

Гранично оптерећење - упуство за дизајнирање мисије

Аркадна кабинска летачка игра са кампањама приповедачког типа.

За више детаља посетите <http://www.limitload.org>.

Сажетак

Сврха овог упутства је да упозна потенцијалног дизајнера са принципом креирања типичне мисије у оквиру игре. Одмах напомињемо да не постоји графички интерфејс и да је неопходно основно познавање програмирања, а потом и познавање основа програмирања у питону, тако да ће цело упутство бити писано с претпоставком да читалац разуме све основне појмове (као што су „функција“, „класа“, „петља“, „објекат“, „стринг“, итд). С тим у виду, процес креирања мисије је заправо врло једноставан док год се не изађе ван оквира досад успостављених образаца. Ићи ћемо корак по корак у објашњавању кода унутар датог примера. Фокус ће бити на два примера. Први пример се бави једноставном мисијом која укључује само најнужније за криерање простог ваздушног окршаја, а други, компликованији, бавиће се и додатним системима који су на располагању, попут покретање дијалога током мисије, више зона у оквиру једне мисије и сл.

Основе

Пре него што пређемо на примере мисија чији ће код бити подробно објашњен, потребно је упознати дизајнера са неким основним поставкама начина на који се типична мисија скриптује.

- Свака мисија се састоји од:

- заглавља
- једне или више зона
- неограниченог броја дијалога током мисије
- сета специфичних опција или заставица
- три могућа општа дијалога на позорници, ван оквира саме мисије

Први, једноставнији пример ће се бавити само заглављем и једном зоном у оквиру целе мисије, што је све најнеопходније да би се направила једна мисија која може да се покрене.

- Зоне су целине у оквиру мисије у којима се одвија сва акција. Свака зона се састоји од три функције:

- улазна функција зоне
- функција петље зоне
- излазна функција зоне

Улазна функција зоне је функција која се прва одврти по учитавању зоне, само једанпут, од почетка до краја. *Функција петље зоне* је функција у којој је смештен главни контекст зоне, цела акција и сви догађаји у оквиру зоне, и она има два мода. Први је проточни мод, што значи да се, као и улазна функција, одврти само једанпут од почетка до краја. Други је понављајући мод, где ће се функција петље вртети од почетка до краја у сваком фрејму (или у свакој секунди, ако се на крај функције стави `return zc.world, 1.0`). У принципу дизајнер зоне треба да изабере на основу тога

шта му је лакше за контекст радње мисије коју је замислио. Проточна функција петље је згоднија за мисије у којима је основни циљ нешто просто, али је радња испрекидана мноштвом скриптованих момената. Понављајућа функција петље је згоднија за мисије где је циљ компликовнији и где нема много скриптоване радње. Подразумевано, функција петље је понављајућа, међутим, ако се било где унутар те функције постави исказ `yield zc.world, float`, функција се аутоматски трансформише у проточну. Можда вреди напоменути да смо ми најчешће користили мод понављајуће функције петље зоне, а тек повремено проточни. Најзад, *излазна функција зоне* је функција која се врти последња по изласку из зоне, само једанпут од почетак до краја.

Теоретски, функција петље зоне је главна функција зоне, и све што је потребно за једну зону се може нагурати у њу, али практично то није препоручљиво из разних разлога. Зато је најбоље написати све три функције за сваку зону и по њима логички разврстати контекст зоне. Нпр. у улазној функцији зоне логично је да се креира свет (земља, облаци и слично), затим поставе разни објекти (попут зграда на земљи и уопште сви они објекти који треба да су ту одмах чим се зона учита), најзад створити играча (мада то није увек логично јер по некад има смисла створити играча накнадно). У излазној функцији зоне има смисла поставити све што се тиче памћења неког стања играча, разградње света унутар зоне, и слично.

- У свакој мисији, било да је она део неке кампање или обичан окршај, функције зоне користе три објекта:
 - `zc` (од „zone context“)
 - `mc` (од „mission context“)
 - `gc` (од „game context“)

На овим трима објектима могу се додељивати атрибути, и они аутоматски имају додељену вредност `None` ако се читају пре него што им је претходно додељена нека конкретна вредност. Један прост услов унутар неке од функција зоне може да изгледа овако:

```
if not zc.enemy_arrive and not mc.objective_patrol_1:
```

када притом нигде претходно нису додељени атрибути `enemy_arrive` и `objective_patrol_1`. Тада ће оба имати аутоматску вредност `None`. Напоменимо да се у питону у логичким изразима `None` третира исто као `False`.

Која је разлика између ових објеката?

Објекат `zc` је везан искључиво за зону унутар мисије у којој се користи. Сви атрибути додељени овом објекту нестају и, ако треба, поново се стварају са зоном у којој су први пут додељени. Атрибут `zc` створен у једној зони не може се позивати у некој другој зони, односно истоимени атрибут у другој зони биће третиран као потпуно други атрибут везан за ту другу зону. Објекат `mc` је везан за целу мисију. Атрибути додељени овом објекту не зависе од зоне у којој се играч налази. Дакле, атрибут `mc` створен у једној зони моћи ће да се прочита у другој зони, и у опште свим функцијама које су везане за ту конкретну мисију. Најзад, атрибут `gc` важи на нивоу целе кампање. Атрибут `gc` додељен у једној мисији кампање може се позвати у било којој другој мисији кампање. Обично када дизајнер кампање задаје атрибут `gc` у оквиру неке мисије, значи да се припрема за нешто велико, што ће имати далекосежне последице. Нпр. пилот Торч се појављује у мисији А и Д. Ако је настрадао у мисији А, онда се неће појавити у мисији Д, јер је дизајнер кампање поставио атрибут `gc`, рецимо `gc.is_torch_alive`, на `False` у тренутку кад је дотични пилот настрадао у мисији А, а затим је проверавао негде у мисији Д, условом `if gc.is_torch_alive:`

Пример 1 - мисија окршаја која се бави обичном ваздушном борбом

Код:

```
# -*- coding: UTF-8 -*-

from pandac.PandaModules import *

from src.core import *
from src.blocks import *
from src.skirmish import *

_, p_, n_, pn_ = make_tr_calls_skirmish(__file__)

mission_shortdes = p_("mission name", "One on One")

mission_longdes = p_("mission description", """
A duel mission.

Primary objectives:
- Shoot down enemy jet.

""").strip()

mission_difficulty = MISSION_DIFFICULTY.EASY
mission_type = MISSION_TYPE.DOGFIGHT

def init_cache (mc, gc):

    cache_bodies(["mig29", "mig29fd", "f16"])

def mission_start (gc):

    mission = Mission(gc)
    mission.add_init(loadf=init_cache)
    mission.add_zone("zero", clat=34.89, clon=43.36,
                    enterf=zone_zero_enter,
                    exitf=zone_zero_exit,
                    loopf=zone_zero_loop)

    mc = mission.context
    mc.player_fuelfill = 0.8
    mc.player_ammo_cannons = [450]
    mc.player_ammo_launchers = [(None, 3), (R27, 2), (R73, 2), (R60, 2)]
    mc.player_mfd_mode = "targid"

    mission.switch_zone("zero")

    mc.world_day_time = hrmin_to_sec(12, 30)
```

```

return mission

def zone_zero_enter (zc, mc, gc):

    setup_world(zc, mc, gc,
                terraintype="00-iraq",
                skytype="default2",
                stratusdens=0.0,
                cumulusdens=0.0,
                cirrusdens=2.0,
                cloudseed=101,
                playercntl=2)

    zc.player = create_player(mc=mc, world=zc.world,
                              pos=Point3(0, 0, 6000),
                              hpr=Vec3(180, 0, 0),
                              speed=220)

def zone_zero_loop (zc, mc, gc):

    zc.world.chaser = zc.world.player.chaser

    yield zc.world, 1.0

    zc.player.show_message("notification", "left",
                          _("Incoming enemy jet!"), duration=4.0)

    yield zc.world, 2.0

    zc.world.player_control_level = 0
    zc.world.action_music.set_context("attacked")

    yield zc.world, 2.0

    pos = pos_from_horiz(zc.player.ac,
                        Point3(choice([-4000, 2000, 0, 2000, 4000]),
                               choice([-10000, -7000, 7000, 10000]),
                               randrange(4000, 7000)),
                        absz=True)
    hpr = hpr_from_horiz(zc.player.ac, Vec3(choice([0, 180]), 0, 0))
    enemyac = F16(world=zc.world, name="blue", side="merc",
                  texture="models/aircraft/f16/f16_tex.png",
                  fuelfill=0.50,
                  pos=pos,
                  hpr=hpr,
                  speed=200,
                  lnammo=[(None, 6), (Aim9, 2)])
    enemyac.set_ap(target=zc.player.ac)

    while not enemyac.shutdown:
        yield zc.world, 1.0

    yield zc.world, 6.0

```

```

zc.world.action_music.set_context("victory")
zc.player.show_message("notification", "left",
                        _("All enemy jet destroyed."), duration=1.0)
zc.player.show_message("notification", "left",
                        _("Good work."), duration=4.0)

yield zc.world, 5.0

mc.mission.end()

def zone_zero_exit (zc, mc, gc):

    if zc.player and zc.player.alive:
        store_player_state(mc, zc.player)
        yield zc.world, zone_flyout(zc) + 3.0

    zc.world.destroy()

# =====
# Background.

```

Ово је заправо поједностављена верзија окршаја „Стижу“.

Код **сваке** мисије почиње увођењем свих потребних фајлова и њиховог садржаја:

```

# -*- coding: UTF-8 -*-

from pandac.PandaModules import *

from src.core import *
from src.blocks import *
from src.skirmish import *

```

Наредна целина је делимично специфична:

```

_, p_, n_, pn_ = make_tr_calls_skirmish(__file__)

mission_shortdes = p_("mission name", "One on One")

mission_longdes = p_("mission description", """
A duel mission.

Primary objectives:
- Shoot down enemy jet.

""").strip()

mission_difficulty = MISSION_DIFFICULTY.EASY

mission_type = MISSION_TYPE.DOGFIGHT

```

Овај код је специфичан за мисије окршаја. `mission_shortdes` задаје име мисије које ће се појавити у менију окршаја приликом избора мисије. У примеру, мисија се зове „One

on One". У наредном реду, `mission_longdes`, могуће је уписати детаљнији опис мисије, који ће бити излистан такође у менију окршаја приликом одабира ове мисије. У следећем реду кода битан је само атрибут објекта `MISSION_DIFFICULTY`. Могући нивои тежине су `.EASY`, `.HARD` и `.EXTREME`. У зависности од вредности `mission_difficulty`, биће исписана процењена тежина мисије у једној од колона мисије. Најзад, у последњем реду ове целине, слично као код нивоа тежине, преко атрибута могуће је задати тип мисије који ће бити такође излистан у једној од колона мисије, у менију окршаја. Могући типови мисије за сада су `.DOGFIGHT` и `.ATTACK`.

Сад долазимо до првог главног елемента сваке мисије:

```
def init_cache (mc, gc):

    cache_bodies(["mig29", "mig29fd", "f16"])

def mission_start (gc):

    mission = Mission(gc)
    mission.add_init(loadf=init_cache)
    mission.add_zone("zero", clat=34.89, clon=43.36,
                    enterf=zone_zero_enter,
                    exitf=zone_zero_exit,
                    loopf=zone_zero_loop)

    mc = mission.context
    mc.player_fuelfill = 0.8
    mc.player_ammo_cannons = [450]
    mc.player_ammo_launchers = [(None, 3), (R27, 2), (R73, 2), (R60, 2)]
    mc.player_mfd_mode = "targid"

    mission.switch_zone("zero")

    mc.world_day_time = hrmin_to_sec(12, 30)

    return mission
```

Ово је део мисије који називамо заглављем мисије. Најпре треба дефинисати функцију за кеширање:

```
def init_cache (mc, gc):

    cache_bodies([
        "mig29", "mig29fd", "f16", "f15", "btr80", "oil_platform",
        "skyscraper_1", "gunboat_1"])
```

У овој функцији треба навести све 3Д моделе који ће се појавити у мисији, као подстрингове њихових имена фајлова. Кеширање није неопходан корак, али је препоручљиво направити кеш свих 3Д модела објеката у мисији како би они унапред били учитани, те да моменат њиховог стварања током мисије не би на кратко замрзао игру. Што је 3Д модел објекта комплекснији, има више полигона и богатије текстуре, то су шансе веће да игра озбиљно штучне током њиховог првог учењавања. Уколико се модел кешира унапред, не би требало да има било каквог застоја током првог стварања објекта са дотичним моделом.

Наредна функција **увек** почиње са:

```
def mission_start (gc):
```

Ово име функције је резервисана реч коју кôд игре специјално препознаје. Њен једини параметар је објекат gc. Унутар функције, опет **увек** на почетку, ставља се:

```
mission = Mission(gc)
```

а потом је уобичајено да се покрене функција кеширања, ако је претходно направљена:

```
mission.add_init(loadf=init_cache)
```

У наредном блоку се додају зоне:

```
mission.add_zone("zero", clat=34.89, clon=43.36,  
                enterf=zone_zero_enter,  
                exitf=zone_zero_exit,  
                loopf=zone_zero_loop)
```

У зависности колико има зона у мисији, толико пута треба позвати mission.add_zone. Пошто у нашем примеру имамо само једну зону у мисији, онда се и ова функција позива једанпут. Унутар те функције се најпре дефинише име зоне као стринг (у нашем случају "zero"), затим географска ширина и дужина, па имена ниже дефинисаних трију функција зоне, улазне функције, функције петље и излазне функције (овде су то zone_zero_enter, zone_zero_loop и zone_zero_exit).

Потом прелазимо на mc = mission.context, што се ради скраћења наредних израза. Затим следи подешавање играчевог авиона, преко низа резервисаних атрибута mc:

- mc.player_fuelfill, укупна количина горива. Број између 0.0 и 1.0, што означава између 0% и 100% горива. Могуће је ставити и број преко 1.0, и тада се играчевом авиону аутоматски додају танкови са додатним горивом (што заузима најмање једну подвесну тачку). У том случају број може ићи од 1.0 до 2.0 (тј. до 200% горива).
- mc.player_ammo_cannons, количина топовске муниције. Цео број.
- mc.player_ammo_launchers, додавање разног оружја и опреме на подвесне тачке којима авион располаже. Ако авион има рецимо 6 подвесних тачака, а дизајнер жели да монтира само две ракете које стоје ближе врховима крила, то може да постигне тако што напише mc.player_ammo_launchers = [(None, 4), (R60, 2)]. Овим ће прве четири подвесне тачке (две лево, две десно) бити прескочене, и две ракете Р-60 биће монтиране на последње две тачке (једна лево, друга десно). Укупан број подвесних тачака, и на којим су положајима, може се наћи у фајлу src/blocks/planes.py, за сваки авион понаособ. Оружје и опрема доступна за монтирање налази се у фајлу src/blocks/weapons.py.
- mc.player_mfd_mode, задаје у ком моду ТВ екрана ће играч започети мисију. Ради се о стрингу, а могућности за сада су "targid" и "overmap".

Наредни исказ:

```
mission.switch_zone("zero")
```

каже у којој ће зони почети мисија. Треба навести једно од задатих имена зона. Како имамо само једну зону, овај исказ може и да се изостави. Ако се изостави, зона се аутоматски бира као прва по редоследу дефинисања.

Потом треба наштеловати време, тј. у колико сати почиње мисија:

```
mc.world_day_time = hrmin_to_sec(12, 30)
```

што је у овом случају пола сата после подне.

На крају функције треба ставити `return mission`, и то би било све у вези са функцијом заглавља мисије.

Сад прелазимо на функције зоне. Улазна функција зоне изгледа овако:

```
def zone_zero_enter (zc, mc, gc):  
  
    setup_world(zc, mc, gc,  
                terraintype="00-iraq",  
                skytype="default2",  
                stratusdens=0.0,  
                cumulusdens=0.0,  
                cirrusdens=2.0,  
                cloudseed=101,  
                playercntl=2)  
  
    zc.player = create_player(mc=mc, world=zc.world,  
                              pos=Point3(0, 0, 6000),  
                              hpr=Vec3(180, 0, 0),  
                              speed=220)
```

Свака функција зоне се **увек** дефинише са три параметра, за три стандардна објекта `zc`, `mc` и `gc`:

```
def zone_zero_enter (zc, mc, gc):
```

где име функције зоне **мора** да одговара имену задатом у једном од `mission.add_zone(...)` из заглавља. Име које смо ми поставили за улазну функцију зоне у заглављу мисије је `zone_zero_enter`. У улазној функцији, обичај је да се најпре направи свет, тј. терен и небо. Функција `setup_world` служи за то. Та функција поседује низ параметара, а ми ћемо се сконцентрисати само на оне испостављене у позиву нашег примера. За цео списак параметара видети ту функцију у фајлу `src/blocks/missiontools.py`. `zc`, `mc`, `gc` су прва три стандардна аргумента. Затим следи избор терена. Треба уписати име жељеног терена, а списак унапред спремљених терена налази се у фајлу `src/blocks/terrains.py`. Затим, треба изабрати врсту неба. Слично као код терена, треба уписати име жељеног неба, а списак свих унапред спремљених типова неба такође се налази у фајлу `src/blocks/terrains.py`. Потом следи избор густине три типа облака: цируса, кумулуса и стартуса (у зависности од терена до терена, нису увек доступна сва три типа облака). Густина било ког типа облака, ако се зада као нула, значи да тог типа облака нема. `cloudseed`, одређује низ случајних бројева на основу којих ће се распоредити облаци према мапи облака. Последњи аргумент, `playercntl`, тиче се тога у ком моду контроле ће играч започети зону (могући уноси су 0, 1 и 2). О овоме ће бити више речи мало касније.

Пошто нам контекст радње ове једноставне мисије то дозвољава, у улазној функцији одмах креирамо и играча, и притом га додељујемо атрибуту `zc.player` да би смо после могли да манипулишемо играчем преко тог атрибута. Функција `create_player` такође поседује низ параметара, од којих ћемо се фокусирати само на испостављене у нашем примеру. Прва два аргумента су попуњена увек на начин који се може видети из приложеног кода, док наредна три дизајнер треба да попуни већ према својој замисли. Та три аргумента су:

- pos, место на коме ће се играчев авион створити у свету. У питању је вектор тј. 3Д координата. Централна 3Д координате света је увек (0, 0, 0). Мање-више установљен стандард за величину терен је 320000x320000 m², а радијус видљивости (докле се види хоризонт) је 80000 m, тако да је величина арене унутар целог терена 320000 минус 2 * 80000, односно квадрат величине 160000x160000 m². То значи да играчев авион може да буде створен између +80000 и -80000 m по x-координати света, исто толико по у-координати, а између надморске висине (0) и неке разумне висине (нпр. 15000 m) по z-координати. Ми смо играчев авион створили у центру хоризонталне равни терена на висини од 6000 метара.
- hpr, оријентација играчевог авиона. Три основне компоненте су правац, нагиб и окрет. У начелу, најважнија је прва компонента, правац. 0 је север, 180 је југ, 90 је запад, 270 је исток. Овде смо играчев авион створили окренутог тачно ка југу.
- speed, брзина којом ће играчев авион летети кад се створи.

