

В целом же, изменение эксцесса (у нас в 3 раза) некатастрофично для ANOVA и KruskalWallis. В этом случае можно рекомендовать их совместное применение. При согласии их выводов заключение можно считать достоверным.

Заключение

Итак, мы может сделать следующие выводы:

1) негомогенность дисперсий (даже сильная) не оказывает катастрофического влияния на критерий ANOVA. В нашем примере с геометрическим ростом СКО наблюдалось возрастание эффективной вероятности ошибки 1-го рода всего с 5% до 10%;

2) тесты множественных сравнений (ANOVA, Kruskal-Wallis) практически нечувствительны к увеличению эксцесса по сравнению с нормальным распределением при условии сонаправленности асимметрий;

3) наблюдается некоторое ухудшение ситуации в случае с распределениями, имеющими разнонаправленные асимметрии.

Представляется полезным итерационным методом найти уровни значимости, соответствующие статистически наблюдаемым ошибкам 1-го рода для используемых «ненормальных» распределений. Имеет смысл провести аналогичные исследования для других распределений с малыми объемами выборок. Также мы планируем аналогичным методом исследовать влияние «ненормальности» на мощность критериев.

Библиографический список

1. Гмурман В.Е. Теория вероятностей и математическая статистика – М., Высшая школа, 2003.
2. Эверитт, Б.С. Большой словарь по статистике / Б.С. Эверитт. - М.: Проспект, 2012. - 736 с.
3. Upton, GrahamCook, Ian. Dictionaryofstatistics. - OxfordUniversityPress, Оксфорд, Великобритания, 2006. – 464 с.
4. Лапач С.Н., Чубенко А.В., Бабич П.Н., Статистические методы в медико-биологических исследованиях с использованием Excel. – 2-е изд. перераб. и доп. – К.: МОРИОН, 2001. – 408 с.
5. С. Гланц. Медико-биологическая статистика. Пер. с англ. — М., Практика, 1998. — 459 с.
6. Кобзарь А. И. «Прикладная математическая статистика», для инженеров и научных работников, 2008. Размер 8,1 Мб, 816 с.

УДК 20.51.01

Ахтырский А.И.

РАЗВИТИЕ ИГРОВОГО ПРОЕКТА “SPACE STATION 13” НА ПЛАТФОРМЕ BYOND ЯЗЫКА DREAM MAKER DEVELOPMENT OF THE “SPACE STATION 13” GAME PROJECT ON THE BYOND PLATFORM OF THE DREAM MAKER LANGUAGE

*Филиал «Протвино» государственного университета «Дубна»
Секция «Информационные технологии»*

Автор: Ахтырский Андрей Игоревич, студент 1 курса, направления «Информатика и вычислительная техника» филиала «Протвино» государственного университета «Дубна».

Научный руководитель: Астафьева Марина Петровна, кандидат технических наук, доцент кафедры информационных технологий филиала «Протвино» государственного университета «Дубна».

Author: Akhtyrskiy Andrey Igorevich, 1st year student of the direction "Informatics and computer engineering" of the branch "Protvino" state University "Dubna".

Scientific adviser: Astafieva Marina Petrovna, candidate of technical sciences, associate professor of the department information technology of the branch "Protvino" state University "Dubna".

Аннотация

В данной статье описано принятие участия в разработке и поддержке развития серверов игры Space Station 13. Цель работы — показать приёмы разработки и поддержки игрового приложения с применением кода, расположенного в открытом доступе на Github.com.

Abstract

This article describes the participation in the development and support of the development of the servers of the Space Station 13 game. The purpose of the work is to show the methods of developing and supporting the game application using the code located in the public domain on Github.com.

Ключевые слова: BYOND, Dream Maker, Space Station 13, Github.com, спрайты

Keywords: Byond, Dream Maker, Space Station 13, Github.com, sprites

Компьютерные игры – важный аспект нашего общества. Они являются актуальными и по сей день, постоянно развиваясь и все время показывая что-то новое. Для людей такое увлечение играет разную роль: для кого-то это заработок в киберспорте или разработке, для кого-то – способ отдохнуть, получить наслаждение от красивой картинке или от игрового процесса. Также в настоящее время интенсивно развиваются открытые платформы и работа при помощи удаленного доступа с открытым кодом базы, что обеспечивает актуальность данной работы.

