

```
from turtle import *
from time import sleep
from random import randint, choice
# nastroyka
text_enter = ""
👋 Вітаємо, колоністе!
📡 Передпольотна перевірка систем...
⚙️ Двигуни: ГОТОВІ.
🧪 Кисневі балони: ЗАПОВНЕНІ.
📦 Вантаж колоніста: ЗАВАНТАЖЕНО.
⚠️ УВАГА: Рівень небезпеки — ВИСОКИЙ.
⚠️ Будь готовий: на тебе чекають суворі випробування та небезпеки.
🌀 Піщані бурі можуть пошкодити обшивку.
☀️ Екстремальні потепління випробують твій костюм на міцність.
💀 Пам'ятай: на цій планеті помилок не вибачають.
🚀 Марс — твоя головна ціль.
🌟 Попереду 200 мільйонів кілометрів порожнечі.
🚀 Приготуйтеся до відправлення на Марс! Зворотний відлік розпочато.
Міцніше тримайтеся за штурвал!
""
```

```
text_legend = f""
📖 Легенда об'єктів:
```

- 1 ⚡ Джерело струму - надає енергію
 - 2 🌳 Древа — виробляють кисень
 - 3 🏠 База — збільшує населення
 - 4 🥕🥦🍆🍅🥔🥕 Огород — дає їжу
 - 5 🛡️ Щит — підвищує захист
- ```
""
```

```
colors = ["#FFFF00", "#00FF00", "#FF00FF", "#00FFFF", "#FF4500", "#ADFF2F",
"#FF1493", "#1E90FF", "#00FF7F", "#FFD700"]
```

```
energy = 100
```

```
oxygen = 100
```

```
population = 50
```

```
food = 100
```

```
defense = 100
```

```
text_resours = f""
```

```
📦 Ресурси та стан бази 📊:
```

```
⚡ Енергія: {energy}
```

```
📡 Кисень: {oxygen}
```

```
👤🚀 Населення: {population}
```

```
📦🍷🍔 Запаси їжі: {food}
```

```
🛡️ Захист: {defense}
```

```
""
```

```
mapa_mars = "0000000000"*10
```

```
screen = Screen()
```

```

screen.setup(width=1000, height=600)
screen.bgcolor("black")
screen.tracer(0)
def make_t(col):
 t = Turtle()
 t.pu()
 t.color(col)
 t.ht()
 return t
t_info = make_t("white")
t_mapa = make_t("black")
t_res = make_t("white")
t_rocket = make_t("white")
t_mission = make_t("blue")
t_timer = make_t("blue")
t_stars = make_t("white")
t_mars = make_t("red")
t_finish = make_t("gold")
t_planet = make_t("purple")
t_shtorm = make_t("grey")
t_attack = make_t("lime")
t_talk = make_t("white")
finish = None
mission = False
mission_res=0
mission_value=0
mission_time=0
cell = 45
x, y = -60, 180
#функція 3 2 1
def fidlic(a):
 for i in range(a,0,-1):
 for y in range(20, 120, 10):
 c= colors[i-1]
 star()
 draw_mars(424, 216, 70)
 t_mars.clear()
 t_mars.goto(-50,0)
 t_mars.color(c)
 t_mars.write(i, font = ("Arial", y,"bold"))
 screen.update()
 sleep(0.02)
 sleep(0.5)
 t_mars.clear()
def fidlice(a, b):
 for i in range(a, 0, -1):
 for y in range(20, 120, 10):
 c = colors[i-1]