То би било све што се тиче улазне функције зоне у нашем примеру. Сад следи функција петље зоне, односно функција у којој је смештена главнина контекста радње у зони:

```
def zone_zero_loop (zc, mc, gc):
    zc.world.chaser = zc.world.player.chaser
    yield zc.world, 1.0
    zc.player.show_message("notification", "left",
                           _("Incoming enemy jet!"), duration=4.0)
    yield zc.world, 2.0
    zc.world.player_control_level = 0
    zc.world.action_music.set_context("attacked")
    yield zc.world, 2.0
    pos = pos_from_horiz(zc.player.ac,
                        Point3(choice([-4000, 2000, 0, 2000, 4000]),
                               choice([-10000, -7000, 7000, 10000]),
                               randrange(4000, 7000)),
                        absz=True)
    hpr = hpr_from_horiz(zc.player.ac, Vec3(choice([0, 180])), 0, 0)
    enemyac = F16(world=zc.world, name="blue", side="merc",
                  texture="models/aircraft/f16/f16_tex.png",
                  fuelfill=0.50,
                  pos=pos,
                  hpr=hpr,
                  speed=200,
                  lnammo=[(None, 6), (Aim9, 2)],
                  skill="rookie")
    enemyac.set_ap(target=zc.player.ac)
    while not enemyac.shutdown:
        yield zc.world, 1.0
    yield zc.world, 6.0
```

```
zc.world.action_music.set_context("victory")
zc.player.show_message("notification", "left",
    _("Hostile jet destroyed."), duration=1.0)
zc.player.show_message("notification", "left",
    _("Good work."), duration=4.0)

yield zc.world, 5.0

mc.mission.end()
```

Као и код улазне функције зоне, у дефинисању функције петље присутна су три стандардна параметра за три стандардна објекта. Оно што се одмах може приметити јесте да у функцији користимо исказ `yield zc.world`, што значи да ће ово бити проточни тип функције петље. По завршетку извршавања улазне функције зоне, одмах отпочиње да се извршава функција петље зоне, и зато што је проточна, извршити ће се само једанпут од почетка до краја. Међутим, у нашем примеру је контекст радње тако скројен да се функција неће брзо одвртети, јер постоје многе препреке на путу до њеног краја.

Први ред унутар функције:

```
zc.world.chaser = zc.world.player.chaser
```

бира камеру, односно кадар из кога ће се гледати у свет. `zc.world.chaser` је резервисан атрибут на који се монтира камера, и аутоматски се кадар погледа у свет пребацује на ту камеру. То међутим ради тако само ако је дизајнер претходно одузео контролу играчу, и тиме контекст тренутне радње убацио у анимациони. Уколико играч поседује контролу, ефекат пребацивања кадра неће бити видљив. Кадар ће се, иза завесе, пребацити на постављену камеру, али се слика на екрану неће пребацити на тај кадар све док дизајнер не пребаци на анимациони контекст радње. Чим дизајнер то учини, аутоматски ће у употреби бити она камера која је последња била додељена на `zc.world.chaser`. Како се даје и одузима контрола играчу, тј. пребацује на анимациони контекст радње, биће објашњено ускоро. Вреди још поменути да је `zc.world.player.chaser` једна од ретких унапред дефнисаних камера, и представља поглед из првог лица играча. За квалитетну режију акције у анимацији, највећи део времена потребно је да дизајнер креира своје специфичне камере и онда да шета између њих. Како се креира нова специфична камера за специфичан кадар, то је компликован посао који ће бити објашњен у другом примеру (који ће се, да подсетимо, бавити компликованијом верзијом једне мисије).

Наредни ред `yield zc.world, 1.0` је обична кратка пауза. Значи да ће функција чекати једну секунду пре него што настави даље извршавање. Написана је зато што је дизајнер проценио да је у току радње потребна кратка пауза, из пре свега естетских разлога.

Затим следи:

```
zc.player.show_message("notification", "left",
    _("write_whatever_you_want"), duration=4.0)
```

Ово је исказ која служи да се шаљу обавештења играчу током мисије. Отвара прозор са обавештењима у доњем левом квадранту екрана, унутар кога ће бити исписано све што је дизајнер мисије написао као стринг у горњем реду. Може се задати и колико дуго ће порука трајати пре него што се уклони, а у нашем примеру то је 4 секунде.

Потом следи још једна пауза у извршавању функције петље, овог пута од две секунде, поново из естетских разлога, а онда долазимо до баратања контролом коју играч има:

```
zc.world.player_control_level = 0
```

`zc.world.player_control_level` је још један резервисани атрибут. Могу му бити додељене три вредности:

- 0 - играч има потпуну контролу над игром
- 1 - играчу је одузето контрола, али је камера и даље строго везана за кабину
- 2 - играчу је одузето контрола, и контекст је пребачен на контекст анимације

Чим се активира контекст анимације, вредност 2, појављују се анимационе пречке у горњем и доњем делу екрана, а кадар се пребацује на последњу камера која је била додељена на `zc.world.chaser`. У том моду, дизајнер мисије има сву слободу да лично изрежира целу сцену (уз помоћ додатног алата којим ћемо се бавити у другом примеру). Када је замишљена сцена завршена (а притом није и крај мисије), дизајнер би требало да врати контролу играчу, тако што ће поново доделити атрибуту `zc.world.player_control_level` вредност 0. У нашем примеру, раније је унутар функције `setup_world` (у улазној функцији зоне) вредност овог атрибута је била постављена на 2, преко параметра `playercntl`. То смо урадили јер је контекст радње у зони почињао као анимација, а сад смо играчу вратили контролу. Параметар `playercntl` није никакав специјални параметар, већ даје исти ефекат као и атрибут `zc.world.player_control_level`. Нама је било згодно да га испољимо у функцији `setup_world`, управо за зоне које почињу анимацијом, чисто да дизајнер мисије не би морао да пише `zc.world.player_control_level = 2` одмах испод позива функције `setup_world`.

Следећи ред се бави пуштањем позадинске музике:

```
zc.world.action_music.set_context("attacked")
```

Стрингови унутар функције `.set_context()` су резервисане речи. По позиву ове функције тренутна позадинска музика ће бити аутомаски угашена, а позвана упаљена.

- "cruising", мирнодопска музика. Понављајућа, позадинска музика када играч безбрижно лети.
- "attacked", нападачка музика. Понављајућа, позадинска музика која се пушта када су играч или неки његов савезник под нападом, или када играч сам припуца на некога.
- "boss", специфична нападачка музика. Понављајућа, позадинска музика која је намењена ситуацијама када је играч под нападом неког опаког или важног противника (нпр. неког аса или неког „капиталног“ возила).
- "victory", победничка кратка свирка. Кратка музика које се одврти само једанпут, и обично се пушта када је мисија успешно обављена.
- "defeat", губитничка кратка свирка. Кратка музика која се одврти само једанпут, и то обично када је мисија пропала.
- "silence", нема никакве музике.

Ово је тренутна листа, у перспективи требало би да буде још врста музика, типа мирнодопска-пријатељска, мирнодопска-претећа, кад играч настрада, итд.

Треба још напоменути да је по читавању сваке зоне на снази аутомаски режим управљања музиком. То значи да ће се, по читавању зоне, аутоматски покренути мирнодопска музика, а потом кад је играч под нападом, аутоматски покренути

нападачка. Када се напад заврши, аутоматски се гаси нападачка и пушта поново мирнодопска (услов да би се прогласио крај напада је да нико никог не циља више од 10 секунди). Уколико дизајнер било где унутар функције зоне ручно пребаци на неку музику, као што смо ми урадили у нашем примеру, онда се цео аутоматски механизам управљања музиком суспендује, и од тада дизајнер мора ручно да барата музиком у тој зони.

У наставку извршавања наше функције петље зоне следи још једна драмска пауза од две секунде, а потом креирамо противника:

```
pos = pos_from_horiz(zc.player.ac,
                    Point3(choice([-4000, 2000, 0, 2000, 4000]),
                           choice([-10000, -7000, 7000, 10000]),
                           randrange(4000, 7000)),
                    absz=True)
hpr = hpr_from_horiz(zc.player.ac, Vec3(choice([0, 180]), 0, 0))
enemyac = F16(world=zc.world, name="blue", side="merc",
             texture="models/aircraft/f16/f16_tex.png",
             fuelfill=0.50,
             pos=pos,
             hpr=hpr,
             speed=200,
             lnammo=[(None, 6), (Aim9, 2)],
             skill="rookie")
enemyac.set_ap(target=zc.player.ac)
```

Противнички авион, у нашем случају један Ф-16, креира се увек на овај начин. F16 је име класе направљене за тај авион. Сви доступни авиони односно, све класе направљене за разне авионе, налазе се у фајлу `src/blocks/planes.py`.

- `world`, параметар који увек мора да буде попуњен на начин као у нашем примеру.
- `name`, име за дотични авион. Углавном не служи ничему, осим за дебаговање и разне друге конзолне исписе.
- `side`, страна којој авион припада. Подразумевано, два авиона којима је уписан различит стринг за овај аргумент ће аутоматски један другог сматрати непријатељом. Постоји начин да се каже да су две различите стране савезници, а о томе ће бити више речи у другом примеру.
- `texture`, служи да се зада нека од текстура на модел авиона. Које су све текстуре доступне, видети у директоријуму модела за дати авион.
- `fuelfill`, количина горива. Од 0.0 до 1.0 (што значи од 0% до 100% попуњености).
- `pos`, место и висина у свету где ће се авион створити. У нашем примеру, позиција је нешто компликованије одређена него како смо били написали код стварања играча:

```
pos = pos_from_horiz(zc.player.ac,
                    Point3(choice([-4000, 2000, 0, 2000, 4000]),
                           choice([-10000, -7000, 7000, 10000]),
                           randrange(4000, 7000)),
                    absz=True)
```

`pos_from_horiz` је функција која каже да се нешто створи на позицији релативној у односу на други објекат, и то равно према хоризонту. У нашем случају ми смо рекли да се противнички авион створи у односу на играчев авион, `zc.player.ac`. Притом смо за координате те позиције задали насумичне изборе, како би се авион увек створио на неком другом месту (било по *x*-, *y*- или *z*-координати), у односу на играчев авион. На

крају, `absz=True` значи да желимо да се висина посматра апсолутно у односу на свет, а не у односу на играча, јер је тако често природније за контекст радње. Подразумевана вредност параметра `absz` је `False`. * `hpr`, оријентација створеног авиона. Као и код позиције, у нашем примеру оријентација је нешто компликованије одређена него како смо били написали код стварања играча:

```
hpr = hpr_from_horiz(zc.player.ac, Vec3(choice([0, 180]), 0, 0))
```

Прича је веома слична као и у случају позиције. 0 за елемент вектора који се тиче правца значи да ће нос овог авиона бити окренут у истом правцу као и играчев нос, а 180 у потпуно супротном правцу. Дакле, ако се противнички авион налази испред играча, а његов угао правца је 180 степени, онда ће играч и његов противник јуришати директно један на другог, као у игри кукавице. * `speed`, брзина којом ће авион летети чим се створи. * `lnammo`, качење опреме на подвесне тачке авиона. Функционише на исти начин као и претходно описани `mc.player.ammo_launchers`. * `skill`, ниво вештине коју пилот дотичног авиона поседује. Тренутно овај аргумент не ради добро и треба га изоставити. У перспективи ће постојати пет нивоа тежине, односно пет вредности: "rookie", "pilot", "veteran", "ace" и "custom".

То би било све што се тиче начина на који се позива нови авион унутар зоне, а одмах следи и задавање понашања тог авиона:

```
enemyac.set_ap(target=zc.player.ac)
```

`.set_ap()` је функција аутопилота. То је веома важна функција, и њоме се одређује не само понашање авиона, већ и добар део вештачке интелигенције авиона. Према задатим аргуменатима, авион ће покушати да изврши наредбу најбоље што уме. У нашем случају наредба је врло проста, `target=zc.player.ac`, што значи да се нападне играч. Ова функција поседује низ параметара:

- `altitude`, надморска висина коју авион треба да достигне.
- `speed`, брзина коју треба да достигне.
- `climbrate`, брзина којом треба да се пење.
- `turnrate`, брзина којом треба да скреће.
- `heading`, правац којим треба да иде.
- `point`, тачка коју треба да досегне.
- `otraltitude`, висина над тлом коју треба да одржава.
- `leader`, задаје неки други објект авиона као вођу формације.
- `formpos`, позиција у формацији.
- `target`, мета коју треба напасти.
- `useab`, вредност `True/False` која каже сме ли авион да користи допунско сагоревање.
- `maxg`, максимално g-оптерећење које сме да се достигне.
- `invert`, да ли летети на леђима (ако је могуће остале параметре остварити и летећи усправно и летећи на леђима).
- `enroute`, вредност `True/False` која каже да ли да се настави лет по дефинисаној маршрути.

Списак свих ових параметара ће вероватно бити прошириван у будућности.

Након стварања и задавања понашања противничком авиону, спуштамо рампу на даље напредовање функције петље зоне:

```
while not enemyac.shutdown:
    yield zc.world, 1.0
```

Овај блок просто значи: док противнички авион није оборен, нема даљег напредовања функције петље. Притом, унутар ове петље може се ставити и само `yield` тј. да проверва услов петље у сваком фрејму. Ми смо ставили `yield zc.world, 1.0`, што значи да проверава тек у свакој секунди времена света, да се не би беспотребно трошиле перформансе. Сад је на играчу да обори противнички авион и омогући наставак извршавања функције петље, а то може да потраје.

Када играч коначно успе да обори противнички авион, рампа се диже, и функција петље наставља даље. Најпре следи још једна драмска пауза, овог пута од шест секунди, а потом:

```
zc.world.action_music.set_context("victory")
```

окидање победничке кратке свирке и:

```
zc.player.show_message("notification", "left",
                       _("Hostile jet destroyed."), duration=1.0)
zc.player.show_message("notification", "left",
                       _("Good work."), duration=4.0)
```

обавештење о позитивном исходу. Овог пута се ради о два обавештења, првом које траје једну секунду и другом које траје четири секунде. То што прво траје једну секунду не значи да ће се уклонити после једне секунде, већ само да ће се друго обавештење појавити скоро истовремено у истом прозору са обавештењима. А тај прозор ће укупно трајати $4 + 1$ секунду, тј. 5 секунди, када ће се уклонити и прво и друго обавештење истовремено.

Потом следи још једна драмска пауза од пет секунди, и на крају:

```
mc.mission.end()
```

За разлику од улазне функције зоне, која чим се изврши аутоматски почиње да се извршава функција петље зоне, код функције петље то није случај. Наиме, радња зоне ће трајно остати задржана унутар функције петље, све док се не изда експлицитна наредба да се пређе на излазну функцију зоне. Та наредба је горе наведени ред.

Закључно са последњим редом унутар функције петље зоне, почиње се извршавати излазна функција зоне:

```
def zone_zero_exit (zc, mc, gc):

    if zc.player and zc.player.alive:
        store_player_state(mc, zc.player)
        yield zc.world, zone_flyout(zc) + 3.0

    zc.world.destroy()
```

Функција се наравно дефинише опет са три стандардна параметра за три стандардна објекта, а потом почиње провером стање играча:

```
if zc.player and zc.player.alive:
```

да ли је присутан и да ли је читав. Ако јесте позива се функција `store_player_state`, која треба да запамти опште стање играча (ниво оштећења, количину горива, количину муниције, број преосталих ракета, итд). Потом следи функција (ефикасно написана, али за почетнике непрегледно углављена) анимације одлетања играча. Та анимација је смештена унутар функције `zone_flyout`, чији се код може видети у фајлу `src/blocks/missiontools.py`.

На крају следи уклањање света унутар зоне:

```
zc.world.destroy()
```

а то у случају нашег примера такође значи и крај мисије и крај целине која се тиче скриптовања једне зоне.

За крај треба поменути и још две функције, а које се користе у нашем примеру посредно, унутар функције `zone_flyout`. То су:

```
zc.world.fade_in(time=1.0)
zc.world.fade_out(time=1.0)
```

Ове две функције служе да слика на екрану урони у црnilo или изрони из црnila. Њихова сврха је пре свега естетска, и служи за глатку режију сцена. Да се ублаже прелази, јер је незграпно да кад се рецимо уклони свет, екран одједном зацрни, уместо да постепено урони у црnilo.

То би било то што се тиче првог примера. Једноставно, зар не?

Пример 2 - комплексна мисија кампање

Пример бр. 2 претпоставља да је читалац добро проучио и разумео пример бр. 1, тако да се нећемо више бавити објашњавањем основне грађе у изради једне мисије.