Задачи, которые необходимо решить для достижения цели работы: опрос и анализ требований пользователей, нахождение и изучение работы определенного кода программы в открытом доступе, создание графического изображения, написание программного кода, тестирование разработанного фрагмента игры.

BYOND (Build Your Own Net Dream) – платформа, о которой будет вестись речь в этой работе, используется для создания различных двумерных игр. Платформа работает под операционными системами Windows и Linux. Созданные игры на этой платформе разработаны исключительно для использования на компьютере. Данная платформа также использует для программирования свой собственный язык Dream Maker. Игровое сообщество является межнациональным. Люди разных стран играют на серверах, вместе разрабатывают новые идеи и игры.

Dream Maker основан на методе наследования – есть “родительские” объекты, которые передают свои переменные “дочерним” объектам. BYOND обладает базой объектов, которые легко позволяют создать новые объекты и передавать им различные значения. Эти объекты называются atom, и включает следующие компоненты: Area – определенная зона, размером в один тайл – мельчайшей единицей измерения пространства в BYOND; Turf – “покрытие” мира (пол, стены, возможности их редактирования); Obj – игровые объекты, которые можно двигать, заставить двигаться

или просто использовать по назначению; Mob – игровой объект, который может быть подконтролен персонажу. База кода на многих серверах открыта, что позволяет изучать её для саморазвития или её улучшения, и расположена, преимущественно, на Github.com.

Также для работы использовались: Aseprite – графический редактор 2D изображений; среда Visual Studio, сайт Github.com, SmartGit – Гит-клиент для работы с сайтом Github.com.

Space Station 13 – многопользовательская ролевая компьютерная игра о работе и жизни персонала на космической станции. Первая версия игры была разработана и выпущена в свет в 2003 году. В игре существуют различные игровые сервера, владельцы и разработчики которых разрабатывают игру по различным направлениям. Основная черта игрового процесса – отыгрывание определенных профессий на космической станции: инженер, детектив, член службы безопасности, врач, капитан и так далее. Игра является сессионной, т.е. каждый раунд длится определенное время, пока не произойдет какое-то критическое событие в игре или пока персонал не вызовет эвакуационный шаттл. Каждый раунд происходят различные события: нападение космических рейдеров, агентов синдиката, неизвестных инопланетных существ и многое другое. Игрок управляет определенным персонажем, который может передвигаться вверх, вниз, влево и вправо, на более современных серверах – ещё и по осе Z. (между этажами станции). Управление происходит с помощью клавиатуры, мышкой. Игрок может взаимодействовать при помощи мышки с различными игровыми объектами: автоматами по продаже напитков, компьютерами, шкафчиками с различным содержимым, газами для разных целей и т.д.

На Рисунке 1 можно увидеть экран игры. Правую половину экрана занимает панель с различными вкладками, отвечающими либо за кнопки быстрого взаимодействия с окружающим миром, либо за информацию о состоянии сервера, игровом мире и т.д. На левой части экрана изображена игровая область – на ней можно увидеть главного персонажа по центру и различные объекты, расположенные вокруг него. В нижней части экрана находится строка ввода, в которую можно вводить команды. К примеру: say Привет мир выведет на экран сообщение от лица персонажа Привет мир, которое будет видно другим персонажам в радиусе видимости говорящего. Над строкой ввода располагается HUD персонажа: инвентарь, состояние персонажа, его здоровье и т.д.



Рисунок 1 – Экран игры

На Рисунках 3-4 показаны примеры программного кода, который будет обрабатывать определенные поставленные задачи.