```

```

 t_timer.clear()
 t_timer.goto(-50, b)
 t_timer.color(c)
 t_timer.write(i, font=("Arial", y, "bold"))
 screen.update()
 sleep(0.02)
 sleep(0.5)
t_timer.clear()
screen.update()

#функція марса
def draw_mars(x, y, size):
 t_mars.goto(x, y - size)
 t_mars.begin_fill()
 t_mars.color("#cc3300")
 t_mars.circle(size)
 t_mars.end_fill()
 for i in range(5):
 t_mars.goto(x + randint(-40, 40), y + randint(-40, 40))
 t_mars.color("#992200")
 t_mars.dot(randint(10, 25))

#функція зірок
def star():
 stars_list = []
 for i in range(80):
 stars_list.append([randint(-500, 500), randint(-300, 300)])
 t_stars.clear()
 for s in stars_list:
 if randint(1, 10) > 1:
 t_stars.goto(s[0], s[1])
 t_stars.dot(randint(1, 4), "white")

функція тексту основна
def wrote_text(x, y, text, t,shr,speed):
 lines = text.split("\n")
 for line in lines:
 show = ""
 for char in line:
 show += char
 t.goto(x, y)
 t.write(show, font=("Arial", shr, "bold"))
 sleep(speed)
 screen.update()
 y -= 30

функція тексту2
def wroter_text(x, y, text, t):
 lines = text.split("\n")
 for line in lines:
 show = ""

```

```

for char in line:
 star()
 draw_mars(424, 216, 70)
 show += char
 t.goto(x, y)
 t.write(show, font=("Arial", 16, "bold"))
 screen.update()
 sleep(0.05)
y -= 30
#функція малювання поля
def draw_mapa():
 xx, yy = x, y
 for i in range(10):
 for j in range(10):
 t_mapa.width(4)
 t_mapa.pu()
 t_mapa.goto(xx,yy)
 t_mapa.pd()
 for k in range(4):
 t_mapa.fd(cell)
 t_mapa.rt(90)
 screen.update()
 sleep(0.01)
 xx+=cell
 xx = x
 yy -=cell
#функція анімації
def animes(x,y,emo,t,a,b,f):
 for i in range(f):
 star()
 draw_mars(424, 216, 70)
 t.clear()
 t.goto(x,y)
 t.write(emo, font = ("Arial", 33 , "bold"))
 screen.update()
 x+=a
 y+=b
 sleep(0.03)
 t.clear()
 t.goto(x, y)
 t.write(emo, font=("Arial", 30, "bold"))
 screen.update()

def animer(x,y,emo,t,a,b,f):
 for i in range(f):
 star()
 draw_mars(-380, -216, 70)
 t.clear()

```

```

t.goto(x,y)
t.write(emo, font = ("Arial", 33 , "bold"))
screen.update()
x+=a
y+=b
sleep(0.03)
t.clear()
t.goto(x, y)
t.write(emo, font=("Arial", 30, "bold"))
screen.update()
#Функція фону
def draw_marsfon():
 tracer(0)
 screen.bgcolor("#BC5434")
 crater_col = "#8D3B28"
 for i in range(15):
 th = make_t(crater_col)
 while True:
 x = randint(-500, 500)
 y = randint(-300, 300)
 if not (-500 <= x <= 0 and -100 <= y <= 250):
 break
 th.goto(x, y)
 th.pd()
 th.begin_fill()
 th.circle(randint(10, 40))
 th.end_fill()
 screen.update()



```

```
def update_res():
```


```




t_res.clear()
text_resours = f""




```


 Ресурси та стан бази :

 Енергія: {energy}

 Кисень: {oxygen}

   Населення: {population}

   Запаси їжі: {food}

 Захист: {defense}

""""

```

t_res.clear()
wrote_text(-500,-50,text_resours,t_res,16,0.01)

```

#функція будівництва

```
def build():
```

```

global mapa_mars

```

```
try:
```

```

num_cell = int(input("Номер клітинки від 1-100:"))-1

```

```

num_build = int(input("🏠 Що хочеш побудувати 1-5 згідно з легендою: "))

```

```

if num_cell >=100 or num_cell < 0:

```

```

 print("Такої клітинки не існує ❌ ")
 return
if num_build <1 or num_build>5:
 print("Такої будівлі не існує ❌ ")
 return
if mapa_mars[num_cell] != "0":
 print("⚠ Тут вже є будівля")
 return
if not check_res(num_build):
 print("Ресурсів для побудови не вистачає ❌ ")
 return
mapa_mars = mapa_mars[:num_cell] + str(num_build) + mapa_mars[num_cell+1:]
if num_build == 1: emoji = "⚡"
elif num_build == 2: emoji = "🌳"
elif num_build == 3: emoji = "🏠"
elif num_build == 4:
 veggies = ["🥕", "🌿", "🍆", "🍅", "🥔", "🥕"]
 emoji = choice(veggies)
elif num_build == 5: emoji = "🛡"
col = num_cell%10
row = num_cell//10
xx = x + cell * col -1
yy = y - cell * row -34
t_mapa.pu()
t_mapa.goto(xx, yy)
t_mapa.write(emoji, font=("Arial", 30, "normal"))
update_res()
screen.update()
except ValueError:
 print("⚠ Помилка! Вводь тільки ЦИФРИ, а не текст!")
#функція перевірки ресурсів
def check_res(nb):
 global defense, food, population, oxygen, energy
 if nb ==1 and food >=10 and oxygen >=10 and population >=5:
 energy+=30
 food-=10
 oxygen -=10
 return True
 elif nb ==2 and energy>=10 and population >=1 and food >= 5 and oxygen >=10:
 oxygen+=25
 food-=10
 energy-=5
 return True
 elif nb ==3 and energy>=20 and food >= 20 and oxygen >=20:
 oxygen-=10
 food-=10
 energy-=10
 population +=15

```

```

return True
elif nb ==4 and energy>=30 and food >= 10 and oxygen >=10 and population >=3:
 oxygen-=10
 food+=30
 energy-=10
 return True
elif nb ==5 and energy>=30 and food >= 10 and oxygen >=10 and population >=10:
 energy-=10
 food-= 10
 oxygen-=10
 population +=5
 defense +=25
 return True
return False

```

#функція діалогу поєднання вайт та з'являються персонажі що розмовляють

```

def wrote_dialog_text(x, y, text, shr, speed, side):
 bot_p, man_p = (-450, 140), (250, 140)
 lines = text.split("\n")
 history = []
 #тут все так як і в функції wrote
 for line in lines:
 show = ""
 for char in line:
 show += char
 t_talk.clear()
 # Малюємо робота та присвоюємо йому ліву сторону
 bx, by = bot_p
 if side == "left":
 bx += randint(-5, 5)
 by += randint(-5, 5)
 t_talk.goto(bx, by)
 t_talk.write("🤖", font=("Arial", 120, "bold"))
 # Малюємо колоніста та присвоюємо йому праву сторону
 mx, my = man_p
 if side == "right":
 mx += randint(-5, 5)
 my += randint(-5, 5)
 t_talk.goto(mx, my)
 t_talk.write("😁", font=("Arial", 120, "bold"))
 # 3. Виводимо текст старі рядки
 temp_y = y + 150
 for old_line in history:
 t_talk.goto(x - 60, temp_y)
 t_talk.write(old_line, font=("Arial", shr, "bold"))
 temp_y -= 40
 # 4. Виводимо рядки нові
 t_talk.goto(x - 60, temp_y)
 t_talk.write(show, font=("Arial", shr, "bold"))

```

```

 screen.update()
 sleep(speed)
 history.append(line) # і додаємо все в список
#функція говоріння
def talk(text, side):
 txt_x = -420 if side == "left" else -90
 wrote_dialog_text(txt_x, -120, text, 22, 0.05, side)
 sleep(2.5)
склеюємо все до кучі в 1 діалог
def first_dialog():
 t_talk.clear()
 talk("Привіт, колоністе! Я твій робот-помічник EARTH_26 з Землі.\nНарешті ми
прибули на Марс!", "left")
 sleep(0.5)
 plan = ""
 План місії:
 🏠 Побудувати першу житлову колонію та базу
 💎 Знайти поклади мінералів 💧
 🍅 🥔 🥕 Створити марсіанський огороz 🌽 🥦 🍆
 🌱 Насадження рослин та дерев 🌳
 ☀️ Розгорнути сонячні панелі ⚡️
 🛡️ Створити захисний щит 🗝️
 🌬️ Налагодити постачання кисню ⚙️
 ""
 talk(plan, "left")
 sleep(0.4)
 talk("Це виглядає легко, EARTH_26.\nСподіваюсь, я впораюсь з планом.", "right")
 sleep(0.4)
 talk("Але це ще не все. На Марсі датчики фіксують загрози. ⚠️\nНасуваються бурі та
шторми 🌀.\nАтаки марсіанів 👁️ та сонячні спалахи☀️.\nТому не варто
розслаблятись!", "left")
 sleep(0.4)
 talk("Ого! Я про це не знав.\nДякую, що ти попередив мене!", "right")
 sleep(0.4)
 talk("Немає за що!\nЦе моя робота.\nУдачі колоністе!!!", "left")
 sleep(0.4)
 t_talk.clear()
 screen.update()

def second_dialog():
 t_talk.clear()
 talk("Вітаю, колоністе!\n", "left")
 sleep(0.2)
 text_win = ""
 🌟 МІСІЯ ВИКОНАНА! МАРС ПІДКОРЕНО! 🚀
 ✅ Усі завдання завершено
 🏠 Першу житлову колонію успішно побудовано!
 ⚡️ Джерело енергії створено, енергія стабільна.

