

Open Source eUC

Que es código abierto,
un poco de mí y como
puedes colaborar



¿Qué es Open Source eUC?

Open Source eUC es una **comunidad** de estudiantes interesados por el **código abierto**, que colabore en **proyectos** de distinto tipo, tamaño y alcance

¿Qué es Open Source?

Open Source



- Modelo de desarrollo basado en **colaboración abierta**
- El software es **público** y queda bajo una **licencia**, que usualmente permite usar, modificar y distribuirlo

En resumen, código es público, cualquiera puede usarlo y modificarlo

opensource.org/osd

Software Libre



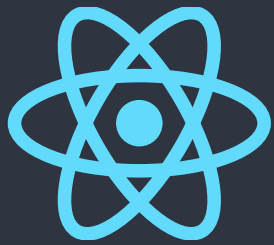
- Todo software libre es de código abierto
- **Movimiento** que busca libertad en el software
- Libertad de correr, estudiar, cambiar y compartir

www.gnu.org/philosophy/open-source-misses-the-point.es

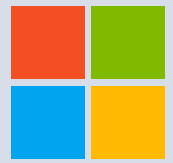
¿Quiénes usan código abierto?

Todo el mundo

Proyectos de código abierto



Empresas que contribuyen a OS



Microsoft

amazon

Google



reddit

NETFLIX



IBM



Meta



Uber

intel®

pero, ¿por qué?

- Software es indivisible: **copias no requieren costo**
- Devuelve **control** a usuarios
- Compartir y colaborar evita **reinventar la rueda**
- Sin fines de lucro, siempre es *net-positive*

devuelta a nosotros...

Beneficios de OS en la Universidad

- **Evita que proyectos se pierdan** en el pasado
- Colaborar ayuda a **mejorar** y **aprender**
- Hay **poco tiempo** para hacer proyectos grandes
- **Potenciar proyectos** que serían **difíciles** sin colaboración

¿Qué **proyectos** o
actividades tenemos?
(por ahora)

Algunos de nuestros proyectos

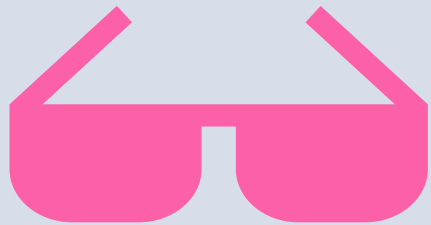


RamosUC

CANVAS  **STREAM**



UbiCate



**awesome
UC**

L^AT_EX
Templates

UCalendar

y además...

charlas

talleres

**apoyo en
proyectos**

**espacios
de trabajo**

programatones

¿Hay que saber X para colaborar?

NO

¿Cómo **yo** llegue a esto?

Sobre mí

- Generación **2019**, Major Software, Minor Data Science
- Aprendía a programar en **enero del 2019** (3 años)
- Me gustó la idea de desafíos: tener un **proyecto semestral** para **aplicar conocimientos** de los cursos

**Introducción a la programación
en Python I: Aprendiendo a
programar con Python**



Jorge Muñoz Gama [+3 instructores más](#)