Целокупан код мисије:

```
# -*- coding: UTF-8 -*-

from pandac.PandaModules import *

from src.core import *
from src.blocks import *
from __init__ import *

def airfield_2000x30 (zc, mc, gc, pos, hpr, side):

    runway_pos = pos
    runway_hpr = hpr
    runway = CustomBuilding(
        world=zc.world, name="af2000x30_runway", side=side,
        strength=6000, minhitdmg=3000, maxhitdmg=5000, rcs=0,
        hitboxdata=[],
        modelpath="models/buildings/runway/runway2000x30.egg",
        texture="models/buildings/runway/runway2000x30_tex.png",
        normalmap="models/buildings/runway/runway2000x30_nm.png",
```

```

        glossmap="models/buildings/runway/runway2000x30_gls.png",
        clamp=False,
        pos=pos,
        hpr=hpr,
        damage=0,
        castshadow=False)
runway_surface = VirtualHorizPoly(
    poly=[pos_from_point(runway_pos, runway_hpr,
                          Point2( 15.0, 1000.0)).getXy(),
          pos_from_point(runway_pos, runway_hpr,
                          Point2( 15.0, -1000.0)).getXy(),
          pos_from_point(runway_pos, runway_hpr,
                          Point2(-15.0, -1000.0)).getXy(),
          pos_from_point(runway_pos, runway_hpr,
                          Point2(-15.0, 1000.0)).getXy()],

    convex=True,
    flush=True,
    elev=(zc.world.elevation(runway_pos) + 0.1),
    gtype=GROUND.RUNWAY)
zc.world.terrains[0].add_virtual_surface(runway_surface)
runway.player_ground_pos = pos_from_point(runway_pos, runway_hpr,
                                           Point3(0.0, -900.0, 0.0))

runway.player_ground_hpr = runway_hpr
return runway

```

```
def init_cache (mc, gc):
```

```

    cache_bodies(["f16", "f18", "mig29", "bradley", "warehouse_1",
                  "runway2000x30"])

```

```
def mission_start (gc):
```

```
    mission = Mission(gc)
```

```
    mission.add_init(loadf=init_cache)
```

```

    mission.add_zone("canary", clat=28.22, clon=-16.04,
                    enterf=cr060_canary_enter, exitf=cr060_canary_exit,
                    loopf=cr060_canary_loop)

```

```

    mission.add_zone("wsahara", clat=24.56, clon=-13.68,
                    enterf=cr060_wsahara_enter, exitf=cr060_wsahara_exit,
                    loopf=cr060_wsahara_loop)

```

```
    mission.switch_zone_pause = 1.0
```

```
    mc = mission.context
```

```
    mc.player_fuelfill = 1.0
```

```
    mc.player_ammo_cannons = [500]
```

```
    mc.player_ammo_launchers = [(None, 4), (Aim9, 4)]
```

```
    mc.player_mfd_mode = "overmap"
```

```
    mission.switch_zone("canary")
```

```
# Alliances
```



```

mc.alliances = [("lw", "dwater")]

# Mission start time
mc.world_day_time = hrmin_to_sec(9, 21)

# Set mission objectives.
mc.objective_patrol_x = False

return mission

# =====
# Canary islands zone

def cr060_canary_enter (zc, mc, gc):

# Create world.
setup_world(zc, mc, gc,
            terraintype="15-canary",
            skytype="default2",
            stratusdens=0.8,
            cumulusdens=0.9,
            cirrusdens=2.0,
            cloudseed=1506,
            playercntl=0,
            alliances=mc.alliances)

# Add base complex, runway, etc.
zc.afrunway = airfield_2000x30(zc, mc, gc, side="lw",
                              pos=Point2(9100, -4200),
                              hpr=Point3(0,0,0))

# Add player and wingmen if any.
# Depending if this is the first or a subsequent entry into the zone,
# set player on the runway or in the air.
if not zc.visited_before:
    zc.player = create_player(
        mc=mc, world=zc.world,
        pos=zc.afrunway.player_ground_pos,
        hpr=zc.afrunway.player_ground_hpr,
        speed=0, onground=True,
        texture="models/aircraft/f16/f16dutch_tex.png")
    zc.hulk = F16(
        world=zc.world, name="hulk", side="lw",
        texture="models/aircraft/f16/f16dutch_tex.png",
        fuelfill=1.0,
        pos=pos_from_horiz(zc.player.ac, Point3(+20, 15, 0)),
        hpr=hpr_from_horiz(zc.player.ac, Vec3(0, 0, 0)),
        speed=0, onground=True,
        lnammo=[(None, 6), (Aim9, 2)],
        skill="veteran")
    zc.draggon = F16(
        world=zc.world, name="draggon", side="lw",
        texture="models/aircraft/f16/f16dutch_tex.png",
        fuelfill=1.0,
        pos=pos_from_horiz(zc.player.ac, Point3(-20, 20, 0)),

```

```

        hpr=hpr_from_horiz(zc.player.ac, Vec3(0, 0, 0)),
        speed=0, onground=True,
        lnammo=[(None, 6), (Aim9, 2)],
        skill="pilot")
    zc.draggon.set_auto_attack(["plane"])
    zc.hulk.set_auto_attack(["plane"])
    formation_triplet(zc.player.ac, zc.draggon, zc.hulk,
                     compact=1.0, jumpto=False)
else:
    zc.player = create_player(
        mc=mc, world=zc.world,
        pos=pos_from_horiz(zc.afrunway, Point3(0, -12137, 1927)),
        hpr=hpr_from_horiz(zc.afrunway, Vec3(0, 0, 0)),
        speed=200,
        texture="models/aircraft/f16/f16dutch_tex.png")

# Add objects.
pass

# Set navpoints/waypoints.
zc.player.add_navpoint(name="nav1",
                       longdes=_("Navpoint 1"), shortdes=_("NAV1"),
                       pos=Point2(0, 0), radius=640000, height=-1,
                       tozone="wsahara")
add_base_waypoint(zc, mc, base=zc.afrunway, name="base",
                  longdes=_("land"), shortdes=_("BASE"))
if zc.visited_before:
    zc.player.update_navpoint("nav1", active=False)

def cr060_canary_loop (zc, mc, gc):

    if (not mc.mission_failed and not zc.visited_before and
        (zc.hulk.shutdown or zc.draggon.shutdown)):
        zc.player.update_navpoint("nav1", active=False)
        mission_failed(zc, mc, gc,
                       reason=_("You shot down your own, pilot.))

    if (not zc.nail_player and mc.mission_failed and
        zc.world.stopwatch("countdown_nailplayer") > 4):
        zc.world.explosion_damage(force=1000, ref=zc.player.ac.pos())
        zc.nail_player = True

    return zc.world, 1.0

def cr060_canary_exit (zc, mc, gc):

    zc.visited_before = True

    if zc.hulk and not zc.hulk.outofbattle:
        zc.hulk.jump_to(
            pos=pos_from_horiz(zc.player.ac, Point3(50, 100, -10)),
            hpr=hpr_from_horiz(zc.player.ac, Vec3(0, 0, 0)),
            speed=zc.player.ac.speed())
        store_plane_state(mc, zc.hulk)

```

```

    mc.hulk_alive = True
else:
    mc.hulk_alive = False

if zc.draggon and not zc.draggon.outofbattle:
    zc.draggon.jump_to(
        pos=pos_from_horiz(zc.player.ac, Point3(-50, 100, 10)),
        hpr=hpr_from_horiz(zc.player.ac, Vec3(0, 0, 0)),
        speed=zc.player.ac.speed())
    store_plane_state(mc, zc.draggon)
    mc.draggon_alive = True
else:
    mc.draggon_alive = False

if zc.player and zc.player.alive:
    store_player_state(mc, zc.player)
    yield zc.world, zone_flyout(zc)

zc.world.destroy()

# =====
# Western Sahara zone

def cr060_wsahara_enter (zc, mc, gc):

    # Create world.
    setup_world(zc, mc, gc,
        terraintype="16-wsahara",
        skytype="default2",
        stratusdens=0.4,
        cumulusdens=1.1,
        cirrusdens=1.8,
        cloudseed=1606,
        playercntl=0,
        alliances=mc.alliances)

    # Add base complex, runway, etc.
    zc.patlpos = Point2(-40002, 18915)
    zc.patxpos = Point2(-21495, -58290)
    zc.basewarehouse1 = Warehouse1(
        world=zc.world, name="barrack_1", side="dwater",
        texture="models/buildings/warehouse/warehouse_1_tex.png",
        normalmap="models/buildings/warehouse/warehouse_1_nm.png",
        pos=(zc.patxpos + Point2(0, 50)),
        hpr=Vec3(270, 0, 0))
    zc.basewarehouse2 = Warehouse1(
        world=zc.world, name="barrack_1", side="dwater",
        texture="models/buildings/warehouse/warehouse_1_tex.png",
        normalmap="models/buildings/warehouse/warehouse_1_nm.png",
        pos=(zc.patxpos + Point2(-25, -50)),
        hpr=Vec3(270, 0, 0))

    # Add player and wingmen if any.
    # Depending if this is the first or a subsequent entry into
    # the zone, set player on the runway or in the air.

```

```

zc.player = create_player(
    mc=mc, world=zc.world,
    pos=Point3(-41212, 18009, 7452),
    hpr=Vec3(215, 0, 0),
    speed=210,
    texture="models/aircraft/f16/f16dutch_tex.png")
if mc.hulk_alive and mc.draggon_alive:
    zc.hulk = recreate_plane(
        mc, world=zc.world, name="hulk",
        texture="models/aircraft/f16/f16dutch_tex.png",
        pos=pos_from_horiz(zc.player.ac, Point3(-200, 200, 35)),
        hpr=hpr_from_horiz(zc.player.ac, Vec3(0, 0, 0)),
        speed=210)
    zc.hulk.set_auto_attack(["plane"])
    zc.draggon = recreate_plane(
        mc, world=zc.world, name="hulk",
        texture="models/aircraft/f16/f16dutch_tex.png",
        pos=pos_from_horiz(zc.player.ac, Point3(-100, 250, 20)),
        hpr=hpr_from_horiz(zc.player.ac, Vec3(0, 0, 0)),
        speed=210)
    zc.draggon.set_auto_attack(["plane"])
    formation_pair(zc.hulk, zc.draggon, compact=0.1, jumpto=True)

# Add objects.
zc.vhc1 = Bradley(
    world=zc.world, name="larmor1", side="dwater",
    texture="models/vehicles/bradley/bradley_tex.png",
    pos=pos_from_horiz(zc.basewarehouse1, Point2(200, 0)),
    hpr=hpr_from_horiz(zc.basewarehouse1, Vec3(0, 0, 0)),
    speed=0.0)
zc.vhc2 = Bradley(
    world=zc.world, name="larmor2", side="dwater",
    texture="models/vehicles/bradley/bradley_tex.png",
    pos=pos_from_horiz(zc.vhc1, Point2(20, 0)),
    hpr=hpr_from_horiz(zc.vhc1, Vec3(0, 0, 0)),
    speed=0.0)
zc.vhc3 = Bradley(
    world=zc.world, name="larmor3", side="dwater",
    texture="models/vehicles/bradley/bradley_tex.png",
    pos=pos_from_horiz(zc.vhc1, Point2(0, -20)),
    hpr=hpr_from_horiz(zc.vhc1, Vec3(0, 0, 0)),
    speed=0.0)
zc.vhc4 = Bradley(
    world=zc.world, name="larmor4", side="dwater",
    texture="models/vehicles/bradley/bradley_tex.png",
    pos=pos_from_horiz(zc.vhc1, Point2(20, -20)),
    hpr=hpr_from_horiz(zc.vhc1, Vec3(0, 0, 0)),
    speed=0.0)
zc.vhc5 = Bradley(
    world=zc.world, name="larmor5", side="dwater",
    texture="models/vehicles/bradley/bradley_tex.png",
    pos=pos_from_horiz(zc.vhc1, Point2(0, -40)),
    hpr=hpr_from_horiz(zc.vhc1, Vec3(0, 0, 0)),
    speed=0.0)
zc.vhc6 = Bradley(
    world=zc.world, name="larmor6", side="dwater",

```

```

    texture="models/vehicles/bradley/bradley_tex.png",
    pos=pos_from_horiz(zc.vhcl, Point2(20, -40)),
    hpr=hpr_from_horiz(zc.vhcl, Vec3(0, 0, 0)),
    speed=0.0)

# Set navpoints/waypoints.
zc.player.add_waypoint(
    name="pat1", longdes=_("patrol point 1"), shortdes=_("PAT1"),
    pos=zc.pat1pos, radius=2000, height=-1)
zc.player.add_waypoint(
    name="patx", longdes=_("patrol point x"), shortdes=_("PATX"),
    pos=zc.patxpos, radius=500, height=500)
zc.player.add_navpoint(
    name="home", longdes=_("Navpoint 1"), shortdes=_("HOME"),
    pos=Point2(0, 0), radius=640000, height=-1,
    tozone="canary")
zc.player.update_navpoint("home", active=False)

def cr060_wsahara_loop (zc, mc, gc):

    if (not zc.first_conv and
        zc.world.stopwatch("countdown_firstconv") > 6):
        if (not zc.first_conv_triggered and mc.hulk_alive and
            mc.draggon_alive):
            zc.world.chaser = zc.world.player.chaser
            zc.world.player_control_level = 2
            zc.player.ac.set_ap(climbrate=0.0, turnrate=0.0, maxg=9.0)
            zc.dialog01 = conv_cr060_01(zc.world, zc.player,
                zc.hulk, zc.draggon)

            zc.dialog01.start()
            zc.first_conv_triggered = True
        elif zc.first_conv_triggered and not zc.dialog01.in_progress():
            zc.world.player_control_level = 0
            zc.player.ac.jump_to(
                pos=Point3(zc.player.ac.pos()[0], zc.player.ac.pos()[1],
                    zc.player.ac.pos()[2]),
                hpr=Vec3(120,0,0), speed=200)
            zc.hulk.jump_to(
                pos=Point3(100000, 100000, 8000),
                hpr=Vec3(270,0,0), speed=190)
            zc.draggon.destroy()
            zc.world.fade_in(0.5)
            zc.first_conv = True

    if not mc.objective_patrol_x and zc.player.at_waypoint("patx"):
        zc.player.show_message("notification", "left",
            _("Objective complete!"), duration=4.0)
        mc.objective_patrol_x = True

    if (not zc.second_conv and mc.objective_patrol_x and
        zc.world.stopwatch("countdown_secondconv") > 3):
        zc.dialog02 = conv_cr060_02(zc.world, zc.player)
        zc.dialog02.start()
        zc.second_conv = True

```

```

if (not zc.third_conv and zc.second_conv and
    not zc.dialog02.in_progress() and
    zc.world.stopwatch("countdown_thirdconv") > 4):
    zc.world.fade_out(0.5)
    zc.world.break_alliance(["lw", "dwater"])
    zc.enemyac5 = F18(
        world=zc.world, name="gray1", side="dwater",
        texture="models/aircraft/f18/f18darkwater_tex.png",
        fuelfill=0.50,
        pos=Point3(500, 520000, 5000),
        hpr=Vec3(180, 0, 0),
        speed=210,
        lnammo=[(None, 4), (Aim9, 2)])
    zc.enemyac6 = F18(
        world=zc.world, name="gray2", side="dwater",
        texture="models/aircraft/f18/f18darkwater_tex.png",
        fuelfill=0.50,
        pos=Point3(-500, 510000, 5000),
        hpr=Vec3(180, 0, 0),
        speed=210,
        lnammo=[(None, 4), (Aim9, 2)])
    zc.chaser_player = TrackChaser(
        world=zc.world, point=Point3(15, 20, 6),
        relto=zc.player.ac, rotrel=True,
        atref=zc.player.ac, upref=zc.player.ac,
        drift=("instlag", 0.0, 0.25),
        shake=("speed-air", 500.0, 2.0))
    zc.chaser_enemyac5 = TrackChaser(
        world=zc.world, point=Point3(-30, 40, -6),
        relto=zc.enemyac5, rotrel=True,
        atref=zc.enemyac5, upref=zc.enemyac5,
        drift=("instlag", 0.0, 0.25),
        shake=("speed-air", 500.0, 2.0))
    zc.dialog03 = conv_cr060_03(zc.world, zc.player, zc.enemyac5,
                                zc.enemyac6, zc.chaser_player,
                                zc.chaser_enemyac5)

    zc.dialog03.start()
    zc.third_conv = True
elif (not zc.begin_fight and zc.third_conv and
     not zc.dialog03.in_progress()):
    zc.world.player_control_level = 0
    zc.player.ac.jump_to(
        pos=Point3(zc.patxpos[0] + 3000, zc.patxpos[1] - 2600, 4005),
        hpr=Vec3(65,0,0), speed=220)
    zc.enemyac5.jump_to(
        pos=pos_from_horiz(zc.player.ac, Point3(4000, 12700, 1453)),
        hpr=Vec3(180,0,0), speed=240)
    zc.enemyac6.jump_to(
        pos=pos_from_horiz(zc.player.ac, Point3(-3000, 14000, 2019)),
        hpr=Vec3(180,0,0), speed=240)
    zc.enemyac5.set_ap(target=zc.player.ac)
    zc.enemyac6.set_ap(target=zc.player.ac)
    zc.chaser_player.destroy()
    zc.chaser_enemyac5.destroy()
    zc.world.fade_in(0.5)
    zc.begin_fight = True

```

```

if (not zc.help_arrived and zc.begin_fight and
    zc.world.stopwatch("countdown_help") > 16):
    formation_pair (zc.player.ac, zc.hulk, compact=6.0, jumpto=True)
    zc.hulk.set_min_fuelfill(0.5)
    zc.hulk.set_auto_attack(["plane"])
    zc.dialog04 = conv_cr060_04(zc.world, zc.player, zc.hulk)
    zc.dialog04.start()
    zc.help_arrived = True

if (not zc.fifth_conv and zc.help_arrived and
    zc.enemyac5.outofbattle and zc.enemyac6.outofbattle):
    zc.world.action_music.set_context("cruising")
    if not zc.hulk.outofbattle:
        zc.dialog05 = conv_cr060_05(zc.world, zc.player, zc.hulk)
        zc.dialog05.start()
    zc.fifth_conv = True

if (not mc.mission_completed and mc.objective_patrol_x and
    zc.fifth_conv and zc.world.stopwatch("countdown_complete") > 8):
    zc.player.update_navpoint("home", active=True)
    mission_completed(zc, mc, gc)

return zc.world, 1.0

def cr060_wsahara_exit (zc, mc, gc):

    zc.visited_before = True

    if zc.player and zc.player.alive:
        store_player_state(mc, zc.player)
        yield zc.world, zone_flyout(zc)

    zc.world.destroy()

# =====
# Mission dialogs.

def conv_cr060_01(world, player, hulk, dragon):

    def fade_out_screen():
        world.fade_out(0.5)

    def player_look_hulk():
        player.headchaser.move_to(atref=hulk,
                                   angspeed=3.0, angacc=1.0)

    def player_look_front():
        player.headchaser.move_to(atref=hprtovect(Vec3(0, 0, 0)),
                                   angspeed=0.5, angacc=0.1)

    return Dialog(
        camnode=world.camera,
        pnode=world.node2d,

```

```

characters={
  "arend": Character(
    width=0.6, pos=(0.15, 0.05), size=10,
    color=rgba(255, 200, 0, 1.0),
    align="c", anchor="bc",
    node=player.ac.node, played=True),
  "hulk": Character(
    width=0.6, pos=(0.15, 0.05), size=10,
    color=rgba(200, 125, 0, 1.0),
    align="c", anchor="bc",
    node=hulk.node),
  "draggon": Character(
    width=0.6, pos=(0.15, 0.05), size=10,
    color=rgba(230, 130, 30, 1.0),
    align="c", anchor="bc",
    node=draggon.node),
},
branches={
  "start": [
    Speech("arend",
      Line(_("We are at the first point."))),
    Speech("arend",
      Line(_("I am going to leave the two of you now."),
        startf=player_look_hulk)),
    Speech("arend",
      Line(_("Be careful out there."))),
    Speech("hulk",
      Line(_("*noise*"), startf=player_look_front)),
    Speech("hulk",
      Line(_("Right, get on with your mission, boss."))),
    Speech("hulk",
      Line(_("We'll be fine."))),
    Pause(time=1.0, startf=fade_out_screen),
  ],
},
wpm speed=150,
)