```

- 🌬️ Постачання кисню працює без збоїв.
- 💧 Ресурси стабілізовано, Марс став рідним домом.
- 👨‍🚀 Екіпаж у безпеці. Ти вписав своє ім'я в історію!
- 🏆 ТИТУЛ: Найвеличніший колоніст Галактики! 🌟

\*\*\*\*\*

```

talk(text_win, "left")
sleep(0.4)
talk("Ура! Ми це зробили, EARTH_26! 😎\nПодивись на цю базу, на цей
огород...\nЧервона планета стала нашою домівкою!", "right")
sleep(0.4)
talk("Так, тут справді було багато роботи!", "left")
sleep(0.4)
talk("Це неймовірно... ПЛАНЕТУ ОЖИВЛЕНО! 🌍🌟\nМАРС ТЕПЕР ПОВНІСТЮ
ПРИДАТНИЙ ДЛЯ ЖИТТЯ!\nСкафандри більше не потрібні.", "right")
sleep(0.4)
talk("Повідомляю Землю: Експедиція успішна ✅.\nТепер можна залишатись тут
жити!" , "left")
sleep(0.4)
t_talk.clear()
screen.update()

```

```
def third_dialog():
```

```

t_talk.clear()
talk("К.к.к.колоністе!\n", "left")
sleep(0.2)
text_lose = ""
🚨 МІСІЯ ПРОВАЛЕНА... ЗВ'ЯЗОК ПЕРЕРВАНО 🚨
🌪️ Шторм розірвав купол колонії та засипав модулі піском.
👁️ Марсіанська атака пробила захист — периметр втрачено.
☀️ Сонячний спалах спалив електроніку та залишки кисню.
💀 Ресурси вичерпано. Життєзабезпечення бази — 0%.
👨‍🚀 Екіпаж не витримав натиску Червоної планети...
📉 Проєкт колонізації офіційно визнано катастрофою.
🌪️ Марс не пробачає помилок. Повертайся сильнішим!

```

\*\*\*\*\*

```

talk(text_lose, "left")
sleep(0.4)
talk("Мені жаль, що я не зміг впоратись. ", "right")
sleep(0.4)
talk("...", "left")
sleep(0.4)
talk("Але я обов'язково повернуся сильнішим!", "right")
sleep(0.4)
talk("Тоді приготуйся до відправлення додому, поки є можливість!" , "left")
sleep(0.4)
t_talk.clear()
screen.update()
#функція класного шторму

```

```

def draw_storm():
 # кольори
 cols = ["#8D3B28", "#7D7D7D", "#4A4A4A", "#BC5434"]
 objs = ["⚡", "🌳", "🌽", "🏠", "🛡️", "🌪️", "☁️", "🌀", "⚙️"]
 #блیمانня фону
 for i in range(50):
 if i % 2 == 0:
 screen.bgcolor("#5A5A5A")
 else:
 screen.bgcolor("#BC5434")
 t_shtorm.clear()
 # ефект тряски
 ox = randint(-20, 20)
 oy = randint(-20, 20)
 # полоски
 for i in range(120):
 t_shtorm.penup()
 tx = randint(-600, 600) + ox
 ty = randint(-400, 400) + oy
 t_shtorm.goto(tx, ty)
 t_shtorm.color(choice(cols))
 t_shtorm.pensize(randint(2, 6))
 t_shtorm.pendown()
 t_shtorm.setheading(randint(170, 190))
 t_shtorm.forward(randint(60, 180))
 # уламки
 for i in range(4):
 t_shtorm.penup()
 ex = randint(-450, 450)
 ey = randint(-300, 300)
 t_shtorm.goto(ex, ey)
 emoji = choice(objs)
 t_shtorm.write(emoji, font=("Arial", 65, "bold"))
 screen.update()
 sleep(0.05)
 # До звичайного вигляду
 screen.bgcolor("#BC5434")
 t_shtorm.clear()
 screen.update()
#функція про місію
def earth_mission():
 steps = [10, 15, 20, 25, 30]
 global mission, mission_res, mission_value, mission_time
 if mission:
 return
 if randint(1, 2) == 1:
 mission = True
 mission_res = randint(1, 5)