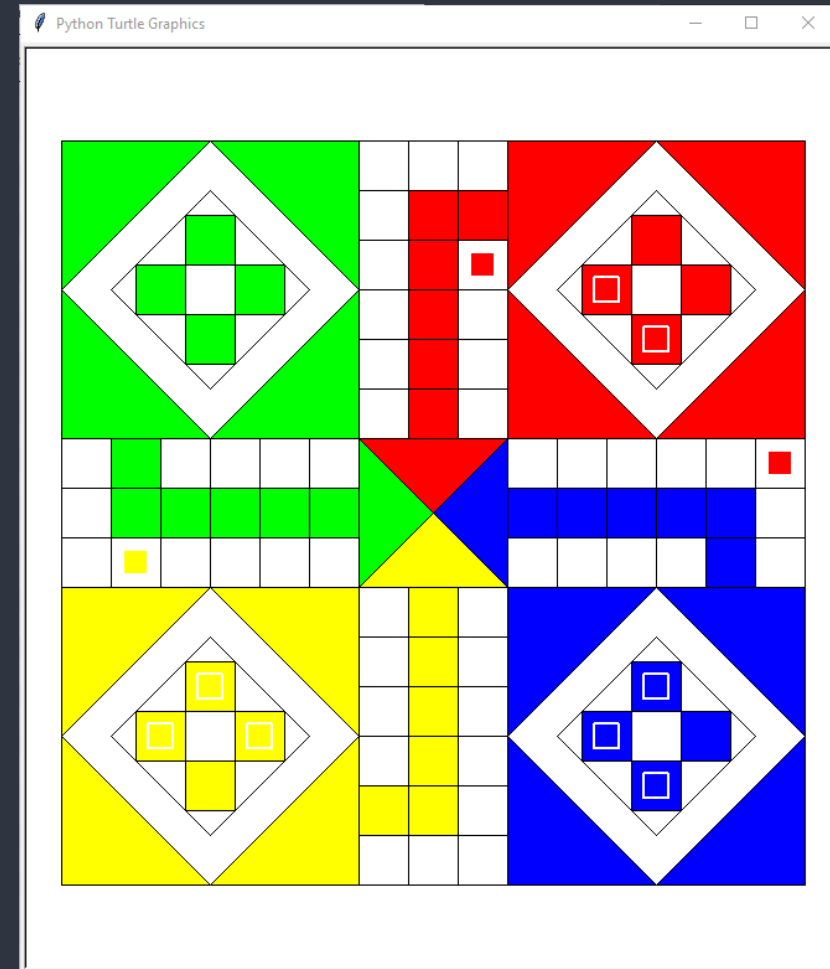
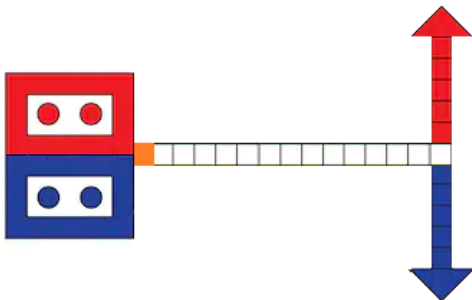
2019-2

All Contests > Tarea 1 IIC1103 [2019-2] > Parte 1 - Tarea 1 [2019-2]

Parte 1 - Tarea 1 [2019-2]

Estabas descansando del exhaustivo semestre que tomaste mientras escuchabas algunas de tus favoritas sinfonías en Stopify. Luego, en un momento, junto a un amigo fuiste transportado de la nada y encerrado en una sala del terror a merced del malévolo músico Ludo-Vico. Él les explica que para salir deben derrotarlo en su última obra maestra, el juego BiLudo una modificación del conocido [Ludo](#). Además, les explica que aún no ha tenido tiempo para realizar el juego, por lo que te pide a ti que lo programes. De lo contrario, no te dará ni la más mínima posibilidad de salir de la sala.

Para la tarea deberás programar una simplificación del Ludo, la cual solo tendrá 2 jugadores, y cada jugador poseerá 2 fichas. Durante su turno, cada jugador deberá tirar un dado para avanzar. Además, ambos jugadores iniciarán desde la misma casilla y el tablero, a diferencia del Ludo original, será una sola línea, tal como se muestra en la siguiente imagen:



```

1 def juego():
2     global meta, tortugas
3     tortugas, datos, avanzar = [], [], False
4     tablero() # Dibujar
5     jugadores = numero_jugs()
6     for x in range(jugadores):
7         datos.append([[-1,-1,-1,-1], input(t03 + str((x+1)) + ": ")])
8     plot(datos)
9     jug_act, en_juego = 1, True
10    while en_juego:
11        print(t04, datos[jug_act-1][1], "\n", t22, datos[jug_act-1][0])
12        en_turno = True
13        #####
14        # Inicio del turno #
15        while en_turno:
16            input(t05) #Enter para empezar el turno
17            dado, mov = random.randint(1,6), False
18            print(t06, dado)
19            #####
20            # Si sale dado = 6 o dado = 1 #
21            if dado == 1 or dado == 6:
22                en_ciclo = True
23                # Inicio ciclo, ya que siempre se
24                # puede hacer algo si es 1 o 6
25                while en_ciclo:
26                    accion = input(t07)
27                    #####
28                    # Inicio de acción de liberar #
29                    if accion == c1:
30                        liberando = True
31                        for x in range(4):
32                            if datos[jug_act-1][0][x] == -1 and liberando:
33                                print(t09, x+1)
34                                datos[jug_act-1][0][x] = 0
35                                en_ciclo, liberando = False, False
36                                liberar(jug_act-1, x)
37                                mov, q = True, 0
38                            if liberando: # No hay por liberar
39                                print(t14)
40                                #####
41                                # Inicio de acción de avanzar #
42                                elif accion == c2:
43                                    cual_avanzar = input(t10)
44                                    if cual_avanzar.isnumeric():
45                                        cual_avanzar = int(cual_avanzar)
46                                    if 1 <= cual_avanzar <= 4:
47                                        if -1 < datos[jug_act-1][0][cual_avanzar-1] < meta:
48                                            avanzar = True
49                                            en_ciclo = False
50                                else:

```

```

285 def tablero():
286     global colores, px
287     # crea la tortuga
288     g = turtle.Turtle()
289     # la hago invisible
290     g.ht()
291     g.speed(0)
292     # Hago el del primer jugador
293     # luego lo repito y lo rota 90°
294     for x in range(4):
295         g.hset(1)
296         g.lt(135 - (90 * x)) #clave
297         g.down()
298         # inicio relleno
299         g.begin_fill()
300         g.fillcolor(colores[x])
301         g.fd(1.5 * px + 2 ** 8.5)
302         g.rt(135)
303         g.fd(px)
304         g.lt(90)
305         g.fd(px + 5)
306         g.rt(90)
307         g.fd(px + 2)
308         g.lt(90)
309         g.fd(px)
310         g.rt(135)
311         g.rt(90)
312         g.fd(px + 6)
313         g.rt(90)
314         g.fd(px + 4)
315         g.lt(90)
316         g.fd(px)
317         g.lt(90)
318         g.fd(px + 4)
319         g.lt(90)
320         g.fd(px)
321         g.rt(135)
322         g.fd(1.5 * px + 2 ** 8.5)
323         # termino relleno
324         g.end_fill()
325         # inicio grid
326         g.rt(90)
327         g.fd(1.5 * px + 2 ** 8.5)
328         g.rt(135)
329         for _ in range(3):
330             g.fd(px + 3)
331             g.lt(90)
332             g.fd(px)
333             g.lt(90)
334             g.fd(px + 3)
335             g.rt(90)
336             g.fd(px)
337             g.rt(90)
338             g.fd(px + 3)
339             g.bk(px)
340             g.rt(90)
341             g.fd(px + 6)
342             g.rt(90)
343             g.fd(px)
344             g.rt(90)
345             g.fd(px + 6)
346             g.lt(90)
347             g.fd(px + 6)
348             g.lt(90)
349             g.fd(px + 6)
350             # termino el grid e inicia el
351             g.lt(90)
352             g.fd(px + 6)
353             g.begin_fill()
354             g.fillcolor("white")
355             g.lt(45)
356             for _ in range(4):
357                 g.fd(px + 5 + 2 ** 8.5)
358                 g.lt(90)
359                 g.end_fill()
360                 g.lt(45)
361                 g.up()
362                 g.fd(px)
363                 g.down()
364                 g.rt(45)
365             for _ in range(4):
366                 g.fd(px + 2 + 2 ** 8.5)
367                 g.lt(90)
368                 g.fd(px + 2 ** 8.5 / 2)
369                 g.lt(45)
370                 g.fd(px)
371             for _ in range(4):
372                 g.begin_fill()
373                 g.fillcolor(colores[x])
374                 for _ in range(4):
375                     g.lt(90)
376                     g.fd(px)
377                     g.end_fill()
378                 g.fd(px)
379                 g.lt(90)
380                 g.up()

```

```

def mover(jug, fic, inicial, final):
    movimiento = inicial
    if final - inicial > 0:
        for _ in range(final - inicial):
            tortugas[jug][fic].fd(px)
            movimiento += 1
            # Sistema para seguir el camino
            if movimiento in [4, 17, 30, 43]:
                tortugas[jug][fic].lt(90)
            elif movimiento in [9, 11, 22, 24, 35, 37, 48, 50, 51]:
                tortugas[jug][fic].rt(90)
            elif movimiento == 52:
                tortugas[jug][fic].lt(90)
        else:
            for _ in range(inicial - final):
                tortugas[jug][fic].bk(px)

if -1 < q <= meta - 6:
    comer_ver = jug_act % jugadores + 1
    for y in range(jugadores-1):
        dif = ((comer_ver - jug_act)%4) * 13
        if q <= dif:
            fc = 1
        else:
            fc = -1
        for z in range(4):
            v1 = q + (fc*dif) == datos[comer_ver-1][0]
            v2 = -1 <= datos[comer_ver-1][0][z] <= met
            if v1 and v2:
                datos[comer_ver-1][0][z] = -1
                print(t19, cual_avanzar, t21, dat
                comido(comer_ver-1, z)
            comer_ver = comer_ver % jugadores + 1