```

```
def conv_cr060_02(world, player):
```

```

return Dialog(
  camnode=world.camera,
  pnode=world.node2d,
  characters={
    "arend": Character(
      width=0.6, pos=(0.15, 0.05), size=10,
      color=rgba(255, 200, 0, 1.0),
      align="c", anchor="bc",
      node=player.ac.node, played=True),
  },
  branches={
    "start": [
      Speech("arend",
        Line(_("..."))),
      Speech("arend",
        Line(_("Those are not Warlord's typical "

```



```

        "armor units...")),
        Speech("arend",
            Line(_("..."))),
        Speech("arend",
            Line(_("Disturbing implications..."))),
    ],
},
wpm speed=150,
)

def conv_cr060_03(world, player, enemyac5, enemyac6, chaser_player,
    chaser_enemyac5):

    def control_level_2():
        world.player_control_level = 2

    def fade_in_screen():
        world.action_music.set_context("boss")
        world.fade_in(0.5)

    def fade_out_screen():
        world.fade_out(0.5)

    def jump_objects_p1():
        world.chaser = chaser_player
        player.ac.jump_to(
            pos=Point3(player.ac.pos()[0], player.ac.pos()[1], 3177),
            hpr=Vec3(30,0,0),
            speed=200)
        enemyac5.jump_to(
            pos=pos_from_horiz(player.ac, Point3(5331, 18231, 4910)),
            hpr=hpr_from_horiz(player.ac, Vec3(180,0,0)),
            speed=200)
        enemyac6.jump_to(
            pos=pos_from_horiz(enemyac5, Point3(30, -50, 20)),
            hpr=hpr_from_horiz(enemyac5, Vec3(0,0,0)),
            speed=200)

    def switch_chaser_player():
        world.chaser = chaser_player

    def switch_chaser_enemyac5():
        world.chaser = chaser_enemyac5

    def player_set_autopilot():
        player.ac.set_ap(altitude=6000)

    def fade_out_screen():
        world.fade_out(0.5)

    return Dialog(
        camnode=world.camera,
        pnode=world.node2d,
        characters={
            "arend": Character(

```

```

        width=0.6, pos=(0.15, 0.05), size=10,
        color=rgba(255, 200, 0, 1.0),
        align="c", anchor="bc",
        node=player.ac.node, played=True),
    "dwpilotcom": Character(
        width=0.6, pos=(0.15, 0.05), size=10,
        color=rgba(25, 50, 100, 1.0),
        align="c", anchor="tc",
        node=player.ac.node),
    "dwpilot": Character(
        width=0.6, pos=(0.15, 0.05), size=10,
        color=rgba(25, 50, 100, 1.0),
        align="c", anchor="bc",
        node=enemyac5.node),
},
branches={
    "start": [
        Pause(time=0.5),
        Pause(time=0.5,
            startf=control_level_2, endf=fade_in_screen),
        Speech("dwpilotcom",
            Line(_("Wrong place-- *mild noise* -- to stick "
                "your nose, Colonel."),
                startf=jump_objects_pl)),
        Speech("dwpilotcom",
            Line(_("And now we have to kill you."))),
        Speech("arend",
            Line(_("..."))),
        Speech("arend", [
            Line(_("Why? What are you doing here?"),
                branch="inquisitive"),
            Line(_("You mercs are in league with Tycoon."),
                branch="guessing"),
            Line(_("*remains silent*"),
                branch="silent"),
        ]),
    ],
    "inquisitive": [
        Speech("dwpilotcom",
            Line(_("..."))),
        Speech("dwpilot",
            Line(_("Nothing personal, Colonel."),
                startf=switch_chaser_enemyac5)),
        Speech("dwpilot",
            Line(_("Just orders. Something "
                "I'm sure you understand."))),
        Speech("dwpilot",
            Line(_("Goodbye."), branch="conclusion")),
    ],
    "guessing": [
        Speech("dwpilotcom",
            Line(_("..."))),
        Speech("dwpilot",
            Line(_("You should have followed your orders "
                "closely, Colonel."),
                startf=switch_chaser_enemyac5)),
    ],
}

```

```

        Speech("dwpilot",
            Line(_("You Dutch boys, barely comprehend "
                "an inch of this whole affair."))),
        Speech("dwpilot",
            Line(_("Goodbye, Colonel."),
                branch="conclusion")),
    ],
    "silent": [
        Speech("dwpilotcom",
            Line(_("..."))),
        Speech("dwpilot",
            Line(_("Ain't that a pity, Dutch boy?"),
                startf=switch_chaser_enemyac5)),
        Speech("dwpilot",
            Line(_("Goodbye."), branch="beginfight")),
    ],
    "conclusion": [
        Speech("arend",
            Line(_("You have yet to finish your task, scum."),
                startf=switch_chaser_player)),
        Speech("arend",
            Line(_("Come, get me."), branch="beginfight")),
    ],
    "beginfight": [
        Speech("arend",
            Line(_("*scowls and pulls the stick*"),
                time=3.0, startf=player_set_autopilot)),
        Pause(time=1.0, startf=fade_out_screen),
    ],
},
wpm speed=150,
)

```

```
def conv_cr060_04(world, player, hulk):
```

```

    return Dialog(
        camnode=world.camera,
        pnode=world.node2d,
        characters={
            "arend": Character(
                width=0.6, pos=(0.15, 0.05), size=10,
                color=rgba(255, 200, 0, 1.0),
                align="c", anchor="bc",
                node=player.ac.node, played=True),
            "hulk": Character(
                width=0.6, pos=(0.15, 0.05), size=10,
                color=rgba(200, 125, 0, 1.0),
                align="c", anchor="bc",
                node=hulk.node),
        },
        branches={
            "start": [
                Speech("hulk",
                    Line(_("Just-- *mild noise* --in time!))),
                Speech("arend",
                    Line(_("*surprised* Hulk??))),
            ],
        },
    )

```

```

        Speech("hulk",
              Line(_("Let's nail these bastards-- *mild noise* "
                    "-- Colonel."))),
    ],
},
wpm speed=150,
)

def conv_cr060_05(world, player, hulk):

def hulk_set_ap():
    hulk.set_ap(heading=340, speed=360, useab=True)

return Dialog(
    camnode=world.camera,
    pnode=world.node2d,
    characters={
        "arend": Character(
            width=0.6, pos=(0.15, 0.05), size=10,
            color=rgba(255, 200, 0, 1.0),
            align="c", anchor="bc",
            node=player.ac.node, played=True),
        "hulk": Character(
            width=0.6, pos=(0.15, 0.05), size=10,
            color=rgba(200, 125, 0, 1.0),
            align="c", anchor="bc",
            node=hulk.node),
    },
    branches={
        "start": [
            Speech("hulk",
                  Line(_("..."))),
            Speech("hulk",
                  Line(_("And that's-- *mild noise* --that."))),
            Speech("arend",
                  Line(_("Hulk! What are you doing here!"))),
            Speech("hulk",
                  Line(_("I had a nagging feeling-- *noise* --"
                        "you would end up in trouble."))),
            Speech("hulk",
                  Line(_("*mild noise* --I told Dragoon to finish "
                        "the patrol, while I come looking for you."))),
            Speech("arend",
                  Line(_("And what if they intercept her, now, huh!?"))),
            Speech("arend",
                  Line(_("You disobeyed my order, captain!"))),
            Speech("hulk",
                  Line(_("I-- but lieutenant can handle herself--"),
                        ctimefac=0.8)),
            Speech("arend",
                  Line(_("*shouts angry* GET BACK to her, NOW!"))),
            Speech("hulk",
                  Line(_("...")), "radiol-c"),
            Speech("hulk",
                  Line(_("Understood, boss. Hulk, out."),
                        startf=hulk_set_ap)),

```

```

        ],
    },
    wpm speed=150,
)

# =====
# Background.

mission_skipmenu = False
mission_skipconfirm = False
mission_menumusic = "audio/music/cr-menu.ogg"
mission_debriefing = "late"
mission_mustdrink = False

def mission_setbg (mc, gc):

    pass

# =====
# Stage dialogs.

def mission_drinkconv (dc, mc, gc):

    def bg_snd1_start():
        dc.sound1 = Sound2D("audio/sounds/_bg-cantina-crowd.ogg",
                             pnode=dc.node, volume=0.8, loop=True,
                             play=True)
        dc.sound2 = Sound2D("audio/sounds/_bg-bar-music-2.ogg",
                             pnode=dc.node, volume=0.4, loop=True,
                             play=True)

    return Dialog(
        pnode=dc.fgnode,
        characters={
            "arend": Character(shortdes=_("Arend"),
                                portrait="eagle.png", prtsize=0.6,
                                width=1.4, pos=(0.7, 0.6), size=14,
                                color=rgba(255, 200, 0, 1.0),
                                align="l", anchor="tr",
                                unfoldfac=0.5,
                                played=True),
            "hulk": Character(shortdes=_("Hulk"),
                                portrait="eagle.png", prtsize=0.6,
                                width=1.4, pos=(-0.7, -0.2), size=14,
                                color=rgba(200, 125, 0, 1.0),
                                align="l", anchor="tl",
                                unfoldfac=0.5),
            "narrator": Character(width=2.0, pos=(-1.15, 0.80),
                                font=FONT_RUS, size=16,
                                color=rgba(40, 40, 255, 1.0),
                                shcolor=None,
                                olcolor=rgba(0, 0, 0, 0.5), olwidth=0.5, olfeather=0.2,
                                align="l", anchor="tl", swipe=0,
                                unfoldfac=0.0,

```

```

        wpm speed=150),
    },
    branches={
        "start": [
            Speech("narrator",
                Line(_("Las Palmas bar, evening"),
                    time=2.0, startf=bg_snd1_start)),
            Pause(time=2.0),
            Entry(["arend", "hulk"]),
            Speech("hulk",
                Line(_("*sneezes while filling his glass*"))),
            Speech("hulk",
                Line(_("Locals are getting on my nerves."))),
            Speech("hulk",
                Line(_("They are constantly snooping around "
                    "the airfield."))),
            Speech("hulk",
                Line(_("These civilians think of us as some sort "
                    "of a goddamn attraction. *swigs*"))),
            Speech("arend",
                Line(_("I believe that others are enjoying that "
                    "attention."))),
            Speech("hulk",
                Line(_("*snorts...* Not I. *...clears his throat*"))),
            Speech("arend",
                Line(_("*leans his elbows at the bar* What is it "
                    "really that is bothering you, captain?"))),
            Speech("arend",
                Line(_("Franz?"))),
            Speech("hulk",
                Line(_("Look, Colonel..."))),
            Speech("hulk",
                Line(_("There is no way those MiGs shot at us."))),
            Speech("hulk",
                Line(_("We shot them down long before they had "
                    "a chance to do anything."))),
            Speech("hulk",
                Line(_("*knocks the bar with his finger* "
                    "Those Dark Water spooks, they did."))),
            Speech("arend",
                Line(_("I have reviewed your report, twice."))),
            Speech("arend",
                Line(_("You said you hadn't really seen that."))),
            Speech("hulk",
                Line(_("*swigs and shakes his head* "
                    "I didn't, but I am sure of it!"))),
            Speech("arend",
                Line(_("You are not offering any solid proof "
                    "to your claims, Hulk."))),
            Speech("arend",
                Line(_("Little can be done, about it. "
                    "You do understand?"))),
            Speech("hulk",
                Line(_("Just... *lifts his elbow, irritated*"))),
            Speech("hulk",
                Line(_("Just keep the Dark Water away from us, "

```

```

        "Arend. Please."))),
    Speech("arend",
        Line(_("*pours drink into a glass*"))),
    Speech("arend",
        Line(_("I am not a fan of them either..."))),
    Speech("arend",
        Line(_("...I will see what I can do."))),
    Speech("arend",
        Line(_("And don't worry about Franz, "
            "we will find him."))),
    Speech("hulk",
        Line(_("*snorts* I am sure we will..."))),
    Speech("hulk",
        Line(_("*swigs* ...unless Warlord's death squad "
            "finds him first, that is."))),
    Pause(time=1.0),
    Exit(),
    Pause(time=1.0),
    ],
    },
)

```

```
def mission_inconv (dc, mc, gc):
```

```
def bg_snd1_start():
```

```
    dc.sound1 = Sound2D("audio/sounds/_bg-empty.ogg",
        pnode=dc.node, volume=0.8, loop=True,
        play=True)
```

```
    dc.sound2 = Sound2D("audio/sounds/_bg-morning-1.ogg",
        pnode=dc.node, volume=0.2, loop=True,
        play=True)
```

```
def set_whiplash_and_hurricane_south():
    mc.whiplash_and_hurricane_south = True
```

```
def set_pyro_andPainter_south():
    mc.pyro_andPainter_south = True
```

```
return Dialog(
    pnode=dc.fgnode,
    characters={
        "arend": Character(shortdes=_("Arend"),
            portrait="eagle.png", prtsize=0.3,
            width=0.9, pos=(0.0, 0.45), size=12,
            color=rgba(255, 200, 0, 1.0),
            align="l", anchor="tc",
            unfoldfac=0.5,
            played=True),
        "dragon": Character(shortdes=_("Dragon"),
            portrait="eagle.png", prtsize=0.3,
            width=0.9, pos=(1.15, -0.45), size=12,
            color=rgba(230, 130, 30, 1.0),
            align="l", anchor="tc",
            unfoldfac=0.5),
        "hulk": Character(shortdes=_("Hulk"),
```

```

        portrait="eagle.png", prtsize=0.3,
        width=0.9, pos=(-0.7, -0.55), size=12,
        color=rgba(200, 125, 0, 1.0),
        align="l", anchor="tc",
        unfoldfac=0.5),
"whiplash": Character(shortdes=_("Whiplash"),
        portrait="eagle.png", prtsize=0.3,
        width=0.9, pos=(-0.25, -0.65), size=12,
        color=rgba(200, 150, 20, 1.0),
        align="l", anchor="tc",
        unfoldfac=0.5),
"hurricane": Character(shortdes=_("Hurricane"),
        portrait="eagle.png", prtsize=0.3,
        width=0.9, pos=(-1.15, -0.45), size=12,
        color=rgba(225, 210, 25, 1.0),
        align="l", anchor="tc",
        unfoldfac=0.5),
"pyro": Character(shortdes=_("Pyro"),
        portrait="eagle.png", prtsize=0.3,
        width=0.9, pos=(0.7, -0.55), size=12,
        color=rgba(210, 110, 10, 1.0),
        align="l", anchor="tc",
        unfoldfac=0.5),
"painter": Character(shortdes=_("Painter"),
        portrait="eagle.png", prtsize=0.3,
        width=0.9, pos=(0.25, -0.65), size=12,
        color=rgba(255, 175, 40, 1.0),
        align="l", anchor="tc",
        unfoldfac=0.5),
"narrator": Character(width=2.0, pos=(-1.15, 0.80),
        font=FONT_RUS, size=16,
        color=rgba(40, 40, 255, 1.0),
        shcolor=None,
        olcolor=rgba(0, 0, 0, 0.5), olwidth=0.5, olfeather=0.2,
        align="l", anchor="tl", swipe=0,
        unfoldfac=0.0,
        wpm speed=150),
},
branches={
    "start": [
        Speech("narrator",
            Line(_("Las Palmas airbase, early morning"),
                time=2.0, startf=bg_snd1_start)),
        Pause(time=2.0),
        Entry(["arend", "dragon", "hulk", "whiplash",
            "hurricane", "pyro", "painter"]),
        Speech("arend",
            Line(_("Alright boys and girls, pay attention..."))),
        Speech("arend",
            Line(_("This morning, we received the latest "
                "schedule from the command."))),
        Speech("arend",
            Line(_("We will be doing the third shift until "
                "the end of this week."))),
        Speech("painter",
            Line(_("Third shift. Very good."))),
    ]
}

```



```

Speech("arend",
      Line(_("*glances briefly at Painter* "
            "As you all know, captain Pinote has gone "
            "missing on his last mission."))),
Speech("arend",
      Line(_("Command will give the green light "
            "to the search party soon."))),
Speech("arend",
      Line(_("And we need to keep the enemy pinned down."))),
Speech("arend",
      Line(_("Today's mission will be an air patrol."))),
Speech("arend",
      Line(_("This is an overview of the whole task. "
            "**points at the display*"))),
Speech("whiplash",
      Line(_("Now that there is seven of us... "
            "who's going to catch a break?"))),
Speech("arend",
      Line(_("That's sorted, Major. Noone will "
            "be sitting idle."), branch="choice")),
],
"choice": [
  Speech("arend", [
    Line(_("You and Hurricane will take patrol routes "
          "to the south."), branch="south"),
    Line(_("Northern routes will be your and "
          "Hurricane's task."), branch="north"),
  ]),
],
"south": [
  Speech("arend",
        Line(_("*looks to the left* Pyro and Painter will "
              "take the northern routes."),
              startf=set_whiplash_and_hurricane_south,
              branch="continue")),
],
"north": [
  Speech("arend",
        Line(_("*looks to the left* Pyro and Painter will "
              "take the southern routes."),
              startf=set_pyro_andPainter_south,
              branch="continue")),
],
"continue": [
  Speech("arend",
        Line(_("*focuses his gaze in front* "
              "And me, Hulk, and Draggon will take "
              "central patrol routes."))),
  Speech("draggon",
        Line(_("Formation of three? That's odd...?"))),
  Speech("arend",
        Line(_("*nods faintly at Draggon* "
              "It will make more sense, Lieutenant."))),
  Speech("arend",
        Line(_("*eyes everyone* That's it. Questions?"))),
  Speech("arend",

```

```

        Line(_("."...")),
        Speech("arend",
            Line(_("Good."))),
        Speech("arend",
            Line(_("Take your mission instructions... "
                "*motions at the pile of tablets in front* "
                "...and prepare your jets."))),
        Speech("arend",
            Line(_("We are taking off in two hours."))),
        Speech("arend",
            Line(_("Dismissed."))),
        Exit(),
        UpdateChar("arend", CharMod(pos=(-0.6, -0.1))),
        UpdateChar("draggon", CharMod(pos=(0.5, 0.2))),
        UpdateChar("hulk", CharMod(pos=(0.6, -0.3))),
        Pause(time=2.0),
        Speech("arend",
            Line(_("Draggon, Hulk, remain for a moment."))),
        Speech("draggon",
            Line(_("Yes, Colonel?"))),
        Speech("hulk",
            Line(_("*faintly absent* What?"))),
        Speech("arend",
            Line(_("*eyes Hulk*"))),
        Speech("arend",
            Line(_("When we enter the area of "
                "our patrol routes..."))),
        Speech("arend",
            Line(_("I will be flying with the two of you, "
                "up to the first patrol point."))),
        Speech("arend",
            Line(_("Then..."))),
        Speech("arend",
            Line(_("You and... *glances briefly at Draggon* "
                "...Draggon will finish the rest of the task, "
                "while I will be flying elsewhere."))),
        Speech("hulk",
            Line(_("*rises his eyebrow faintly* Elsewhere...?"))),
        Speech("arend",
            Line(_("Yes, I am going on a reconnaissance."))),
        Speech("arend",
            Line(_("To record a possible area of interest."))),
        Speech("hulk",
            Line(_("I see..."))),
        Speech("draggon",
            Line(_("Understood, Colonel."))),
        Pause(time=1.0),
        Exit(),
        Pause(time=1.0),
    ]
},
)