```

```

mission_time = randint(20, 40)
list_res = [0,energy,oxygen,population, food, defense]
mission_value =list_res[mission_res] + choice(steps)
emojis_res =["", "⚡", "🌳", "🏠", "🌽", "🛡️"]
emojis = emojis_res[mission_res]
wrote_text(-470,250,f"📡 Земля: новий квест 📁. Потрібно підняти баланс до
{mission_value} {emojis} за {mission_time} 🕒. Початок...", t_mission,18,0.05)
mission_timer()
#функція про таймер місії
def mission_timer():
 global mission_time
 if mission:
 t_timer.clear()
 wrote_text(-270,190, f"⌚ :{mission_time}", t_timer, 30 ,0.05)
 screen.update()
 if mission_time <=0:
 t_timer.clear()
 wrote_text(-400,300, "Час вийшов 🚫!", t_timer, 15, 0.05)
 check_mission()
 else:
 mission_time -= 1
 screen.update()
 screen.ontimer(mission_timer,1000)
#функція перевірки місії
def check_mission():
 global mission, energy,oxygen,population, food, defense
 mission = False
 emojis_res =["", "⚡", "🌳", "🏠", "🌽", "🛡️"]
 list_res = [0,energy,oxygen,population, food, defense]
 if mission_value<= list_res[mission_res]:
 if randint(1, 2) == 1:
 mission_resurs = randint(1, 5)
 list_res[mission_resurs] += 25
 energy, oxygen, population, food, defense = list_res[1], list_res[2], list_res[3],
list_res[4], list_res[5]
 text = f"📡 Земля: 🏆 Успіх! Бонус + 25 до {emojis_res[mission_resurs]} ресурсу!"
 else:
 energy += 5; oxygen += 5; population += 5; food += 5; defense += 5
 text = "📡 Земля: 🏆 Успіх! Вся база отримала по +5 до кожного з ресурсів!"
 else:
 if randint(1, 2) == 1:
 mission_resurs = randint(1, 5)
 list_res[mission_resurs] -= 20
 energy, oxygen, population, food, defense = list_res[1], list_res[2], list_res[3],
list_res[4], list_res[5]
 text = f"📡 Земля: ❌ Місія провалена втрачено 20 {emojis_res[mission_resurs]}
ресурсу!"
 else:

```

```

 energy -= 5; oxygen -= 5; population -= 5; food -= 5; defense -= 5
 text = "🌍 Земля: ❌ Місія провалена втрачено -5 з кожного ресурсу!"
t_mission.clear()
wrote_text(-400,260, text, t_mission, 15, 0.05)
update_res()
screen.update()
screen.ontimer(clear_mission, 3000)
def clear_mission():
 t_timer.clear()
 t_mission.clear()
#функція природні катастрофи
def mars_disaster():
 global energy, oxygen, population, food, defense
 n = randint(1, 5)
 if n ==1:
 wrote_text(-400,270,"🌀 УВАГА: НА МАРСІ НАСУВАЄТЬСЯ ШТОРМ! РУЙНУВАННЯ
БУДІВЕЛЬ!", t_planet, 15, 0.05)
 draw_storm()
 storm()
 elif n ==2:
 wrote_text(-400,270, "👁️ УВАГА: НЕВІДОМИЙ ОБ'ЄКТ НА РАДАРАХ! МАРСІАНИ
НАБЛИЖАЮТЬСЯ!", t_planet, 15, 0.05)
 draw_attack()
 attack()
 elif n == 3:
 wrote_text(-410,270, "☢️ УВАГА: ВИЯВЛЕНО СОНЯЧНИЙ СПАЛАХ! ☀️", t_planet,
15, 0.05)
 draw_solar()
 drought()
 screen.ontimer(clear_mars,2000)
#функція шторму
def storm():
 global oxygen, population, food, defense, mapa_mars
 t_planet.clear()
 #mapa_mars = mapa_mars.replace()
 if defense > 100:
 food-=10
 oxygen-=5
 defense-=5
 wrote_text(-400,247,"🌀 ШТОРМ ЧАСТКОВО ВІДБИТО ⚠️! ВТРАТИ ПО -5 і -10!",
t_planet, 15, 0.05)
 sleep(3)
 else:
 population-=10
 food-=20
 oxygen-=10
 defense-=10