Как можно увидеть на Рисунке 2, первый объект – “стоковая” часть оружия, второй объект – дуло, третий объект – ресивер и оптический прицел, совмещенные в один объект с помощью кода. Четвертый объект – результат соединения стоковой части и ресивера, а финальным объектом будет являться уже готовое оружие. На Рисунке 4 изображена коробка, в которой будут храниться эти объекты и рядом с ней – её код.



Рисунок 2 – Фрагменты и готовое оружие

```

/obj/item/weapon/weaponparts
    var/part_color = ""
    name = "weaponpart"
    desc = "how did you get it?"
    icon = "icons/obj/weaponparts.dmi"

/obj/item/weapon/weaponparts/heavysniper/stock
    name = "sniper stock"
    desc = "This is a sniper stock. You need to attach the reciever."
    icon_state = "sniper_stock"

/obj/item/weapon/weaponparts/heavysniper/reciever
    name = "sniper reciever"
    desc = "This is a sniper reciever. You need to attach it to the stock."
    icon_state = "sniper_reciever"

/obj/item/weapon/weaponparts/heavysniper/stockreciever
    name = "sniper stock with reciever"
    desc = "This is a sniper stock with reciever. Now attach the barrel."
    icon_state = "sniper_stockreciever"

/obj/item/weapon/weaponparts/heavysniper/barrel
    name = "sniper rifle barrel"
    desc = "This is a barrel from a sniper rifle."
    icon_state = "sniper_barrel"

/obj/item/weapon/weaponparts/heavysniper/stock/attackby(obj/item/W, mob/user,)
    if(istype(W,/obj/item/weapon/weaponparts/heavysniper/reciever))
        to_chat(user, "You attach the reciever to the stock")
        var/obj/item/weapon/weaponparts/heavysniper/stockreciever/HS = new (get_turf(src))
        if(loc == user)
            equip_slot = user.get_inventory_slot(src)
            if(equip_slot in list(slot_r_hand, slot_l_hand))
                user.drop_from_inventory(src)
                user.equip_to_slot_if_possible(HS, equip_slot)

        qdel(W)
        qdel(src)

/obj/item/weapon/weaponparts/heavysniper/stockreciever/attackby(obj/item/W, mob/user)
    if(istype(W,/obj/item/weapon/weaponparts/heavysniper/barrel))
        to_chat(user, "You attach the barrel to the stock")
        var/obj/item/weapon/gun/projectile/heavysniper/HS = new (get_turf(src))
        if(loc == user)
            equip_slot = user.get_inventory_slot(src)
            if(equip_slot in list(slot_r_hand, slot_l_hand))
                user.drop_from_inventory(src)
                user.equip_to_slot_if_possible(HS, equip_slot)

        qdel(W)
        qdel(src)

```

Рисунок 3 – Программный код оружия



```

/obj/item/weapon/storage/box/syndie_kit/antimaterial_rifle
    name = "Sniper rifle box"
    desc = "Sniper rifle kit. One shot for real men."
    icon_state = "box_of_doom_big"
    w_class = ITEM_SIZE_HUGE

/obj/item/weapon/storage/box/syndie_kit/antimaterial_rifle/New()
    ..()
    new /obj/item/ammo_casing/a145(src)
    new /obj/item/weapon/weaponparts/heavysniper/disassembled(src)
    new /obj/item/weapon/weaponparts/heavysniper/stock(src)
    new /obj/item/weapon/weaponparts/heavysniper/barrel(src)

```

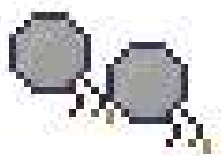
Рисунок 4 – Коробка для хранения объектов и её программный код