```

```
def mission_outconv (dc, mc, gc):
```

```

def bg_snd1_start():
    dc.sound1 = Sound2D("audio/sounds/_bg-empty.ogg",
                        pnode=dc.node, volume=0.8, loop=True,
                        play=True)
    dc.sound2 = Sound2D("audio/sounds/_bg-morning-1.ogg",
                        pnode=dc.node, volume=0.2, loop=True,
                        play=True)

return Dialog(
    pnode=dc.fgnode,
    characters={
        "arend": Character(shortdes=_("Arend"),
                            portrait="eagle.png", prtsize=0.6,
                            width=1.4, pos=(-0.7, 0.6), size=14,
                            color=rgba(255, 200, 0, 1.0),
                            align="l", anchor="tl",
                            unfoldfac=0.5,
                            played=True),
        "whiplash": Character(shortdes=_("Whiplash"),
                              portrait="eagle.png", prtsize=0.6,
                              width=1.4, pos=(0.7, -0.2), size=14,
                              color=rgba(200, 150, 20, 1.0),
                              align="l", anchor="tr",
                              unfoldfac=0.5),
        "painter": Character(shortdes=_("Painter"),
                              portrait="eagle.png", prtsize=0.6,
                              width=1.4, pos=(0.7, -0.2), size=14,
                              color=rgba(255, 175, 40, 1.0),
                              align="l", anchor="tr",
                              unfoldfac=0.5),
        "narrator": Character(width=2.0, pos=(-1.15, 0.80),
                              font=FONT_RUS, size=16,
                              color=rgba(40, 40, 255, 1.0),
                              shcolor=None,
                              olcolor=rgba(0, 0, 0, 0.5), olwidth=0.5, olfeather=0.2,
                              align="l", anchor="tl", swipe=0,
                              unfoldfac=0.0,
                              wpm speed=150),
    },
    branches={
        "start": [
            Speech("arend",
                  Line(_("*climbs down the ladder to the ground*"))),
            Speech("arend",
                  Line(_("..."),
                    cond=mc.whiplash_and_hurricane_south,
                    branch="painter")),
            Speech("arend",
                  Line(_("..."),
                    cond=mc.pyro_andPainter_south,
                    branch="whiplash")),
        ],
        "painter": [
            Speech("painter",
                  Line(_("*paces fast* Colonel, what happened to you!? "
                    "Draggon said--"), ctimefac=0.8)),
        ]
    }
)

```

```

Speech("arend",
    Line(_("*frowned* Shit, that's what happened, "
        "Major."))),
Speech("arend",
    Line(_("From now on, Dark Water is to be considered "
        "a hostile force!"))),
Speech("painter",
    Line(_("Dark Wat-- hostile?"), ctime="cut")),
Speech("arend",
    Line(_("Status on all groups. Report."))),
Speech("painter",
    Line(_("*scratches back of the head* "
        "Disturbing news, Colonel."))),
Speech("painter",
    Line(_("Southern group... both Hurricane and "
        "Whiplash have gone missing."))),
Speech("arend",
    Line(_("Missing?"))),
Speech("painter",
    Line(_("Tower confirmed, they were shot down."))),
Speech("arend",
    Line(_("*palms his face*"))),
Speech("painter",
    Line(_("Northern group, Pyro and I, returned from "
        "our patrol sortie with little to report."),
        ctime="cut")),
Speech("arend",
    Line(_("I have to speak with the command, "
        "immediately."))),
Speech("painter",
    Line(_("*paces off*"))),
Exit("painter"),
Pause(time=1.0),
Exit(),
Pause(time=1.0),
],
"whiplash": [
Speech("painter",
    Line(_("*paces fast* Colonel! "
        "What happened out there!?"))),
Speech("arend",
    Line(_("*frowned* Shit, that's what happened, "
        "Major."))),
Speech("arend",
    Line(_("From now on, Dark Water is to be considered "
        "a hostile force!"))),
Speech("painter",
    Line(_("Fuck, Hulk was right."))),
Speech("arend",
    Line(_("Status on all groups. Report."))),
Speech("painter",
    Line(_("*frowns* Bad news, Colonel."))),
Speech("painter",
    Line(_("Both Pyro and Painter have gone missing "
        "on their southern sortie."))),
Speech("arend",

```

```

        Line(_("Missing?")),
        Speech("painter",
            Line(_("It appears they were shot down."))),
        Speech("arend",
            Line(_("*palms his face*"))),
        Speech("painter",
            Line(_("Me and Hurricane, we finished our sortie "
                "with little to report."), ctime="cut")),
        Speech("arend",
            Line(_("I have to speak with the command, "
                "immediately."))),
        Speech("painter",
            Line(_("*paces off*"))),
        Exit("painter"),
        Pause(time=1.0),
        Exit(),
        Pause(time=1.0),
    ]
},
)

```

Kao и обично, на почетку уводимо све потребене функције за конструисање мисије, а одмах потом креирамо и функцију једне зграде:

```

def airfield_2000x30 (zc, mc, gc, pos, hpr, side):

    runway_pos = pos
    runway_hpr = hpr
    runway = CustomBuilding(
        world=zc.world, name="af2000x30_runway", side=side,
        strength=6000, minhitdmg=3000, maxhitdmg=5000, rcs=0,
        hitboxdata=[],
        modelpath="models/buildings/runway/runway2000x30.egg",
        texture="models/buildings/runway/runway2000x30_tex.png",
        normalmap="models/buildings/runway/runway2000x30_nm.png",
        glossmap="models/buildings/runway/runway2000x30_gls.png",
        clamp=False,
        pos=pos,
        hpr=hpr,
        damage=0,
        castshadow=False)
    runway_surface = VirtualHorizPoly(
        poly=[pos_from_point(runway_pos, runway_hpr,
            Point2( 15.0, 1000.0)).getXy(),
            pos_from_point(runway_pos, runway_hpr,
            Point2( 15.0, -1000.0)).getXy(),
            pos_from_point(runway_pos, runway_hpr,
            Point2(-15.0, -1000.0)).getXy(),
            pos_from_point(runway_pos, runway_hpr,
            Point2(-15.0, 1000.0)).getXy()],
        convex=True,
        flush=True,
        elev=(zc.world.elevation(runway_pos) + 0.1),
        gtype=GROUND.RUNWAY)
    zc.world.terrains[0].add_virtual_surface(runway_surface)

```

```

runway.player_ground_pos = pos_from_point(runway_pos, runway_hpr,
                                           Point3(0.0, -900.0, 0.0))
runway.player_ground_hpr = runway_hpr
return runway

```

Ово је функција у којој креирамо писту, чији је модел 2000 метара дугачак и 30 метара широк. Зграду, тј. писту, могли смо да креирамо и директно у улазној функцији зоне, користећи класу CustomBuilding. Међутим, писта је специфична врста зграда, јер на њу може да слеће авион, па је неопходно такву зграду додатно дефинисати. Потребно је омеђати специјалну површину, путем класе VirtualHorizPoly, која ће бити регистрована као равна површина писте. Због свега тога је код за потпуно креирање овог објекта подужи, па смо одлучили да га сместимо у засебну функцију. У перспективи, све ове специјалне функције требало би да буду написане у засебном генералном фајлу кампање (посебно ако се користе у разним мисијама те кампање), а не у фајловима самих мисија.

Пошто смо дефинисали функцију писте, следи заглавље мисије:

```

def init_cache (mc, gc):

    cache_bodies(["f16", "f18", "mig29", "bradley",
                 "warehouse_1", "runway2000x30"])

def mission_start (gc):

    mission = Mission(gc)

    mission.add_init(loadf=init_cache)

    mission.add_zone("canary", clat=28.22, clon=-16.04,
                    enterf=cr060_canary_enter, exitf=cr060_canary_exit,
                    loopf=cr060_canary_loop)
    mission.add_zone("wsahara", clat=24.56, clon=-13.68,
                    enterf=cr060_wsahara_enter, exitf=cr060_wsahara_exit,
                    loopf=cr060_wsahara_loop)

    mission.switch_zone_pause = 1.0

    mc = mission.context
    mc.player_fuelfill = 1.0
    mc.player_ammo_cannons = [500]
    mc.player_ammo_launchers = [(None, 4), (Aim9, 4)]
    mc.player_mfd_mode = "overmap"

    mission.switch_zone("canary")

    # Alliances
    mc.alliances = [("lw", "dwater")]

    # Mission start time
    mc.world_day_time = hrmin_to_sec(9, 21)

    # Set mission objectives.
    mc.objective_patrol_x = False

```

```
return mission
```

Као што се може приметити, у односу на пример бр. 1, сад креирамо две зоне, са по три стандардне функције (улазна, петља, излазна). `mission.switch_zone_pause` је резервисани атрибут који каже колико секунди се чека пре него што се играч пусти у другу зону, након што се та зона учитала. Тај атрибут има чисту естеску вредност, за ублажавање прелаза.

Потом опремамо играча, затим наводимо у којој ће зони почети мисија, задајемо време почетка мисије, и онда складиштимо у један `mc` атрибут листу савезништва између неких фракција које ће се појавити у мисији ("`lw`" је фракција којој ће припадати и играч). То савезништво још увек није дефинисано, а овај атрибут ће у ту сврху бити употребљен мало касније. На крају дефинишемо `mc` атрибуте објектива у мисији. Како имамо само један објектив у мисији, дефинишемо само један атрибут. Атрибути објектива овако дефинисани нису ни на који начин специфични у односу на било који други произвољно направљени атрибут. Једини разлог зашто резервишемо атрибуте за објективе у заглављу мисије јесте чисто организациони. Ако унапред постоје, смислено је већ у заглављу мисије дефинисати `mc` атрибуте који ће се после користити у неким од функција зона мисије.

Сад прелазимо на улазну функцију прве зоне у мисији, "`canary`":

```
# =====
# Canary islands zone

def cr060_canary_enter (zc, mc, gc):

    # Create world.
    setup_world(zc, mc, gc,
                terraintype="15-canary",
                skytype="default2",
                stratusdens=0.8,
                cumulusdens=0.9,
                cirrusdens=2.0,
                cloudseed=1506,
                playercntl=0,
                alliances=mc.alliances)

    # Add base complex, runway, etc.
    zc.af runway = airfield_2000x30(zc, mc, gc, side="lw",
                                   pos=Point2(9100, -4200),
                                   hpr=Point3(0,0,0))

    # Add player and wingmen if any.
    # Depending if this is the first or a subsequent entry into
    # the zone, set player on the runway or in the air.
    if not zc.visited_before:
        zc.player = create_player(
            mc=mc, world=zc.world,
            pos=zc.af runway.player_ground_pos,
            hpr=zc.af runway.player_ground_hpr,
            speed=0, onground=True,
            texture="models/aircraft/f16/f16dutch_tex.png")
        zc.hulk = F16(
            world=zc.world, name="hulk", side="lw",
```

```

        texture="models/aircraft/f16/f16dutch_tex.png",
        fuelfill=1.0,
        pos=pos_from_horiz(zc.player.ac, Point3(+20, 15, 0)),
        hpr=hpr_from_horiz(zc.player.ac, Vec3(0, 0, 0)),
        speed=0, onground=True,
        lnammo=[(None, 6), (Aim9, 2)],
        skill="veteran")
zc.draggon = F16(
    world=zc.world, name="draggon", side="lw",
    texture="models/aircraft/f16/f16dutch_tex.png",
    fuelfill=1.0,
    pos=pos_from_horiz(zc.player.ac, Point3(-20, 20, 0)),
    hpr=hpr_from_horiz(zc.player.ac, Vec3(0, 0, 0)),
    speed=0, onground=True,
    lnammo=[(None, 6), (Aim9, 2)],
    skill="pilot")
zc.draggon.set_auto_attack()
zc.hulk.set_auto_attack()
formation_triplet(zc.player.ac, zc.draggon, zc.hulk,
                  compact=1.0, jumpto=False)
else:
    zc.player = create_player(
        mc=mc, world=zc.world,
        pos=pos_from_horiz(zc.afrunway, Point3(0, -12137, 1927)),
        hpr=hpr_from_horiz(zc.afrunway, Vec3(0, 0, 0)),
        speed=200,
        texture="models/aircraft/f16/f16dutch_tex.png")

# Add objects.
pass

# Set navpoints/waypoints.
zc.player.add_navpoint(
    name="nav1", longdes=_("Navpoint 1"), shortdes=_("NAV1"),
    pos=Point2(0, 0), radius=640000, height=-1,
    tozone="wsahara")
add_base_waypoint(zc, mc, base=zc.afrunway, name="base",
                  longdes=_("land"), shortdes=_("BASE"))
if zc.visited_before:
    zc.player.update_navpoint("nav1", active=False)

```

Путем функције `setup_world`, прво креирамо свет (терен, облаци, итд). У односу на пример бр. 1, може се приметити позив једног додатног аргумента, `alliances`. У тај аргумент функције смештамо листу савезништва, `mc.alliances`, коју смо претходно направили у заглављу мисије. Сад је савезништво између фракција скроз дефинисано, у овој зони, те ће се две наведене фракције сматрати пријатељским и међусобно се неће нападати. На радару играча, сви пријатељски авиони биће означени другачије (кружићем уместо трогулом).

После стварања света, позивамо функцију за писту коју смо претходно направили, на одређеној позицији на терену (9100, -4200), окренуту у одређеном смеру (0, што ће рећи север). Писту треба постављати увек на оном делу терена који је скроз раван. У супротном ће доћи до преклапања писте и терена. Уколико жељено место на терену није равно, онда је тај део терена могуће поравнати у неком малом радијусу (рецимо 4 километара) око жељене тачке. Те специјалне равне површине на терену могуће је дефинисати у `*.dat` фајловима терена на следећи начин:


```
[flat-runway]
centerx=9.100
centery=-4.200
#centerz=
radius=3.0
radiusout=4.0
```

Кад се следећи пут изгради терен, изградиће се са равним површинама око задатих координата.

У наставку креирамо играча и његова два партнера, Хулка и Драгона, који су сви заједно приземљени, onground=True, и постављени на почетак писте:

```
# Add player and wingmen if any.
# Depending if this is the first or a subsequent entry into the zone,
# set player on the runway or in the air.
if not zc.visited_before:
    zc.player = create_player(
        mc=mc, world=zc.world,
        pos=zc.af runway.player_ground_pos,
        hpr=zc.af runway.player_ground_hpr,
        speed=0, onground=True,
        texture="models/aircraft/f16/f16dutch_tex.png")
    zc.hulk = F16(
        world=zc.world, name="hulk", side="lw",
        texture="models/aircraft/f16/f16dutch_tex.png",
        fuelfill=1.0,
        pos=pos_from_horiz(zc.player.ac, Point3(+20, 15, 0)),
        hpr=hpr_from_horiz(zc.player.ac, Vec3(0, 0, 0)),
        speed=0, onground=True,
        lnammo=[(None, 6), (Aim9, 2)],
        skill="veteran")
    zc.draggon = F16(
        world=zc.world, name="draggon", side="lw",
        texture="models/aircraft/f16/f16dutch_tex.png",
        fuelfill=1.0,
        pos=pos_from_horiz(zc.player.ac, Point3(-20, 20, 0)),
        hpr=hpr_from_horiz(zc.player.ac, Vec3(0, 0, 0)),
        speed=0, onground=True,
        lnammo=[(None, 6), (Aim9, 2)],
        skill="pilot")
    zc.draggon.set_auto_attack(["plane"])
    zc.hulk.set_auto_attack(["plane"])
    formation_triplet(zc.player.ac, zc.draggon, zc.hulk,
                      compact=1.0, jumpto=False)
else:
    zc.player = create_player(
        mc=mc, world=zc.world,
        pos=pos_from_horiz(zc.af runway, Point3(0, -12137, 1927)),
        hpr=hpr_from_horiz(zc.af runway, Vec3(0, 0, 0)),
        speed=200,
        texture="models/aircraft/f16/f16dutch_tex.png")
```

Важно је напоменути да, и ако су Хулк и Драгон два пријатељска пилота, они су истовремено и независни, односно нису вингмени играча. Код специфичан за

вингмене постоји, али тренутно једва функционише, тако да се тим кодом нећемо бавити овог пута.

Пријатељским авионима задата је специфична наредба:

```
.set_auto_attack(["plane"])
```

Ова наредба каже компјутерски контролисаном авиону да сам бира своје мете, искључиво у оквиру фамилије објеката "plane". Кад се зада ова наредба, авион ће, управљајући се према подацима са својих сензора, аутоматски нападати све оне објекте из наведене фамилије који припадају непријатељским фракцијама (односно свим фракцијама које нису наведене као савезничке). Може се задати и низ фамилија, `.set_auto_attack(["plane", "vehicle", "building"])`, а може и да се не зада ниједна фамилија `.set_auto_attack()`, у ком случају ће авион нападати све фамилије против којих је наоружан.

Следећа битна команда је функција формације:

```
formation_triplet(zc.player.ac, zc.draggon, zc.hulk,  
compact=1.0, jumpto=False)
```

Ово је функција формације за три авиона која је дефинисана у `src/blocks/missiontools.py`. Тренутно постоје још и `formation_pair` и `formation_twopairs`. Као први наведени авион, играч је вођа формације, а Хулк и Драгон се налазе с леве и десне стране играча. Компактност формације је `compact=1.0` (1.0 је нека основна вредност за међусобну удаљеност актера формације и може бити било који позитиван број). `jumpto` је често веома користан аргумент за специфичну употребу. Када се ова функција изврши, ако је овај аргумент постављен на `True`, сви пратећи авиони у формацији аутоматски ће да скоче на своје позиције у односу на вођу, а ако је на `False`, онда ће пратећи авиони путем навигационог аутопилота настојати да долете на своје позиције у формацији.