```

```

wrote_text(-400,247,"💣 КРИТИЧНІ РУЙНУВАННЯ! Шторм пробив обшивку
❌Втрати по -10 і -20!", t_planet, 15, 0.05)
sleep(3)
t_planet.clear()
update_res()
#функція атаки марсіан
def draw_attack():
 # Ядрьоні кольори
 laser_cols = ["#00FF00", "#39FF14", "#7FFF00", "#ADFF2F"]
 fo_objs = ["👁️", "👁️", "💣", "🔥", "💧", "🚀"]
 # 60 циклів тобто довжина атаки 60 кадрів
 for i in range(60):
 # блимання з зміною фону на фіолетовий
 if i % 2 == 0:
 screen.bgcolor("#310062")
 else:
 screen.bgcolor("#BC5434")
 t_attack.clear()
 # багато паличко типу лазери
 for i in range(25):
 t_attack.pensize(randint(2, 8))
 t_attack.color(choice(laser_cols))
 t_attack.penup()
 t_attack.goto(randint(-600, 600), 400)
 t_attack.pendown()
 t_attack.goto(randint(-500, 500), randint(-350, 100))
 t_attack.penup()
 # появлення різних емодзі
 for i in range(5):
 t_attack.penup()
 ex = randint(-480, 480)
 ey = randint(-320, 320)
 t_attack.goto(ex, ey)
 emoji = choice(fo_objs)
 t_attack.write(emoji, font=("Arial", 75, "bold"))
 screen.update()
 sleep(0.04)
 # повертаємо фон
 screen.bgcolor("#BC5434")
 t_attack.clear()
 screen.update()
#функція сонячної атаки
def draw_solar():
 # 1. розміри сонця та його малювання
 for size in range(10, 600, 30):
 t_solar.clear()
 t_solar.goto(0, -size/4)
 # промені

```

```

t_solar.color("#FFA500")
t_solar.pensize(size/10)
for i in range(16):
 t_solar.pu()
 t_solar.goto(0, 0)
 t_solar.setheading(i * 30 + size/2) # щоб вони ніби крутилися
 t_solar.pd()
 t_solar.forward(size + 20)
 t_solar.pu()
центр сонця
t_solar.goto(0, 0)
t_solar.color("#FFD700")
t_solar.dot(size)
screen.update()
sleep(0.03)
2. різні емоджі та ефект засліплення
solar_objs = ["☀️", "☢️", "🔥", "💥", "⚠️"]
for i in range(80):
 # зміна кольору на білий з шансом 50 на 50
 if i % 2 == 0:
 screen.bgcolor("white")
 else:
 screen.bgcolor("#FFFFFFE0")
 t_solar.clear()

кружки типу спалахи
for i in range(5):
 t_solar.goto(randint(-400, 400), randint(-300, 300))
 t_solar.color(choice(["#FFD700", "white", "#FFA500"]))
 t_solar.dot(randint(200, 700))

емоджі
for i in range(6):
 t_solar.goto(randint(-450, 450), randint(-300, 300))
 t_solar.write(choice(solar_objs), font=("Arial", 80, "bold"))
screen.update()
sleep(0.07)
повертаємо старий фон
screen.bgcolor("#BC5434")
t_solar.clear()
screen.update()
def attack():
 global energy, oxygen, population, food, defense, mapa_mars
 k = randint(50,150)
 t_planet.clear()
 if defense>k and defense<150:
 wrote_text(-400,247, "👁️ АТАКУ ВІДБИТО, АЛЕ Є ВТРАТИ ПО -5 і -10! ⚠️", t_planet,
15, 0.05)
 sleep(3)