```

2020-1



+

iCalendar
estándar



2020-2



+



UCalendar

2020 ... 2021

Que compañía me llama

Busca si una compañía tiene registrado el número que te llamó.

+56

La búsqueda requiere al menos 5 números

- VOIP ANALYSIS S.A
- COMPATEL CHILE LIMITADA
- CLARO CHILE S.A.

Nota: El número puede ser particular, y la compañía es solo el agente registrador.



GMail filter helper

gist_id get filter from gist

```
{
  from:(noreply@uc.cl | "no-reply@uc.cl")
  ~from:("Rectoría" | "Rectoria")
}
{
  from:(evaldoc@uc.cl)
  ~("iniciado" | "ÚLTIMO DÍA" | "ULTIMOS")
}
from:{
  alimentacionsaludestudiantil@uc.cl
  | ansiedad@uc.cl
}
```

Group

Match emails from:

noreply@uc.cl

no-reply@uc.cl

Except:

Match emails from:

Rectoría

Rectoria

Group

Match emails from:

evaldoc@uc.cl

Except:

iniciado

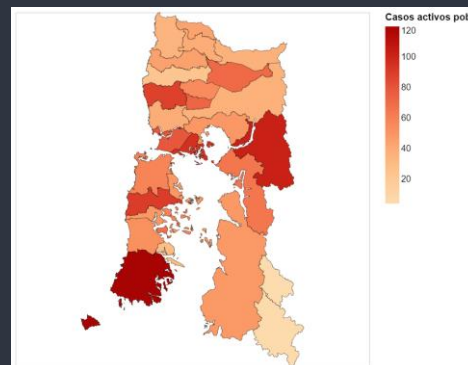
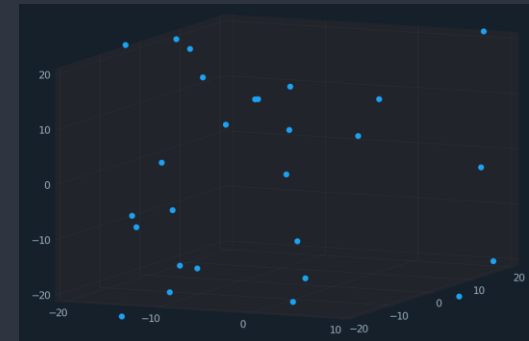
ÚLTIMO DÍA

ULTIMOS

Match emails from:

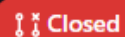
alimentacionsaludestudiantil@uc.cl

ansiedad@uc.cl



github.com/benjavicente
computationalthinking.mit.edu
benjavicente.github.io/summer-of-code-2020-2021

Add middle mouse buttom click behavior #31



Closed

benjavicente wants to merge 3 commits into `wachunei:develop` from `benjavicente:develop`



Conversation 5



Commits 3



Checks 4



Files changed 2



benjavicente commented on 12 Aug 2020

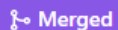
Contributor



Opción de correo UC dirige directamente a gmail #11

Edit

<> Code



Merged

wachunei merged 1 commit into `wachunei:develop` from `benjavicente:develop` on 10 Jun 2020



Conversation 1



Commits 1



Checks 0



Files changed 1

+1 -1



benjavicente commented on 10 Jun 2020 • edited

Contributor

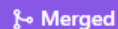


Reviewers

EverythingLite 1.4.1.988 #3328

Edit

<> Code



Merged

msftbot merged 2 commits into `microsoft:master` from `benjavicente:EverythingLite` on 30 Aug 2020



Conversation 8



Commits 2



Checks 4



Files changed 1

+23 -0



benjavicente commented on 30 Aug 2020 • edited by msftbot[bot]

bot

Contributor



Reviewers

2021

Inicia Open Source UC

2021



RamosUC

django

fix: cache posible values #54

Merged agucova merged 2 commits into main from fix-cache-good-branch on 8 Aug 2021

Conversation 6 Commits 2 Checks 2 Files changed 1



benjavicente commented on 2 Aug 2021

Member

Esta PR realiza un cambio en el funcionamiento del caché, que se resume en lo siguiente:

```
cache.get_or_set("result", expensive_computation()) # ✗  
cache.get_or_set("result", expensive_computation)    # ✓
```

`cache.get_or_set` se utiliza 3 veces en la aplicación y en todas se estaba entregando un diccionario para hacer el `set`.

