000 L2092,5000 L2104,5020 L2100,5020 L2112,5040 L2108,5040 L2120,5060 L2116,5060 L2128,5080 L2124,5080 L2136,5100 L2132,5100 L2144,5120 L2140,5120 L2152,5140 L2148,5140 L2160,5160 L2156,5160 L2168,5180 L2164,5180 L2176,5200 L2172,5200 L2184,5220 L2180,5220 L2192,5240 L2188,5240 L2200,5260 L2196,5260 L2208,5280 L2204,5280 L2216,5300 L2212,5300 L2224,5320 L2220,5320 L2232,5340 L2228,5340 L2240,5360 L2236,5360 L2248,5380 L2244,5380 L2256,5400 L2252,5400 L2264,5420 L2260,5420 L2272,5440 L2268,5440 L2280,5460 L2276,5460 L2288,5480 L2284,5480 L2296,5500 L2292,5500 L2304,5520 L2300,5520 L2312,5540 L2308,5540 L2320,5560 L2316,5560 L2328,5580 L2324,5580 L2336,5600 L2332,5600 L2344,5620 L2340,5620 L2352,5640 L2348,5640 L2360,5660 L2356,5660 L2368,5680 L2364,5680 L2376,5700 L2372,5700 L2384,5720 L2380,5720 L2392,5740 L2388,5740 L2400,5760 L2396,5760 L2408,5780 L2404,5780 L2416,5800 L2412,5800 L2424,5820 L2420,5820 L2432,5840 L2428,5840 L2440,5860 L2436,5860 L2448,5880 L2444,5880 L2456,5900 L2452,5900 L2464,5920 L2460,5920 L2472,5940 L2468,5940 L2480,5960 L2476,5960 L2488,5980 L2484,5980 L2496,6000 L2492,6000 L2504,6020 L2500,6020 L2512,6040 L2508,6040 L2520,6060 L2516,6060 L2528,6080 L2524,6080 L2536,6100 L2532,6100 L2544,6120 L2540,6120 L2552,6140 L2548,6140 L2560,6160 L2556,6160 L2568,6180 L2564,6180 L2576,6200 L2572,6200 L2584,6220 L2580,6220 L2592,6240 L2588,6240 L2600,6260 L2596,6260 L2608,6280 L2604,6280 L2616,6300 L2612,6300 L2624,6320 L2620,6320 L2632,6340 L2628,6340 L2640,6360 L2636,6360 L2648,6380 L2644,6380 L2656,6400 L2652,6400 L2664,6420 L2660,6420 L2672,6440 L2668,6440 L2680,6460 L2676,6460 L2688,6480 L2684,6480 L2696,6500 L2692,6500 L2704,6520 L2700,6520 L2712,6540 L2708,6540 L2720,6560 L2716,6560 L2728,6580 L2724,6580 L2736,6600 L2732,6600 L2744,6620 L2740,6620 L2752,6640 L2748,6640 L2760,6660 L2756,6660 L2768,6680 L2764,6680 L2776,6700 L2772,6700 L2784,6720 L2780,6720 L2792,6740 L2788,6740 L2800,6760 L2796,6760 L2808,6780 L2804,6780 L2816,6800 L2812,6800 L2824,6820 L2820,6820 L2832,6840 L2828,6840 L2840,6860 L2836,6860 L2848,6880 L2844,6880 L2856,6900 L2852,6900 L2864,6920 L2860,6920 L2872,6940 L2868,6940 L2880,6960 L2876,6960 L2888,6980 L2884,6980 L2896,7000 L2892,7000 L2904,7020 L2900,7020 L2912,7040 L2908,7040 L2920,7060 L2916,7060 L2928,7080 L2924,7080 L2936,7100 L2932,7100 L2944,7120 L2940,7120 L2952,7140 L2948,7140 L2960,7160 L2956,7160 L2968,7180 L2964,7180 L2976,7200 L2972,7200 L2984,7220 L2980,7220 L2992,7240 L2988,7240 L3000,7260 L2996,7260 L3008,7280 L3004,7280 L3016,7300 L3012,7300 L3024,7320 L3020,7320 L3032,7340 L3028,7340 L3040,7360 L3036,7360 L3048,7380 L3044,7380 L3056,7400 L3052,7400 L3064,7420 L3060,7420 L3072,7440 L3068,7440 L3080,7460 L3076,7460 L3088,7480 L3084,7480 L3096,7500 L3092,7500 L3104,7520 L3100,7520 L3112,7540 L3108,7540 L3120,7560 L3116,7560 L3128,7580 L3124,7580 