Приметно је да играча стварамо два пута, и да је стварање играча условљено упитом, `if not zc.visited_before`. Пошто ова мисија има две зоне, треба узети у обзир два случаја: када је играч први пут у зони и када играч сваки наредни пут уђе у зону. `visited_before` је `zc` атрибут који једноставно прати то стање. На почетку је постављен на `None`, што ће играча створити приземљеног на писти, а кад играч први пут напусти зону овај атрибут ће бити постављен на `True`, тако да ће сваки наредни пут када играч уђе у ову зону, бити створен негде у ваздуху. У нашем примеру конкретно, биће створен у близини писте, окренут таман како треба за слетање.

Треба још напоменути да играч може да има избор да ручно слети или да аутоматски слети (ако неће да ризикује да се сруши после, могуће, тешке мисије коју је једва обавио). Мада је омогућавање овог избора стриктно на дизајнеру кампање.

У наредном реду, обично стварамо неке конкретне објекте (нпр. противничке авионе) ако су присутни одмах по уласку у зону. Како у овој зони нема никаквих специјалних објеката, прелазимо одмах на стварање навигационих тачака:

```
# Set navpoints/waypoints.  
zc.player.add_navpoint(  
    name="nav1", longdes=_("Navpoint 1"), shortdes=_("NAV1"),  
    pos=Point2(0, 0), radius=640000, height=-1,  
    tozone="wsahara")  
add_base_waypoint(zc, mc, base=zc.af runway, name="base",  
    longdes=_("land"), shortdes=_("BASE"))  
if zc.visited_before:  
    zc.player.update_navpoint("nav1", active=False)
```

Навигационе тачке су један од најважнијих елемената скриптовања мисије. У игри постоје две основне врсте тачака за навигацију:

- waypoint
- navpoint

"Navpoint", или навигациона тачка, јесте апстрактна тачка која није видљива играчу, а која служи пре свега да повезује зоне. У овој зони, названој "canary", постоји само једна навигациона тачка, и она води у једину другу зону у овој мисији, `tozone="wsahara"`. Позиција те навигационе тачке је у центру терена `pos=Point2(0, 0)`, њен радијус `radius=640000` тј. 640 километара, а њена висина бесконачна, `height=-1`. Када је висина `-1` тј. бесконачна, радијус тачке је бесконачно високи ваљак. Када је висина неки позитиван број, онда је радијус сфера чији се центар налази на тој задатој висини.

Зашто смо поставили ову навигациону тачку у центар света зоне, њен радијус да буде огромних 640 километра, а њена висина бесконачна? То је зато што желимо да су цео терен (који је, подсећамо, површине 320x320 километра квадратних) и цео небески свод унутар радијуса ове навигационе тачке. Када је играч унутар тог радијуса, и притом није под нападом најмање 10 секунди, а сама тачка није закључана, играч добија могућност да са листе која се појави на екрану притиском на тастер одлети у другу зону. Уколико навигациона тачка својим радијусом покрива цео терен, то значи да ће играч увек бити унутар радијуса навигационе тачке, а то пак значи да играч с било ког места у зони може да покрене одлетање у другу зону, ако су испуњени сви остали услови. Понекад, у зависности од контекста радње унутар зоне, има смисла да је радијус навигационе тачке мали и да је сама тачка смештена у неки угао арене унутар зоне, тако да играч, коме рецимо истиче време или бежи од некога, мора да уђе у радијус навигационе тачке да би могао и да покрене одлетање у наредну зону.

"Waypoint", или путна тачка, јесте неко обележено место од интереса унутар зоне. Те тачке су видљиве на мапи терена (која се тренутно може позвати на ТВ екрану у авиону играча) и излистане су на ХУД-у у навигационом режиму. Иако је њихова сврха пре свега оријентациона, да упуте играча на одређено место, ове тачке треба посматрати и као окидаче где дизајнер мисије планира да отпочне неки специфичан контекст радње. У овој зони направили смо једну доста специфичну врсту путне тачке, која у себи садржи и навигациону тачку, и која је везана пре свега за писту и процес слетања. Типична врста путне тачка биће употребљена у улазној функцији наредне зоне, но зарад компактности овог упуства, одмах ћемо је показати овде:

```
zc.player.add_waypoint(  
    name="pat1", longdes=_("patrol point 1"), shortdes=_("PAT1"),  
    pos=zc.pat1pos, radius=2000, height=-1)
```

Да још напоменемо, за разлику од навигационе тачке где су имена, подведена под аргументе `longdes` и `shortdes` углавном небитна, код путне тачке `shortdes` је име које ће се појавити на мапи и на ХУД-у. Тако да код путних тачка треба пазити како се називају.

На крају ове улазне функције зоне налази се још један упит:

```
if zc.visited_before:  
    zc.player.update_navpoint("nav1", active=False)
```

Намера је да, чим се играч врати назад у ову зону, навигациона тачка која га води у другу зону буде закључана. Јер ако играч одлети поново у другу зону, а контекст те друге зоне није предвиђен за повратак играча, цео контекст радње би могао да се распадне. Тиме што смо просто укинули могућност повратака у другу зону, тако што смо закључали навигациону тачку по повратку, обезбедили смо се од могућих

непредвиђених последица. Ово међутим није најсрећније решење, јер одузима слободу играчу. Генерално правило је да што играч има мање слободе, то је лакше предвидети шта све играч може да уради, али и лакше оштетити имерзивност игре.

То би било све што се тиче улазне функције зоне, "canary". Сад следи функција петље:

```
def cr060_canary_loop (zc, mc, gc):  
  
    if (not mc.mission_failed and not zc.visited_before and  
        (zc.hulk.shutdown or zc.draggon.shutdown)):  
        zc.player.update_navpoint("nav1", active=False)  
        mission_failed(zc, mc, gc,  
                      reason=_("You shot down your own, pilot."))  
  
    if (not zc.nail_player and mc.mission_failed and  
        zc.world.stopwatch("countdown_nailplayer") > 4):  
        zc.world.explosion_damage(force=1000, ref=zc.player.ac.pos())  
        zc.nail_player = True  
  
    return zc.world, 1.0
```

Одмах се може приметити да је ово понављајућа функција петље. У зони "canary" се у принципу не дешава ништа. Нема никакве специјалне радње. Скоро да смо могли ништа и да не напишемо у овој функцији. Да изгледа само овако:

```
def cr060_canary_loop (zc, mc, gc):  
  
    return zc.world, 1.0
```

Међутим...

Иако се формално ништа не дешава, ипак су у формацији са играчем присутни још Хулк и Драгон. Код који смо написали у функцији петље усмерен је пре свега на играче паметњаковиће:

```
if (not mc.mission_failed and not zc.visited_before and  
    (zc.hulk.shutdown or zc.draggon.shutdown)):  
    zc.player.update_navpoint("nav1", active=False)  
    mission_failed(zc, mc, gc,  
                  reason=_("You shot down your own, pilot."))  
  
if (not zc.nail_player and mc.mission_failed and  
    zc.world.stopwatch("countdown_nailplayer") > 4):  
    zc.world.explosion_damage(force=1000, ref=zc.player.ac.pos())  
    zc.nail_player = True
```

Уколико играч обори макар једног од своја два партнера, најпре одмах закључавамо могућност летења у зону "wsahara", затим проглашавамо пропалу мисију са наведеним разлогом:

```
mission_failed(zc, mc, gc,  
              reason=_("You shot down your own, pilot."))
```

и најзад, након 4 секунде, дижемо играча у ваздух функцијом:

```
zc.world.explosion_damage(force=1000, ref=zc.player.ac.pos())
```

Генерално говорећи, дизајнер мисије треба увек да има спремљен план и за ону сорту играча која нема намеру да игра игру озбиљно, која намерава да се блесави, те која намерно ради ствари које очигледно не треба да ради. Ми смо се у овом примеру одлучили за најјефикасније, али истовремено и најмање имерзивно решење, а то је да ликвидирамо играча. Уколико жели, дизајнер може да узме у обзир и ову врсту понашања у контекст радње, па да, рецимо, на крају пропале мисије лика кога играч води на позорници дочека претпостављени који ће да му најпре натрљати нос, а потом га шутнути из даље кампање, уз велики дебели -- "GAME OVER!". Неки од стандарних решења у играма данас, за овај конкретан проблем пуцања на саборце, јесу искључивање пријатељске ватре или уништавање играча, као што смо ми урадили.

Пошто је ова функција петље понављајућа, па се цела функција изнова и изнова извршава у свакој секунди, јасно је да радњу више не можемо да градимо секвенцијално, као код проточне верзије. То значи да функција петље углавном мора бити састављена од гомила упита са затварачима. Затварачи, попут `zc.nail_player = True`, служе да кад једном извршавање кода прође услов упита, у наредном пролазу кроз функцију петље више не може да се уђе у извршавање тог контекста.

Такође, ако је потребно да се чека са извршавањем неког контекста, више не можемо просто да зауставимо извршавање целе функције са `yield`, `zc.world` као што смо чинили у проточној верзији функције петље, већ морамо да користимо штоперице:

```
zc.world.stopwatch("<NAME>")
```

Штоперица је функција света која када се изврши први пут креће одмах да одбројава од 0 до задатог времена у секундама:

```
zc.world.stopwatch("countdown_nailplayer") > 4
```

Штоперицу је такође могуће ресетовати назад на нулу:

```
zc.world.stopwatch("countdown_nailplayer").reset()
```

У принципу, штоперица је веома користна алатка у процесу прављења понављајуће функције петље.

Сад следи излазна функција зоне "canary", која ће почети да се извршава оног тренутка када играч каже аутопилоту да га одведе у другу зону:

```
def cr060_canary_exit (zc, mc, gc):  
    zc.visited_before = True  
  
    if zc.hulk and not zc.hulk.outofbattle:  
        zc.hulk.jump_to(  
            pos=pos_from_horiz(zc.player.ac, Point3(50, 100, -10)),  
            hpr=hpr_from_horiz(zc.player.ac, Vec3(0, 0, 0)),  
            speed=zc.player.ac.speed()  
        )  
        store_plane_state(mc, zc.hulk)  
        mc.hulk_alive = True  
    else:  
        mc.hulk_alive = False
```

```

if zc.draggon and not zc.draggon.outofbattle:
    zc.draggon.jump_to(
        pos=pos_from_horiz(zc.player.ac, Point3(-50, 100, 10)),
        hpr=hpr_from_horiz(zc.player.ac, Vec3(0, 0, 0)),
        speed=zc.player.ac.speed())
    store_plane_state(mc, zc.draggon)
    mc.draggon_alive = True
else:
    mc.draggon_alive = False

if zc.player and zc.player.alive:
    store_player_state(mc, zc.player)
    yield zc.world, zone_flyout(zc)

zc.world.destroy()

```

Најпре постављамо већ раније описани атрибут `zc.visited_before = True`. Затим проверавамо стање Хулка и Драгона. Ако је Хулк присутан, `zc.hulk`, и није уклоњен из битке, `not zc.hulk.outofbattle` (атрибут `.outofbattle` је исто што и `.shutdown` or `.retreat`), тада скачемо дотичног пилота на одређену позицију близу играча, `zc.hulk.jump_to()`, затим памтимо његово опште стање, `store_plane_state(mc, zc.hulk)`, и постављамо један атрибут `mc.hulk_alive = True`, који ће бити задужен да прати његово присуство у целој мисији. Све то исто радимо и за Драгона, а на крају тог низа памтимо и стање играча. Најзад, окидамо функцију анимације одлетања, `zone_flyout(zc)`, и онда после краћег времена уклањамо целу зону, `zc.world.destroy()`.

Овде је важно запазити функцију:

```

.jump_to(
    pos=pos_from_horiz(zc.player.ac, Point3(50, 100, -10)),
    hpr=hpr_from_horiz(zc.player.ac, Vec3(0, 0, 0)),
    speed=zc.player.ac.speed())

```

У питању је једна веома важна функција која је апсолутно незаменљива за режирање било какве анимације у контексту радње унутар једне зоне. Ова функција просто служи да објекат премести на одређену позицију. Обично када се спусти завеса на секунд или два, односно зацрни екран, то је прилика да режисер све актере у наступајуће анимацији скочи на своје позиције. Ова функција је специфична за сваку фамилију објеката понаособ, а код фамилије авиона конкретно три су кључна аргумента:

- `pos`, позиција где ће објекат бити скочен.
- `hpr`, ротација објекта после скока.
- `speed`, брзина којом ће се објекат после скока кретати. Притом ће то бити фиксна брзина, од које компјутерски навођени објекат неће ни најмање одступати док год лети тачно право. Дакле режисер може да рачуна да му објекат неће одшетати из кадра услед малог одступања у задатој брзини због убразања или несавршености аутопилота да хирушки одржава жељену брзину.

Ову функцију смо искористили у излазној функцији зоне "sahara" управо да би сва три авиона била видљива у кадру приликом анимације одлетања играча и његових партнера из зоне.

Прелазимо на зону "sahara" у којој се дешава главни контекст радње у мисији. Улазна функција зоне изгледа овако:

```

# =====
# Western Sahara zone

def cr060_wsahara_enter (zc, mc, gc):

    # Create world.
    setup_world(zc, mc, gc,
                terraintype="16-wsahara",
                skytype="default2",
                stratusdens=0.4,
                cumulusdens=1.1,
                cirrusdens=1.8,
                cloudseed=1606,
                playercntl=0,
                alliances=mc.alliances)

    # Add base complex, runway, etc.
    zc.patlpos = Point2(-40002, 18915)
    zc.patxpos = Point2(-21495, -58290)
    zc.basewarehouse1 = Warehouse1(
        world=zc.world, name="barrack_1", side="dwater",
        texture="models/buildings/warehouse/warehouse_1_tex.png",
        normalmap="models/buildings/warehouse/warehouse_1_nm.png",
        pos=(zc.patxpos + Point2(0, 50)),
        hpr=Vec3(270, 0, 0))
    zc.basewarehouse2 = Warehouse1(
        world=zc.world, name="barrack_1", side="dwater",
        texture="models/buildings/warehouse/warehouse_1_tex.png",
        normalmap="models/buildings/warehouse/warehouse_1_nm.png",
        pos=(zc.patxpos + Point2(-25, -50)),
        hpr=Vec3(270, 0, 0))

    # Add player and wingmen if any.
    # Depending if this is the first or a subsequent entry into
    # the zone, set player on the runway or in the air.
    zc.player = create_player(
        mc=mc, world=zc.world,
        pos=Point3(-41212, 18009, 7452),
        hpr=Vec3(215, 0, 0),
        speed=210,
        texture="models/aircraft/f16/f16dutch_tex.png")
    if mc.hulk_alive and mc.draggon_alive:
        zc.hulk = recreate_plane(
            mc, world=zc.world, name="hulk",
            texture="models/aircraft/f16/f16dutch_tex.png",
            pos=pos_from_horiz(zc.player.ac, Point3(-200, 200, 35)),
            hpr=hpr_from_horiz(zc.player.ac, Vec3(0, 0, 0)),
            speed=210)
        zc.hulk.set_auto_attack(["plane"])
        zc.draggon = recreate_plane(
            mc, world=zc.world, name="hulk",
            texture="models/aircraft/f16/f16dutch_tex.png",
            pos=pos_from_horiz(zc.player.ac, Point3(-100, 250, 20)),
            hpr=hpr_from_horiz(zc.player.ac, Vec3(0, 0, 0)),
            speed=210)
        zc.draggon.set_auto_attack(["plane"])

```

```

    formation_pair(zc.hulk, zc.draggon, compact=0.1, jumpto=True)

# Add objects.
zc.vhc1 = Bradley(
    world=zc.world, name="larmor1", side="dwater",
    texture="models/vehicles/bradley/bradley_tex.png",
    pos=pos_from_horiz(zc.basewarehouse1, Point2(200, 0)),
    hpr=hpr_from_horiz(zc.basewarehouse1, Vec3(0, 0, 0)),
    speed=0.0)
zc.vhc2 = Bradley(
    world=zc.world, name="larmor2", side="dwater",
    texture="models/vehicles/bradley/bradley_tex.png",
    pos=pos_from_horiz(zc.vhc1, Point2(20, 0)),
    hpr=hpr_from_horiz(zc.vhc1, Vec3(0, 0, 0)),
    speed=0.0)
zc.vhc3 = Bradley(
    world=zc.world, name="larmor3", side="dwater",
    texture="models/vehicles/bradley/bradley_tex.png",
    pos=pos_from_horiz(zc.vhc1, Point2(0, -20)),
    hpr=hpr_from_horiz(zc.vhc1, Vec3(0, 0, 0)),
    speed=0.0)
zc.vhc4 = Bradley(
    world=zc.world, name="larmor4", side="dwater",
    texture="models/vehicles/bradley/bradley_tex.png",
    pos=pos_from_horiz(zc.vhc1, Point2(20, -20)),
    hpr=hpr_from_horiz(zc.vhc1, Vec3(0, 0, 0)),
    speed=0.0)
zc.vhc5 = Bradley(
    world=zc.world, name="larmor5", side="dwater",
    texture="models/vehicles/bradley/bradley_tex.png",
    pos=pos_from_horiz(zc.vhc1, Point2(0, -40)),
    hpr=hpr_from_horiz(zc.vhc1, Vec3(0, 0, 0)),
    speed=0.0)
zc.vhc6 = Bradley(
    world=zc.world, name="larmor6", side="dwater",
    texture="models/vehicles/bradley/bradley_tex.png",
    pos=pos_from_horiz(zc.vhc1, Point2(20, -40)),
    hpr=hpr_from_horiz(zc.vhc1, Vec3(0, 0, 0)),
    speed=0.0)

# Set navpoints/waypoints.
zc.player.add_waypoint(
    name="pat1", longdes=_("patrol point 1"), shortdes=_("PAT1"),
    pos=zc.pat1pos, radius=2000, height=-1)
zc.player.add_waypoint(
    name="patx", longdes=_("patrol point x"), shortdes=_("PATX"),
    pos=zc.patxpos, radius=500, height=500)
zc.player.add_navpoint(
    name="home", longdes=_("Navpoint 1"), shortdes=_("HOME"),
    pos=Point2(0, 0), radius=640000, height=-1,
    tozone="canary")
zc.player.update_navpoint("home", active=False)

```

Као и обично прво креирамо свет унутар зоне, затим додајемо две зграде, складишта, на одређеним координатама, а онда креирамо играча, и играчева два партнера уколико њихови тс атрибути потврђују да су још увек живи. У наставку додајемо и

неке објекте на замељи. Укупно 6 оклопних возила, постављених у близини два складишта која смо претходно направили. На крају додајемо три путне тачке и једну навигациону тачку, коју одмах и закључавамо. Та навигациона тачка води у претходну зону, а ми смо је закључали из истих разлога о којима смо раније говорили. Да се контекст радње не би случајно распао, ако нисмо добро покрили све могућности шетања између зона.