El problema es que, al momento de llamar a `get_or_set` este diccionario ya está generado, y por lo tanto, también debería haberse realizado las 5 y 1 consultas respectivas de cada uno de estos casos, entonces no se está evitando repetir la operación de generar estos datos.

Para evitar esto se entrega una función que será llamada solo si no existe el elemento en caché ([documentación](#)).

Para probar la diferencia, corrí el servidor en modo de producción, cambie el tiempo de caché del cliente en `cache_control` a 0, el tiempo de caché de los datos del en `get_or_set` a 180 y borré tanto el caché del navegador como el del servidor.

El promedio de respuesta de planifica, medido por el navegador, bajó de 116ms a 30ms (n=10). Esto debe afectar también al creador de horarios en la misma proporción, y en una proporción menor al navegador de cursos.

Además junté los cachés en 1, ya que 2 poseían la misma información y el último tenía un subconjunto de datos del los otros 2.

solo 1 cambio:

agregar 1 función y mover unas 7 líneas
(sin conocer django!)

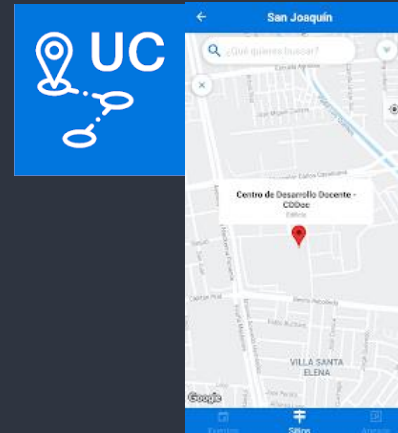
=

respuestas sobre

2 veces más rápidas

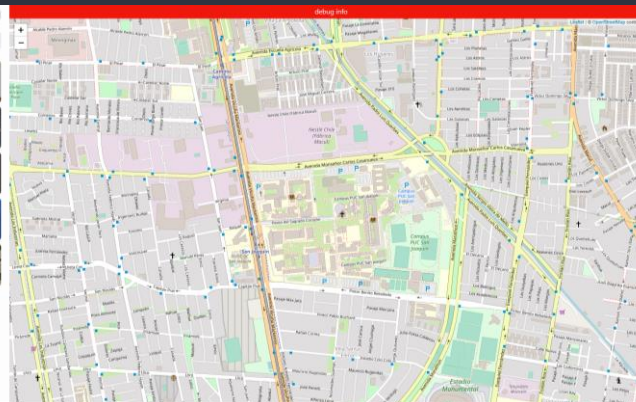
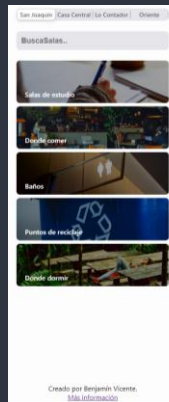
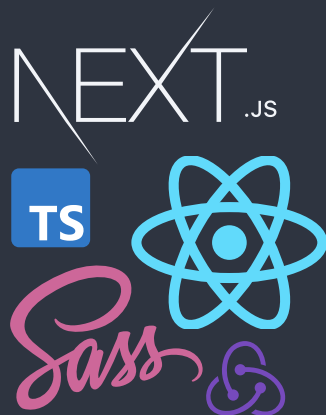
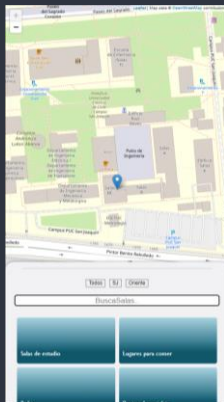
2021

- Problema: 3 gens. que recién conocerán el campus
- ¿Hay soluciones? **Sí, 2**
- ¡Pero se pueden mejorar!
- Una tiene **datos abiertos** ✨