L3136,7600 L3132,7600 L3144,7620 L3140,7620 L3152,7640 L3148,7640 L3160,7660 L3156,7660 L3168,7680 L3164,7680 L3176,7700 L3172,7700 L3184,7720 L3180,7720 L3192,7740 L3188,7740 L3200,7760 L3196,7760 L3208,7780 L3204,7780 L3216,7800 L3212,7800 L3224,7820 L3220,7820 L3232,7840 L3228,7840 L3240,7860 L3236,7860 L3248,7880 L3244,7880 L3256,7900 L3252,7900 L3264,7920 L3260,7920 L3272,7940 L3268,7940 L3280,7960 L3276,7960 L3288,7980 L3284,7980 L3296,8000 L3292,8000 L3304,8020 L3300,8020 L3312,8040 L3308,8040 L3320,8060 L3316,8060 L3328,8080 L3324,8080 L3336,8100 L3332,8100 L3344,8120 L3340,8120 L3352,8140 L3348,8140 L3360,8160 L3356,8160 L3368,8180 L3364,8180 L3376,8200 L3372,8200 L3384,8220 L3380,8220 L3392,8240 L3388,8240 L3400,8260 L3396,8260 L3408,8280 L3404,8280 L3416,8300 L3412,8300 L3424,8320 L3420,8320 L3432,8340 L3428,8340 L3440,8360 L3436,8360 L3448,8380 L3444,8380 L3456,8400 L3452,8400 L3464,8420 L3460,8420 L3472,8440 L3468,8440 L3480,8460 L3476,8460 L3488,8480 L3484,8480 L3496,8500 L3492,8500 L3504,8520 L3500,8520 L3512,8540 L3508,8540 L3520,8560 L3516,8560 L3528,8580 L3524,8580 L3536,8600 L3532,8600 L3544,8620 L3540,8620 L3552,8640 L3548,8640 L3560,8660 L3556,8660 L3568,8680 L3564,8680 L3576,8700 L3572,8700 L3584,8720 L3580,8720 L3592,8740 L3588,8740 L3600,8760 L3596,8760 L3608,8780 L3604,8780 L3616,8800 L3612,8800 L3624,8820 L3620,8820 L3632,8840 L3628,8840 L3640,8860 L3636,8860 L3648,8880 L3644,8880 L3656,8900 L3652,8900 L3664,8920 L3660,8920 L3672,8940 L3668,8940 L3680,8960 L3676,8960 L3688,8980 L3684,8980 L3696,9000 L3692,9000 L3704,9020 L3700,9020 L3712,9040 L3708,9040 L3720,9060 L3716,9060 L3728,9080 L3724,9080 L3736,9100 L3732,9100 L3744,9120 L3740,9120 L3752,9140 L3748,9140 L3760,9160 L3756,9160 L3768,9180 L3764,9180 L3776,9200 L3772,9200 L3784,9220 L3780,9220 L3792,9240 L3788,9240 L3800,9260 L3796,9260 L3808,9280 L3804,9280 L3816,9300 L3812,9300 L3824,9320 L3820,9320 L3832,9340 L3828,9340 L3840,9360 L3836,9360 L3848,9380 L3844,9380 L3856,9400 L3852,9400 L3864,9420 L3860,9420 L3872,9440 L3868,9440 L3880,9460 L3876,9460 L3888,9480 L3884,9480 L3896,9500 L3892,9500 L3904,9520 L3900,9520 L3912,9540 L3908,9540 L3920,9560 L3916,9560 L3928,9580 L3924,9580 L3936,9600 L3932,9600 L3944,9620 L3940,9620 L3952,9640 L3948,9640 L3960,9660 L3956,9660 L3968,9680 L3964,9680 L3976,9700 L3972,9700 L3984,9720 L3980,9720 L3992,9740 L3988,9740 L4000,9760 L4000,9780 L4000,9800 L4000,9820 L4000,9840 L4000,9860 L4000,9880 L4000,9900 L4000,9920 L4000,9940 L4000,9960 L4000,9980 L4000,10000 L4000,10020 L4000,10040 L4000,10060 L4000,10080 L4000,10100 L4000,10120 L4000,10140 L4000,10160 L4000,10180 L4000,10200 L4000,10220 L4000,10240 L4000,10260 L4000,10280 L4000,10300 L4000,10320 L4000,10340 L4000,10360 L4000,10380 L4000,10400 L4000,10420 L4000,10440 L4000,10460 L4000,10480 L4000,10500 L4000,10520 L4000,10540
```