Још једном треба напоменути да играти на сигурну варијанту, као што смо ми нпр. овде урадили са закључавањем навигационих тачака, није препоручљиво. Јер што се више оваквих ограничења стави пред играча, то играч стиче јачи утисак да га игра води за руку, а то смањује општи осећај игровости игре и имерзију. Дизајнер кампање у принципу треба да балансира између слободе коју играч има у игрању мисија и својих способности да покрије све случајеве те слободе.

Прелазимо на функцију петље:

```
def cr060_wsahara_loop (zc, mc, gc):

    if (not zc.first_conv and
        zc.world.stopwatch("countdown_firstconv") > 6):
        if (not zc.first_conv_triggered and mc.hulk_alive and
            mc.draggon_alive):
            zc.world.chaser = zc.world.player.chaser
            zc.world.player_control_level = 2
            zc.player.ac.set_ap(climbrate=0.0, turnrate=0.0, maxg=9.0)
            zc.dialog01 = conv_cr060_01(zc.world, zc.player,
                                       zc.hulk, zc.draggon)

            zc.dialog01.start()
            zc.first_conv_triggered = True
        elif zc.first_conv_triggered and not zc.dialog01.in_progress():
            zc.world.player_control_level = 0
            zc.player.ac.jump_to(
                pos=Point3(zc.player.ac.pos()[0], zc.player.ac.pos()[1],
                           zc.player.ac.pos()[2]),
                hpr=Vec3(120,0,0), speed=200)
            zc.hulk.jump_to(
                pos=Point3(100000, 100000, 8000),
                hpr=Vec3(270,0,0), speed=190)
            zc.draggon.destroy()
            zc.world.fade_in(0.5)
            zc.first_conv = True

    if not mc.objective_patrol_x and zc.player.at_waypoint("patx"):
        zc.player.show_message("notification", "left",
                               _("Objective complete!"), duration=4.0)
        mc.objective_patrol_x = True

    if (not zc.second_conv and mc.objective_patrol_x and
        zc.world.stopwatch("countdown_secondconv") > 3):
        zc.dialog02 = conv_cr060_02(zc.world, zc.player)
        zc.dialog02.start()
        zc.second_conv = True

    if (not zc.third_conv and zc.second_conv and
        not zc.dialog02.in_progress() and
        zc.world.stopwatch("countdown_thirdconv") > 4):
        zc.world.fade_out(0.5)
```

```

zc.world.break_alliance(["lw", "dwater"])
zc.enemyac5 = F18(
    world=zc.world, name="gray1", side="dwater",
    texture="models/aircraft/f18/f18darkwater_tex.png",
    fuelfill=0.50,
    pos=Point3(500, 520000, 5000),
    hpr=Vec3(180, 0, 0),
    speed=210,
    lnammo=[(None, 4), (Aim9, 2)])
zc.enemyac6 = F18(
    world=zc.world, name="gray2", side="dwater",
    texture="models/aircraft/f18/f18darkwater_tex.png",
    fuelfill=0.50,
    pos=Point3(-500, 510000, 5000),
    hpr=Vec3(180, 0, 0),
    speed=210,
    lnammo=[(None, 4), (Aim9, 2)])
zc.chaser_player = TrackChaser(
    world=zc.world, point=Point3(15, 20, 6),
    relto=zc.player.ac, rotrel=True,
    atref=zc.player.ac, upref=zc.player.ac,
    drift=("instlag", 0.0, 0.25),
    shake=("speed-air", 500.0, 2.0))
zc.chaser_enemyac5 = TrackChaser(
    world=zc.world, point=Point3(-30, 40, -6),
    relto=zc.enemyac5, rotrel=True,
    atref=zc.enemyac5, upref=zc.enemyac5,
    drift=("instlag", 0.0, 0.25),
    shake=("speed-air", 500.0, 2.0))
zc.dialog03 = conv_cr060_03(zc.world, zc.player, zc.enemyac5,
                            zc.enemyac6, zc.chaser_player,
                            zc.chaser_enemyac5)

zc.dialog03.start()
zc.third_conv = True
elif (not zc.begin_fight and zc.third_conv and
      not zc.dialog03.in_progress()):
    zc.world.player_control_level = 0
    zc.player.ac.jump_to(
        pos=Point3(zc.patxpos[0] + 3000, zc.patxpos[1] - 2600, 4005),
        hpr=Vec3(65,0,0),
        speed=220)
    zc.enemyac5.jump_to(
        pos=pos_from_horiz(zc.player.ac, Point3(4000, 12700, 1453)),
        hpr=Vec3(180,0,0),
        speed=240)
    zc.enemyac6.jump_to(
        pos=pos_from_horiz(zc.player.ac, Point3(-3000, 14000, 2019)),
        hpr=Vec3(180,0,0),
        speed=240)
    zc.enemyac5.set_ap(target=zc.player.ac)
    zc.enemyac6.set_ap(target=zc.player.ac)
    zc.chaser_player.destroy()
    zc.chaser_enemyac5.destroy()
    zc.world.fade_in(0.5)
    zc.begin_fight = True

```

```

if (not zc.help_arrived and zc.begin_fight and
    zc.world.stopwatch("countdown_help") > 16):
    formation_pair (zc.player.ac, zc.hulk, compact=6.0, jumpto=True)
    zc.hulk.set_min_fuelfill(0.5)
    zc.hulk.set_auto_attack(["plane"])
    zc.dialog04 = conv_cr060_04(zc.world, zc.player, zc.hulk)
    zc.dialog04.start()
    zc.help_arrived = True

if (not zc.fifth_conv and zc.help_arrived and
    zc.enemyac5.outofbattle and zc.enemyac6.outofbattle):
    zc.world.action_music.set_context("cruising")
    if not zc.hulk.outofbattle:
        zc.dialog05 = conv_cr060_05(zc.world, zc.player, zc.hulk)
        zc.dialog05.start()
    zc.fifth_conv = True

if (not mc.mission_completed and mc.objective_patrol_x and
    zc.fifth_conv and zc.world.stopwatch("countdown_complete") > 8):
    zc.player.update_navpoint("home", active=True)
    mission_completed(zc, mc, gc)

return zc.world, 1.0

```

Ова функција изгледа веома компликовано, али ко разуме систем функционсања скриптовања мисије, заправо из овог кода може директно да чита комплетну радњу мисије. Пошто је ово понављајућа функција петље, све целине радње су подведене под разне упите. Обичај је да се те целине унутар функције петље сортирају према просечно очекиваном редоследу извршавања радње, зарад лакшег праћења догађаја. То сортирање међутим апсолутно не значи да ће се сви догађаји заиста и извршити редоследом којим су целине наведене.

Функција петље почиње са:

```

if (not zc.first_conv and
    zc.world.stopwatch("countdown_firstconv") > 6):
    if (not zc.first_conv_triggered and mc.hulk_alive and
        mc.draggon_alive):
        zc.world.chaser = zc.world.player.chaser
        zc.world.player_control_level = 2
        zc.player.ac.set_ap(climbrate=0.0, turnrate=0.0, maxg=9.0)
        zc.dialog01 = conv_cr060_01(zc.world, zc.player,
            zc.hulk, zc.draggon)

        zc.dialog01.start()
        zc.first_conv_triggered = True
    elif zc.first_conv_triggered and not zc.dialog01.in_progress():
        zc.world.player_control_level = 0
        zc.player.ac.jump_to(
            pos=Point3(zc.player.ac.pos()[0], zc.player.ac.pos()[1],
                zc.player.ac.pos()[2]),
            hpr=Vec3(120, 0, 0), speed=200)
        zc.hulk.jump_to(
            pos=Point3(100000, 100000, 8000),
            hpr=Vec3(270, 0, 0),
            speed=190)
        zc.draggon.destroy()

```

```
zc.world.fade_in(0.5)
zc.first_conv = True
```

Преведено ова целина значи: након 6 секунди од почетка извршавања функције петље зоне, приперми кадар иза завесе, `zc.world.chaser = zc.world.player.chaser`, одузми контролу играчу, `zc.world.player_control_level = 2` (чиме аутоматски улазимо у мод анимације), кажи аутопилоту да исправи нос играчевог авиона тако да лети право, `zc.player.ac.set_ap(climbrate=0.0, turnrate=0.0, maxg=9.0)`, затим окини први дијалог у мисији:

```
zc.dialog01 = conv_cr060_01(zc.world, zc.player,
                           zc.hulk, zc.draggon)
zc.dialog01.start()
```

и на крају затвори трајно улаз у овај подупит са `zc.first_conv_triggered = True`. Што се тиче дијалога, издвојени код представља уобичајени начин да се окине дијалог међу актерима током мисије. Тој функцији дијалога прослеђујемо обавезно `zc.world` (мада смо могли и само `zc`, пошто је `world` већ монтиран на тај објекат), а потом и прослеђујемо све актере дијалога, играча, Хулка и Драгона. Функција дијалога, `conv_cr060_01`, изгледа овако:

```
def conv_cr060_01(world, player, hulk, draggon):

    def fade_out_screen():
        world.fade_out(0.5)

    def player_look_hulk():
        player.headchaser.move_to(atref=hulk,
                                  angspeed=3.0, angacc=1.0)

    def player_look_front():
        player.headchaser.move_to(atref=hprtovec(Vec3(0, 0, 0)),
                                  angspeed=0.5, angacc=0.1)

    return Dialog(
        camnode=world.camera,
        pnode=world.node2d,
        characters={
            "arend": Character(
                width=0.6, pos=(0.15, 0.05), size=10,
                color=rgba(255, 200, 0, 1.0),
                align="c", anchor="bc",
                node=player.ac.node, played=True),
            "hulk": Character(
                width=0.6, pos=(0.15, 0.05), size=10,
                color=rgba(200, 125, 0, 1.0),
                align="c", anchor="bc",
                node=hulk.node),
            "draggon": Character(
                width=0.6, pos=(0.15, 0.05), size=10,
                color=rgba(230, 130, 30, 1.0),
                align="c", anchor="bc",
                node=draggon.node),
        },
        branches={
```

```

    "start": [
        Speech("arend",
            Line_("We are at the first point.")),
        Speech("arend",
            Line_("I am going to leave the two of you now."),
            startf=player_look_hulk)),
        Speech("arend",
            Line_("Be careful out there.")),
        Speech("hulk",
            Line_("*noise*"), startf=player_look_front)),
        Speech("hulk",
            Line_("Right, get on with your mission, boss.")),
        Speech("hulk",
            Line_("We'll be fine.")),
        Pause(time=1.0, startf=fade_out_screen)
    ],
},
wpm speed=150,
)

```

Ликови у дијалогу су дефинисани преко класе Character. У тој класи се могу задати разни арумент за обликовање текста (ширина текста, величина, 2Д позија, боја слова итд), али и чвор објекта за који је тај лик везан. Нпр. Аренд је лик кога контролише играч, па је његов чвор везан за авион играча, node=player.ac.node. Такође, ако је дотични лик и играч, важно је то и нагласити са played=True.

Функција дијалога се извршава паралелно с функцијом петље зоне, а с обзиром на то да се извршава од прве до последње линије дијалога, дијалог је уствари функција слична проточној верзији функције петље. То значи да је функција дијалог заправо идеално место за секвенцијално режирање анимације, тако што се на сваку линију дијалога дода функција која каже шта се у том тренутку, док лик изговара ту линију дијалога, дешава. Функције радње се дефинишу у заглављу функције дијалога:

```

def fade_out_screen():
    world.fade_out(0.5)

def player_look_hulk():
    player.headchaser.move_to(atref=hulk,
                              ang speed=3.0, angacc=1.0)

def player_look_front():
    player.headchaser.move_to(atref=hprtovect(Vec3(0, 0, 0)),
                              ang speed=0.5, angacc=0.1)

```

Нпр. у другој линији дијалога извршава се функција startf=player_look_hulk. Аренд, тј. играч, скренуће поглед из првог лица на Хулка, ма где се његов авион налазио у том тренутку. Да подсетимо, раније смо камеру иза завесе у функцији петље поставили на прво лице са zc.world.chaser = zc.world.player.chaser. Брзине и убзрања, као што су ang speed и angacc, служе да контролишу брзину скретања погледа, а ми смо се нашом комбинацијом вредности трудили да окрет изгледа што природније. До краја овог низа линија дијалога извршиће се и остале дефинисане функције већ према замисли режисера анимације.

Тренутно постоје четири класе линије дијалога са више засебних пропратних параметара:

- Speech, реченица коју задати лик изговара.

- `cond`, задавање услова који мора да буде испуњен да би се та линија дијалога извршила. Ако услов врати `True` линија дијалога ће се извршити, ако врати `False` линија дијалога ће бити прескочена.
 - `branch`, дијалози могу да се гранају, а на овај аргумент може да се монитра нека одређена грана дијалога која ће се покренути након што се заврши тренутна линија.
 - `startf`, задавање функције која креће са извршавањем кад почне линија дијалога.
 - `endf`, задавање функције која креће са извршавањем на крају линије дијалога.
 - `time`, у секундама се задаје колико ће линија дијалога тачно трајати. Ако се то време не зада онда ће линија дијалога трајати према просечној брзини исписивања речи за цео дијалог (у нашем случају та просечна брзина је `wpm=150`).
 - `ctime`, време у секундама после кога почиње наредна линија и ако се претходна није завршила.
 - `ctimefact`, проценат трајања линије дијалога, од 0.0 до 1.0, после кога почиње наредна линија дијалога и ако се претходна није завршила.
 - `voice`, задавање звучног записа на линију дијалога. Ако текст линије дијалога постоји у облику аудио записа, тај запис се може задати на овај аргумент, а онда трајање линије дијалога више не зависи од просечне брзине исписивања речи, него од трајања тог аудио записа.
- **Pause, пауза у дијалогу.**
 - `time`, трајање паузе у секундама.
 - `startf`, задавање функције која креће са извршавањем на почетку паузе.
 - `endf`, задавање функције која креће са извршавањем на крају паузе.
 - **UpdateChar, промена стања неког лика. Нпр. премештање позиције његовог портрета негде друге на екрану.**
 - `speaker`, учесник дијалога кога треба изменити.
 - `charmod`, објекат који садржи жељене измене над ликом, типа `CharMod`.
 - `Entry`, увођење једног или више ликова на позорницу, с листе претходно задатих ликова у дијалогу. Ако није задато, сваки лик ће бити аутоматски уведен у дијалог чим буде изговорио своју прву линију. Пример: `Entry()` (да се сви појаве), `Entry("arend")` или `Entry(["arend", "hulk", "draggon"])`. Ова класа служи пре свега за дијалог на позорници.
 - `Exit`, извођење једног или више ликова са позорнице. Функционише на истом принципу као и увођење. Ова класа служи пре свега за дијалог на позорници.

У наставку почетне целине функције петље, питамо да ли се дијалог 01 претходно окинуо и да ли се завршио:

```
elif zc.first_conv_triggered and not zc.dialog01.in_progress():
```

Ако је услов испуњен, извршавање кода улази у овај други подуслов. Најпре се враћа контрола играчу, `zc.world.player_control_level = 0`, а потом играча и Хулка премештамо на одређене позицију. Хулка намерно премештамо негде далеко изван света зоне да безбрижно лети право, док Драгона, пилота који нам више неће требати нигде у мисији, уклањамо трајно са, `zc.draggon.destroy()`. Треба имати у виду да `.destroy()` **не** поставља, `.shutdown = True`, тако да се објекти могу безбедно

уклањати овим путем. На крају дижемо завесу, `zc.world.fade_in(0.5)`, и трајно затвармо улаз у упит прве целине са, `zc.first_conv = True`.

У наставку радње функције петље, очекује се да ће играч да лети ка главном објективу мисије јер нема ништа друго што би могао да ради, па је то наредна целина коју скрптујемо:

```
if not mc.objective_patrol_x and zc.player.at_waypoint("patx"):
    zc.player.show_message("notification", "left",
        _("Objective complete!"), duration=4.0)
    mc.objective_patrol_x = True
```

Сад користимо онај атрибут `mc.objective_patrol_x` који смо претходно дефинисали у заглављу мисије. Питамо да ли објектив још увек није извршен и да ли се играч налази унутар радијуса путне тачке "patx". Ако је услов задовољен, то значи да је играч извршио главни објектив мисије. Најпре га обавештавамо о позитивном исходу, `zc.player.show_message("notification", "left", _("Objective complete!"), duration=4.0)`. Притом тај објектив тј. атрибут овде користимо и као затварач за упит ове друге целине функције петље.

Радијус и висина путне тачке "patx" били су врло пажљиво постављени:

```
zc.player.add_waypoint(
    name="patx", longdes=_("patrol point x"), shortdes=_("PATX"),
    pos=zc.patxpos, radius=500, height=500)
```

Радијус од свега 500 метара, и висина од свега 500 метара. То значи да би играч могао да уђе у радијус ове путне тачке, морао је да лети врло близу земље, а ако је летео веома близу земље, онда је могао да примети зграде, или што је још важније, оних шест оклопних возила које смо поставили у улазној функцији зоне. Тиме је обавио поенту своје мисије, која је била извиђање. Међутим, у контексту приче те зграде, а посебно та возила, није требало да буду ту. Аренд је очекивао да неће затећи ништа, а ако и затекне нешто, да ће то бити стара совјетска техника коју су разне државе Северне Африке давно куповале. Међутим, затекао је скроз нова америчка оклопна возила бредли, у бојама плаћеничке групе Мрачна вода, који би требало да буду савезници Арендове ескадриле.