На Рисунке 5 и 6 два других объекта: подошвы и линзы. Оба объекта являются улучшениями для других объектов. Подошва делает обувь нескользящей, а линзы позволяют видеть в очках живые объекты сквозь стены.

```
/obj/item/clothing/glasses/powered/thermal/lens
name = "Thermal lenses"
desc = "Lenses for glasses."
toggleable = FALSE
icon_state = "thermal_lens"
body_parts_covered = 0
slot_flags = 0

/obj/item/clothing/glasses/attackby(var/obj/item/Z, var/mob/user)

if (istype(Z, /obj/item/clothing/glasses/powered/thermal/lens))
    overlay = global_hud.thermal
    vision_flags = SEE_MOBS
    see_invisible = SEE_INVISIBLE_NOLIGHTING
    protection = flash_protection
    flash_protection = FLASH_PROTECTION_REDUCED
    origin_tech = list(TECH_ILLEGAL = 3)
    to_chat(usr, "You attached your lenses to your glasses")
    have_lenses = 1
    qdel(Z)
```



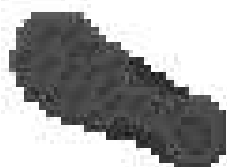
thermal_lens

```
/obj/item/clothing/glasses/powered/thermal/attackby(var/obj/item/X, var/mob/user)
if(istype(X, /obj/item/clothing/glasses/powered/thermal/lens))
    to_chat(usr, "This glasses already have thermal implant")

/obj/item/clothing/glasses/verb/detach_lenses()
set name = "Detach lenses"
set category = "Object"
set src in view(1)

if (have_lenses == 1)
    flash_protection = protection;
    see_invisible = -1;
    vision_flags = 0;
    origin_tech = 0;
    have_lenses = 0;
    overlay = 0;
    to_chat(usr, "You detach lenses from your glasses");
    var/obj/item/clothing/glasses/powered/thermal/lens/THL = new()
    usr.put_in_hands(THL)
else to_chat(usr, "You haven't got any lenses in your glasses");
```

Рисунок 5 – Линза и её программный код



no_slip_sole

```
//No-slip module for shoes
/obj/item/noslipmodule
name = "no slip sole"
desc = "Attach it to your shoe."
icon = 'icons/inventory/feet/icon.dmi'
icon_state = "no_slip_sole"
```



```

/obj/item/clothing/shoes/attachby(var/obj/item/I, var/mob/user)
var/global/knives
+   if(istype(I,/obj/item/noslipmodule))
+       if (item_flags != 0)
+           noslip = item_flags
+           module_inside = 1
+           to_chat(user, "You attached no slip sole")
+           permeability_coefficient = 0.05
+           item_flags = NOSLIP | SILENT
+           origin_tech = list(TECH_ILLEGAL = 3)
+           siemens_coefficient = 0 // DAMN BOI
+           qdel(I)
+
if(!knives)
    knives = list(
        /obj/item/weapon/material/knife,

/obj/item/clothing/shoes/verb/detach_noslipmodule()
set name = "Detach accessory"
set category = "Object"
set src in view(1)

if (module_inside == 1 )
    if (noslip != 0)
        item_flags = noslip
        var/obj/item/noslipmodule/NSM = new()
        usr.put_in_hands(NSM)
else to_chat(usr, "You haven't got any accessories in your shoes")

```

Рисунок 6 – Подошва и её программный код

Также BYOND поддерживает изменение игрового пространства с помощью интерфейса внутри редактора кода. (тоже описал что где и как отображено) Изучена работа с сайтом Github.com. Этот сервис является очень удобным не только для разработки игр, но и для совместной разработки приложений.

В заключение можно отметить, что цель работы достигнута – показаны приёмы разработки и поддержки игрового приложения с применением кода, расположенного в открытом доступе на Github.com, а также выполнены поставленные задачи:

1. Изучена структура кода Dream Maker и кода самой игры Space Station 13.
2. Изучен также редактор игрового пространства и возможности взаимодействия с ним.
3. Изучена работа с графическим 2D редактором и разработаны с его помощью изображения для объектов в игре.
4. Разработаны сами объекты для игры.
5. Проверена и протестирована работоспособность объектов

В данной работе были показаны лишь небольшие примеры разработки, на протяжении трёх лет было исследовано и создано больше новых объектов для различных игровых серверов.

Библиографический список

1. Сайт платформы Byond <http://www.byond.com/>.
2. Официальная вики сервера Chaotic Onyx https://wiki.ss13.ru/index.php?title=Main_Page.