```

```

 population-=5
 energy-=5
 food-=10
 defense-=5
elif defense<k:
 population-=10
 energy-=20
 food-=30
 defense-=20
 wrote_text(-400,247, "😱 АТАКУ НЕ ВІДБИЛИ. 🚨 КРИТИЧНИЙ ПРОРИВ ТА ВТРАТИ
ПО -15 I -10 🚨!", t_planet, 15, 0.05)
 sleep(3)
else:
 defense -=20
 population +=5
 wrote_text(-400,247, "✅ АТАКУ МАРСІАНІВ УСПІШНО ВІДБИЛИ 🛡️. Щити
витримали навантаження на 100%, щит -20, жителі +5.", t_planet, 15, 0.05)
 sleep(3)
t_planet.clear()
update_res()
def drought():
 global energy, oxygen, population, food, defense, mapa_mars
 t_planet.clear()
 if defense > 100 and oxygen> 120:
 food-=5
 oxygen-=5
 defense-=15
 wrote_text(-410,247,"☀️ СОНЯЧНИЙ СПАЛАХ ЧАСТКОВО ВІДБИТО!, АЛЕ Є
ВТРАТИ ПО -5 i -10! ⚠️", t_planet, 15, 0.05)
 sleep(3)
 elif defense > 100 and oxygen< 120:
 energy-=5
 population-=5
 food-=10
 oxygen-=10
 defense-=10
 wrote_text(-410,247,"☀️ СОНЯЧНИЙ СПАЛАХ ЧАСТКОВО ВІДБИТО!, АЛЕ Є
ВТРАТИ ПО -10 i -15! ⚠️", t_planet, 15, 0.05)
 sleep(3)
 else:
 energy-=15
 population-=10
 food-=20
 oxygen-=20
 defense-=20
 wrote_text(-400,247,"🚨 КРИТИЧНА РАДІАЦІЯ! Магнітний щит пробито ❌. Втрати
від -10 по -20", t_planet, 15, 0.05)
 sleep(3)

```

```

t_planet.clear()
update_res()
def clear_mars():
 t_planet.clear()
#перевірка виграшу Я не встиг доробити недоліки проєкта!
def check_finish():
 global finish, defense, food, population, oxygen, energy
 resours = [defense, food, population, oxygen, energy]
 for res in resours:
 if res <= 0:
 finish = "lose"
 break
 if defense >250 and food > 250 and population>150 and oxygen>250 and energy>250:
 finish = "win"
 elif mapa_mars.find("0") == -1:
 min_res = min(resours)
 finish = "win"
 return finish

#склеюємо гру
wroter_text(-340,200,text_enter,t_info)
sleep(2)
t_info.clear()
fidlic(10)
animes(-510, -310, "🚀", t_rocket, 15, 9,60)
sleep(1)
wrote_text(60,220,"Посадка пройшла успішно!\nТи на марсі 🪐!",t_info,16,0.05)
t_info.clear()
sleep(3)
t_rocket.clear()
t_timer.clear()
t_stars.clear()
t_mars.clear()
draw_marsfon()
screen.update()
first_dialog()
fidlice(5,0)
draw_mapa()
wrote_text(-495,180,text_legend,t_info,16,0.05)
update_res()
screen.update()
while True:
 build()
 n = randint(1,6)
 if n == 1:
 mars_disaster()
 elif n ==6:
 earth_mission()

```

```
 if check_finish():
 break
mission = False
t_info.clear()
t_res.clear()
t_mapa.clear()
t_mission.clear()
if finish == "lose":
 sleep(1)
 third_dialog()
 sleep(0.5)
 fidlic(10)
 animer(-380, -216, "🚀", t_rocket, 15, 9,80)
 print("Ти пройшов гру на погану кінцівку!")
elif finish == "win":
 sleep(1)
 second_dialog()
 print("Ти пройшов гру на хорошу кінцівку!")
```