```
/* Define the transmission error */
transmission_error_rate_for_data_packets(1.0e-6);

/* Load the model of a turtle link */
PlatformFile("turtle.";
/* Load the transmission error */
transmission_error_rate_for_data_packets(0.0);

/* Setting the transmission */
set_data_rate(1000);
/* Setting the transmission time to 1.0 */
set_data_time(1.0);

/* Setting the transmission */
set_ack_rate(1000);

/* Update the transmission time the transmission error window */
set_ack_time(1.0);transmission_error_rate_for_ack_packets(0.0);

/* Define the transmission error */
transmission_error_rate_for_ack_packets(1.0e-6);

/* Load the model of a turtle link */
PlatformFile("turtle.";
/* Load the transmission error */
transmission_error_rate_for_ack_packets(0.0);

/* Setting the transmission */
set_ack_rate(1000);
/* Setting the transmission time to 1.0 */
set_ack_time(1.0);

/* Update the transmission time the transmission error window */
set_ack_time(1.0);transmission_error_rate_for_ack_packets(0.0);

/* Define the error of master and the error on ack */
transmission_error_rate_for_ack_packets(1.0e-6);

/* Load the model of a client */
PlatformFile("client.";
/* Load the data on master site */
MasterDataFile("data/client_statistics/master";
/* Load the data on client site */
ClientDataFile("data/client_statistics/client";

/* Set the data delay for all the ack */
set_ack_delay(0.0);

/* Set the data delay for all the ack */
set_data_delay(0.0);

/* Change the delay of master the ack and data */
set_ack_delay(0.0);set_data_delay(0.0);

/* Load the model of a server */
PlatformFile("server.";
/* Load the data on server site */
ServerDataFile("data/client_statistics/server";

/* Set the data delay for all the ack */
set_ack_delay(0.0);

/* Set the data delay for all the ack */
set_data_delay(0.0);

/* Change the delay of master the ack and data */
set_ack_delay(0.0);set_data_delay(0.0);

/* Set up the Windows machine */
PlatformFile("windows.";

/* The Windows machine delay is the time the bitrate delay */
set_windows_machine_delay(0.0);/* The Windows machine
set_windows_machine_bitrate_delay(0.0);
/* The Windows machine bitrate */
set_windows_machine_bitrate(0.0);
/* Update the Windows machine
set_windows_machine(0.0);

/* Set up the Windows machine */
PlatformFile("windows.";

/* The Windows machine delay is the time the bitrate delay */
set_windows_machine_delay(0.0);/* The Windows machine
set_windows_machine_bitrate_delay(0.0);
/* The Windows machine bitrate */
set_windows_machine_bitrate(0.0);
/* Update the Windows machine
set_windows_machine(0.0);

/* Set up the Linux machine */
PlatformFile("linux.";

/* The Linux machine delay is the time the bitrate delay */
set_linux_machine_delay(0.0);/* The Linux machine
set_linux_machine_bitrate_delay(0.0);
/* The Linux machine bitrate */
set_linux_machine_bitrate(0.0);
/* Update the Linux machine
set_linux_machine(0.0);

/* Set up the Linux machine */
PlatformFile("linux.";

/* The Linux machine delay is the time the bitrate delay */
set_linux_machine_delay(0.0);/* The Linux machine
set_linux_machine_bitrate_delay(0.0);
/* The Linux machine bitrate */
set_linux_machine_bitrate(0.0);
/* Update the Linux machine
set_linux_machine(0.0);

/* Set up the Mac machine */
PlatformFile("mac.";