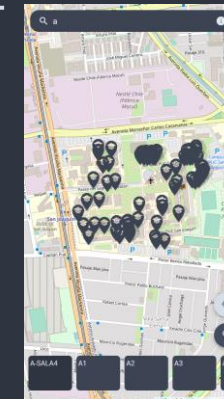


play.google.com/store/apps/details?id=cl.puc.SmartCampus

UbiCate



SVELTEKIT

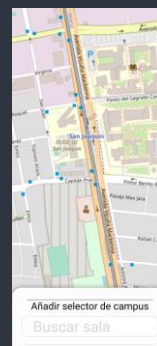
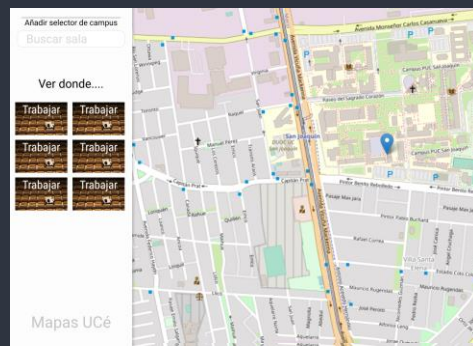


Código

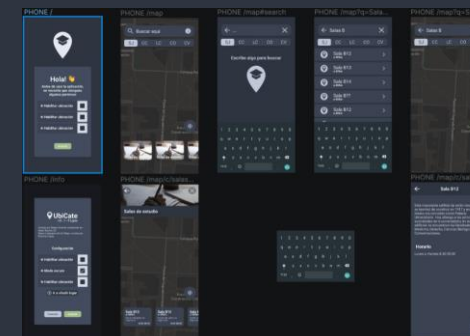
Diseños



Simplex

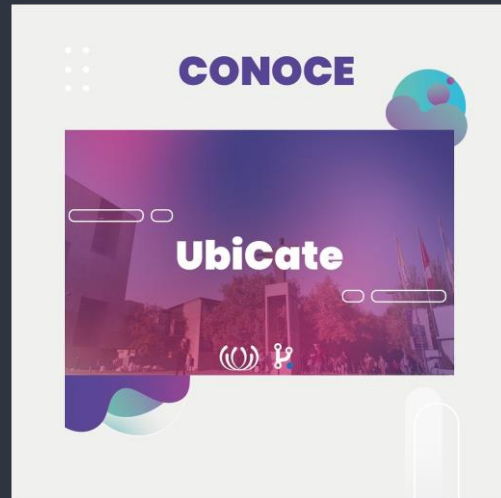


Interactivos

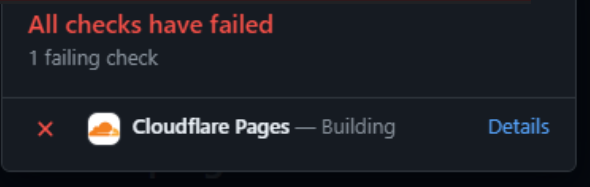


github.com/benjavicente/uc-maps-beta
github.com/open-source-uc/busca-salas
github.com/open-source-uc/UbiCate

UbiCate



sobre 2k visitas!



www.instagram.com/p/Caw1LoguSNG

¿Fue un fracaso?


NO

No fue un fracaso


- **Aprendí muchas cosas** en el proceso
- Cree una **herramienta que le sirvió a muchas personas**
- Proyecto abierto para que **cualquiera lo pueda mejorar** o rehacer en un futuro

¿Cómo **tú** te puedes unir?

Ve y programa código abierto

- ¡Crea **repositorios públicos**!
- **Comparte** y **apoya** ideas y proyectos
- Busca **temas** y **problemas** que te interesen
- No esperes que todo funcione, **busca aprender** 

Únete a la comunidad

- Únete a la comunidad **Open Source eUC** 
- **Siempre hay una comunidad** con intereses similares
- **No se pierde nada** si uno “solo está”
- Atrevete a **preguntar** y **ayudar** a gente

Gracias!



**Benjamín
Vicente**

@benjavicente

GitHub: open-source-uc

Instagram: opensource_euc

Telegram: [@open_source_uc](https://t.me/+kEnfzvaMRHQ1ZjYxy)