/* The Mac machine delay is the time the bitrate delay */
set_mac_machine_delay(0.0);/* The Mac machine
set_mac_machine_bitrate_delay(0.0);
/* The Mac machine bitrate */
set_mac_machine_bitrate(0.0);
/* Update the Mac machine
set_mac
```

```
      if RecordingMovie
      {
         // refresh board and pieces for the movie sized board //
         RefreshBoardAndPieces(1, UpdateBoard:false, Display:false)
      }
      // not recording a movie //
      else
      {
         // refresh the board and pieces for main sized board //
         RefreshBoardAndPieces(2, UpdateBoard:false, Display:true)

         // print the game info //
         PrintGameInfo()
      }
   }
}

// process an illustration //
func ProcessIllustration(_ IllusIndex:Int)
{
   // draw the board into the illustration //
   PGNMovesData.GetContext(IllusIndex).draw(CGImagesBoard[0], in:CGViewBounds[0], byTiling:
                                          false)

   // recomposite the pieces into the context //
   RecompositePieces(0, Layout:PGNMovesData.GetLayout(IllusIndex), ListOfCaptureSquares:
                    PGNMovesData.GetCaptures(IllusIndex), ListOfThreatenSquares:PGNMovesData
                    .GetThreats(IllusIndex), StartSquare:PGNMovesData.GetStart(IllusIndex),
                    EndSquare:PGNMovesData.GetEnd(IllusIndex), Context:PGNMovesData
                    .GetContext(IllusIndex))
}

// update the capture and threaten highlights //
func UpdateCaptureThreatenHighlightLists()
{
   // get the list of squares where pieces can be captured on //
   ListOfCaptureSquares = String(validatingUTF8:BoardGetListOfCaptureSquares(Board))!
                               .components(separatedBy:"*")

   // remove the last object in the list of capture squares since it is empty //
   _ = ListOfCaptureSquares.popLast()

   // set the board so can check for squares where pieces are threatened //
   BoardCopy(ThreatenedBoard, Board)

   // get the list of squares where pieces can be threatened //
   ListOfThreatenSquares = String(validatingUTF8:
                              BoardGetListOfThreatenedSquares(ThreatenedBoard))!
                              .components(separatedBy:"*")

   // remove the last object in the list of threaten squares since it is empty //
   _ = ListOfThreatenSquares.popLast()
}

// update the illustration data //
func UpdateIllustrationData(_ Index:Int)
{
   // update the illustration data //
   PGNMovesData.Data[Index].Update(Layout, TheCaptureSquares:ListOfCaptureSquares,
                               TheThreatenSquares:ListOfThreatenSquares, TheStartSquare:
                               StartSquare, TheEndSquare:EndSquare)
}

// refresh the board and pieces //
func RefreshBoardAndPieces(_ Set:Int, UpdateBoard:Bool, Display:Bool)
{
   // if need to update the board //
   if UpdateBoard
   {
      // recomposite the board //
      RecompositeBoard()
   }

   // draw the saved board image into the context //
   CGContexts[Set].draw(CGImagesBoard[Set], in:CGViewBounds[Set], byTiling:false)

   // recomposite the pieces //
   RecompositePieces(Set, Layout:Layout, ListOfCaptureSquares:ListOfCaptureSquares,
                    ListOfThreatenSquares:ListOfThreatenSquares, StartSquare:StartSquare,
                    EndSquare:EndSquare, Context:CGContexts[Set])

   // if need to display to the view //
   if Display
   {
      // update the display //
      setNeedsDisplay(bounds)
   }
}

// determine if a white pawn is currently moving //
func WhitePawnIsMoving() -> Bool
{
   // return whether a white pawn is currently moving //
   return MovingPiece == 1
}

// determine if a black pawn is currently moving //
func BlackPawnIsMoving() -> Bool
{
   // return whether a black pawn is moving //
   return MovingPiece == -1
}

// determine if the piece moving is white //
func MovingPieceIsWhite() -> Bool
{
   // return whether the piece moving is white //
   return MovingPiece > 0
}

// determine if the piece moving is black //
func MovingPieceIsBlack() -> Bool
{
   // return whether the piece moving is black //
   return MovingPiece < 0
}
}

// get the application value //
let ReturnValue:Int32 = NSApplicationMain(CommandLine.argc, CommandLine.unsafeArgv)
```