У наредној, трећој целини функције петље покрећемо нови дијалог, који је уствари монолог:

```
if (not zc.second_conv and mc.objective_patrol_x and
    zc.world.stopwatch("countdown_secondconv") > 3):
    zc.dialog02 = conv_cr060_02(zc.world, zc.player)
    zc.dialog02.start()
    zc.second_conv = True
```

У `conv_cr060_02`, Аренд говори сам за себе па констатује:

```
def conv_cr060_02(world, player):
    return Dialog(
        camnode=world.camera,
        pnode=world.node2d,
        characters={
            "arend": Character(
                width=0.6, pos=(0.15, 0.05), size=10,
```

```

        color=rgba(255, 200, 0, 1.0),
        align="c", anchor="bc",
        node=player.ac.node, played=True),
    },
    branches={
        "start": [
            Speech("arend",
                Line(_("..."))),
            Speech("arend",
                Line(_("Those are not Warlord's typical "
                    "armor units..."))),
            Speech("arend",
                Line(_("..."))),
            Speech("arend",
                Line(_("Disturbing implications..."))),
        ],
    },
    wpmspeed=150,
)

```

али није свестан шта му се спрема, чиме ће се бавити наредна, четврта целина. Сад креће једна комплексно режирана анимација:

```

if (not zc.third_conv and zc.second_conv and
    not zc.dialog02.in_progress() and
    zc.world.stopwatch("countdown_thirdconv") > 4):
    zc.world.fade_out(0.5)
    zc.world.break_alliance(["lw", "dwater"])
    zc.enemyac5 = F18(
        world=zc.world, name="gray1", side="dwater",
        texture="models/aircraft/f18/f18darkwater_tex.png",
        fuelfill=0.50,
        pos=Point3(500, 520000, 5000),
        hpr=Vec3(180, 0, 0),
        speed=210,
        lnammo=[(None, 4), (Aim9, 2)])
    zc.enemyac6 = F18(
        world=zc.world, name="gray2", side="dwater",
        texture="models/aircraft/f18/f18darkwater_tex.png",
        fuelfill=0.50,
        pos=Point3(-500, 510000, 5000),
        hpr=Vec3(180, 0, 0),
        speed=210,
        lnammo=[(None, 4), (Aim9, 2)])
    zc.chaser_player = TrackChaser(
        world=zc.world, point=Point3(15, 20, 6),
        relto=zc.player.ac, rotrel=True,
        atref=zc.player.ac, upref=zc.player.ac,
        drift=("instlag", 0.0, 0.25),
        shake=("speed-air", 500.0, 2.0))
    zc.chaser_enemyac5 = TrackChaser(
        world=zc.world, point=Point3(-30, 40, -6),
        relto=zc.enemyac5, rotrel=True,
        atref=zc.enemyac5, upref=zc.enemyac5,
        drift=("instlag", 0.0, 0.25),

```



```

        shake=("speed-air", 500.0, 2.0))
zc.dialog03 = conv_cr060_03(zc.world, zc.player, zc.enemyac5,
                            zc.enemyac6, zc.chaser_player,
                            zc.chaser_enemyac5)

zc.dialog03.start()
zc.third_conv = True
elif (not zc.begin_fight and zc.third_conv and
      not zc.dialog03.in_progress()):
zc.world.player_control_level = 0
zc.player.ac.jump_to(
    pos=Point3(zc.patxpos[0] + 3000, zc.patxpos[1] - 2600, 4005),
    hpr=Vec3(65,0,0), speed=220)
zc.enemyac5.jump_to(
    pos=pos_from_horiz(zc.player.ac, Point3(4000, 12700, 1453)),
    hpr=Vec3(180,0,0), speed=240)
zc.enemyac6.jump_to(
    pos=pos_from_horiz(zc.player.ac, Point3(-3000, 14000, 2019)),
    hpr=Vec3(180,0,0), speed=240)
zc.enemyac5.set_ap(target=zc.player.ac)
zc.enemyac6.set_ap(target=zc.player.ac)
zc.chaser_player.destroy()
zc.chaser_enemyac5.destroy()
zc.world.fade_in(0.5)
zc.begin_fight = True

```

У контексту радње, Аренд бива пресретнут од стране два плаћеничка ловца. Главни упит ове целине:

```

if (not zc.third_conv and zc.second_conv and
    not zc.dialog02.in_progress() and
    zc.world.stopwatch("countdown_thirdconv") > 4):

```

састављен је тако да услов никако није могао да буде испуњен док се није претходно одврteo conv_cr060_02. А да би дотични разговор могао да се одврти, претходно је морао да буде испуњен главни објектив мисије. Ако се пажљиво прати ток извршавања кода функције петље, може се посредно закључити да је овај услов довољно добро написан, у смислу да се неће извршити кад му није време. Иначе у овако компликованим условима се врло лако греша, па је мисију увек неопходно подробно истестирати. Што је радња мисије сложенија, то и услови упита обично постају све сложенији како радња одмиче.

Унутар упита, најпре спуштамо завесу, `zc.world.fade_out(0.5)`, затим стварамо два авиона, `zc.enemyac5` и `zc.enemyac6`, а онда и две камере. За наступајућу анимацију неопходно је направити два специјална кадра, па због тога конструишемо те две специјалне камере:

```

zc.chaser_player = TrackChaser(
    world=zc.world, point=Point3(15, 20, 6),
    relto=zc.player.ac, rotrel=True,
    atref=zc.player.ac, upref=zc.player.ac,
    drift=("instlag", 0.0, 0.25),
    shake=("speed-air", 500.0, 2.0))
zc.chaser_enemyac5 = TrackChaser(
    world=zc.world, point=Point3(-30, 40, -6),
    relto=zc.enemyac5, rotrel=True,

```

```
atref=zc.enemyac5, upref=zc.enemyac5,  
drift=("instlag", 0.0, 0.25),  
shake=("speed-air", 500.0, 2.0))
```

Постоје укупно четири класе камера:

- TrackChaser
- SwivelChaser
- ElasticChaser
- PointChaser

У највећем броју случајева, за режирање неке анимације довољна је само једна или више TrackChaser камера, зато ћемо се у овом упутству фокусирати само на ту класу. Класе камера, као и објашњење њихових аргумената, налазе се у фајлу `src/blocks/chaser.py`.

Две направљене камере су заправо врло једноставне, јер су потпуно везане за два објекта. Прва је везана за авион играча, а друга за један од два противничка авиона. Једино што је код камере конкретно намештено јесте тачка у којој камера стоји, `point=Point3(15, 20, 6)` и `point=Point3(-30, 40, -6)`. То што су камере везане за објекте значи да се крећу заједно с њима, а то за шта је камера везана одређено је преко `relto=zc.player.ac` и `relto=zc.enemyac5`. Име аргумента `relto` је скраћено од „релативно у односу на...“. Кадар тј. где камера гледа, одређује аргумент `atref`. Камере из примера гледају у `atref=zc.player.ac` и `atref=zc.enemyac5`. Када је камера везана за објекат онда је и тачка у којој стоји такође везана за објекат, тј. релативна према том објекту. Нпр. `point=Point3(15, 20, 6)` значи 15 метара десно од објекта, 20 метара испред објекта и 6 метара изнад објекта, док `point=Point3(-30, 40, -6)` значи 30 метара лево од објекта, 40 метара испред објекта и 6 метара испод објекта. То су дакле позиције с којих ће камере тј. кадрови гледати у своје задате објекте.

Треба имати у виду да је нос свих модела авиона, возила и бродова (па и зграда, мада код зграда то није много битно), усмерен дуж ипсилон-плус осе њиховог координатног система. Зато ипсилон-оса одређује да ли је нешто испред или иза, и то не само код класа камера, већ и код функција као што су `pos_from_horiz`.

Након што смо креирали камере, отпочињемо нови дијалог, `conv_cr060_03`. Том дијалогу сада шаљемо не само актере, `zc.player`, `zc.enemyac5` и `zc.enemyac6`, већ и две камере које смо направили, `zc.chaser_player` и `zc.chaser_enemyac5`. На крају трајно затварамо улаз у овај упит са `zc.third_conv = True`.

Функција дијалога изгледа овако:

```
def conv_cr060_03(world, player, enemyac5, enemyac6, chaser_player,  
                  chaser_enemyac5):  
  
    def control_level_2():  
        world.player_control_level = 2  
  
    def fade_in_screen():  
        world.action_music.set_context("boss")  
        world.fade_in(0.5)  
  
    def fade_out_screen():  
        world.fade_out(0.5)  
  
    def jump_objects_p1():
```

```

world.chaser = chaser_player
player.ac.jump_to(
    pos=Point3(player.ac.pos()[0], player.ac.pos()[1], 3177),
    hpr=Vec3(30,0,0),
    speed=200)
enemyac5.jump_to(
    pos=pos_from_horiz(player.ac, Point3(5331, 18231, 4910)),
    hpr=hpr_from_horiz(player.ac, Vec3(180,0,0)),
    speed=200)
enemyac6.jump_to(
    pos=pos_from_horiz(enemyac5, Point3(30, -50, 20)),
    hpr=hpr_from_horiz(enemyac5, Vec3(0,0,0)),
    speed=200)

def switch_chaser_player():
    world.chaser = chaser_player

def switch_chaser_enemyac5():
    world.chaser = chaser_enemyac5

def player_set_autopilot():
    player.ac.set_ap(altitude=6000)

def fade_out_screen():
    world.fade_out(0.5)

return Dialog(
    camnode=world.camera,
    pnode=world.node2d,
    characters={
        "arend": Character(
            width=0.6, pos=(0.15, 0.05), size=10,
            color=rgba(255, 200, 0, 1.0),
            align="c", anchor="bc",
            node=player.ac.node, played=True),
        "dwpilotcom": Character(
            width=0.6, pos=(0.15, 0.05), size=10,
            color=rgba(25, 50, 100, 1.0),
            align="c", anchor="tc",
            node=player.ac.node),
        "dwpilot": Character(
            width=0.6, pos=(0.15, 0.05), size=10,
            color=rgba(25, 50, 100, 1.0),
            align="c", anchor="bc",
            node=enemyac5.node),
    },
    branches={
        "start": [
            Pause(time=0.5),
            Pause(time=0.5,
                startf=control_level_2, endf=fade_in_screen),
            Speech("dwpilotcom",
                Line_("Wrong place-- *mild noise* -- to stick "
                    "your nose, Colonel."),
                startf=jump_objects_p1)),
    ]

```

```

    Speech("dwpilotcom",
        Line(_("And now we have to kill you."))),
    Speech("arend",
        Line(_("..."))),
    Speech("arend", [
        Line(_("Why? What are you doing here?"),
            branch="inquisitive"),
        Line(_("You mercs are in league with Tycoon."),
            branch="guessing"),
        Line(_("*remains silent*"),
            branch="silent"),
    ]),
],
"inquisitive": [
    Speech("dwpilotcom",
        Line(_("..."))),
    Speech("dwpilot",
        Line(_("Nothing personal, Colonel."),
            startf=switch_chaser_enemyac5)),
    Speech("dwpilot",
        Line(_("Just orders. Something "
            "I'm sure you understand."))),
    Speech("dwpilot",
        Line(_("Goodbye."), branch="conclusion")),
],
"guessing": [
    Speech("dwpilotcom",
        Line(_("..."))),
    Speech("dwpilot",
        Line(_("You should have followed your orders "
            "closely, Colonel."),
            startf=switch_chaser_enemyac5)),
    Speech("dwpilot",
        Line(_("You Dutch boys, barely comprehend "
            "an inch of this whole affair."))),
    Speech("dwpilot",
        Line(_("Goodbye, Colonel."),
            branch="conclusion")),
],
"silent": [
    Speech("dwpilotcom",
        Line(_("..."))),
    Speech("dwpilot",
        Line(_("Ain't that a pity, Dutch boy?"),
            startf=switch_chaser_enemyac5)),
    Speech("dwpilot",
        Line(_("Goodbye."), branch="beginfight")),
],
"conclusion": [
    Speech("arend",
        Line(_("You have yet to finish your task, scum."),
            startf=switch_chaser_player)),
    Speech("arend",
        Line(_("Come, get me."), branch="beginfight")),
],
"beginfight": [

```

```

        Speech("arend",
            Line(_("*scowls and pulls the stick*"),
                time=3.0, startf=player_set_autopilot)),
        Pause(time=1.0, startf=fade_out_screen),
    ],
},
wpm speed=150,
)

```

Ово је први дијалог у коме играч има избор реченица, и у коме због тога постоји гранање. На почетку дијалога одузимамо контролу играчу, `startf=control_level_2` а онда, преко функције `fade_in_screen`, дижемо завесу коју смо претходно спустили (усто и мењамо позадинску музику). Завеса је трајла укупно 1 секунду. У истом моменту кад смо отпочели дизање завесе, путем претходно дефинисане функције заглавља `startf=jump_objects_p1`, такође смо и променили кадар и преместили све актере у анимацији на своје позиције. Аренда су пресрела два ловца, наводних савезника његове ескадриле, која нису ни најмање задовољна тиме што је пуковник гурнуо нос где му није место, и саопштавају му да ће због тога морати да га одстреле. Избор који играч има у наставку:

```

Speech("arend", [
    Line(_("Why? What are you doing here?"),
        branch="inquisitive"),
    Line(_("You mercs are in league with Tycoon."),
        branch="guessing"),
    Line(_("*remains silent*"),
        branch="silent"),
]),

```

у случају овог дијалога чисто је козметичке природе, јер суштински не одређује ништа и шта год играч изабере води ка истом исходу. Једино што се разликује јесте мало другачија реакција противника на одабрану реченицу играча.

Оно што се може приметити током дијалога јесте да сваки час мењамо кадар, преко функција `switch_chaser_player` и `switch_chaser_enemyac5`. То значи да ће током дијалога у кадру бити час играч, час противник с којим разговара. У последњој линији дијалога, кажемо аутопилоту играчевог авиона да крене да се пење на висину од 6000 метара, `startf=player_set_autopilot`. По истеку те линије, спуштамо завесу, `startf=fade_out_screen`.

До краја четврте целине функције петље, по завршетку дијалога, унутар упита:

```

elif (not zc.begin_fight and zc.third_conv and
      not zc.dialog03.in_progress()):

```

враћамо контролу играчу, премештамо све авионе на своје позиције, задајемо играча као мету противничким авионима, уклањамо две камере које смо направили (јер нам више неће требати) и најзад дижемо завесу. Игра поново почиње.

У начелу, кад год нека камера више није потребна, пожељно ју је почистити са `.destroy()`, јер ако се не почисти она и даље функционише у позадини, непотребно трошећи перформансе.

У петој целини функције петље, 16 секунди од почетка борбе између Аренда и његова два противника, Хулк притиче у помоћ:

```

if (not zc.help_arrived and zc.begin_fight and
    zc.world.stopwatch("countdown_help") > 16):
    formation_pair (zc.player.ac, zc.hulk, compact=6.0, jumpto=True)
    zc.hulk.set_min_fuelfill(0.5)
    zc.hulk.set_auto_attack(["plane"])
    zc.dialog04 = conv_cr060_04(zc.world, zc.player, zc.hulk)
    zc.dialog04.start()
    zc.help_arrived = True

```

Хулк је правилно проценио да ће пуковник, на својој неформалној мисији, упасти у невољу, па га је очигледно пратио. Под овим упитом, најпре премештамо Хулка у формацију с играчем, користећи аргумент формације `jumpto=True`, а потом га и допуњавамо горивом, `zc.hulk.set_min_fuelfill(0.5)`, за случај да му је ниво горива опасно низак, јер је играч, из ма ког разлога, губио време током мисије. Задајемо му да аутоматски бира своје мете, покрећемо нови дијалог, `zc.dialog04`, и на крају затварамо улаз у овај упит, `zc.help_arrived = True`. У четвртом дијалогу, који је тотално базичан, Хулк и Аренд размењују пар кратиких порука:

```

def conv_cr060_04(world, player, hulk):

    return Dialog(
        camnode=world.camera,
        pnode=world.node2d,
        characters={
            "arend": Character(
                width=0.6, pos=(0.15, 0.05), size=10,
                color=rgba(255, 200, 0, 1.0),
                align="c", anchor="bc",
                node=player.ac.node, played=True),
            "hulk": Character(
                width=0.6, pos=(0.15, 0.05), size=10,
                color=rgba(200, 125, 0, 1.0),
                align="c", anchor="bc",
                node=hulk.node),
        },
        branches={
            "start": [
                Speech("hulk",
                    Line_(_("Just-- *mild noise* --in time!")),
                Speech("arend",
                    Line_(_("*surprised* Hulk??")),
                Speech("hulk",
                    Line_(_("Let's nail these bastards-- *mild noise* "
                        "-- Colonel."))),
            ],
        },
        wpm=150,
    )

```

Током скриптовања ове целина, запитали смо се шта би се десило да играч некако успе да обори противничке авионе за мање од 16 секунди, дакле пре него што Хулк пристигне? Могуће је да би се контекст радње у том случају (благо) распао. На крају смо проценили да је апсолутно немогуће да играч обори оба авиона за свега 16 секунде без евенуталне употребе неког варања, па смо ипак одлучили да тај случај не покријемо.

Међутим...

Генерално гледано, овакав тип претпоставки од стране дизајнера мисије високо је ризичан и треба га избегавати кад год је то могуће, путем адекватног скриптовања у коме се покривају све могућности. Оно што се дизајнеру чини да је немогуће, у датом тренутку, може на крају да испадне итекако могуће, те да услед такве једне лоше процене дизајнера, мисија буде избаована.

У шестој целини функције петље, скриптујемо успешан завршетак битке:

```
if (not zc.fifth_conv and zc.help_arrived and
    zc.enemyac5.outofbattle and zc.enemyac6.outofbattle):
    zc.world.action_music.set_context("cruising")
    if not zc.hulk.outofbattle:
        zc.dialog05 = conv_cr060_05(zc.world, zc.player, zc.hulk)
        zc.dialog05.start()
    zc.fifth_conv = True
```

Када су два противничка авиона најзад избачена из битке, најпре мењамо позадинску музику, поново на мирнодопску, а потом окидамо нови дијалог, али само у случају да је Хулк још увек читав:

```
if not zc.hulk.outofbattle:
    zc.dialog05 = conv_cr060_05(zc.world, zc.player, zc.hulk)
    zc.dialog05.start()
```

У том петом дијалогу, видимо да Аренд није ни најмање задовољан Хулковим присуством:

```
def conv_cr060_05(world, player, hulk):

    def hulk_set_ap():
        hulk.set_ap(heading=340, speed=360, useab=True)

    return Dialog(
        camnode=world.camera,
        pnode=world.node2d,
        characters={
            "arend": Character(
                width=0.6, pos=(0.15, 0.05), size=10,
                color=rgba(255, 200, 0, 1.0),
                align="c", anchor="bc",
                node=player.ac.node, played=True),
            "hulk": Character(
                width=0.6, pos=(0.15, 0.05), size=10,
                color=rgba(200, 125, 0, 1.0),
                align="c", anchor="bc",
                node=hulk.node),
        },
        branches={
            "start": [
                Speech("hulk",
                    Line(_("..."))),
                Speech("hulk",
                    Line(_("And that's-- *mild noise* --that."))),
                Speech("arend",
```

```

        Line(_("Hulk! What are you doing here!")),
        Speech("hulk",
            Line(_("I had a nagging feeling-- *noise* --"
                "you would end up in trouble."))),
        Speech("hulk",
            Line(_("*mild noise* --I told Draggon to finish "
                "the patrol, while I come looking for you."))),
        Speech("arend",
            Line(_("And what if they intercept her, now, huh!?"))),
        Speech("arend",
            Line(_("You disobeyed my order, captain!"))),
        Speech("hulk",
            Line(_("I-- but lieutenant can handle herself--"),
                ctimefac=0.8)),
        Speech("arend",
            Line(_("*shouts angry* GET BACK to her, NOW!"))),
        Speech("hulk",
            Line(_("...")), "radiol-c"),
        Speech("hulk",
            Line(_("Understood, boss. Hulk, out."),
                startf=hulk_set_ap)),
    ],
},
wpm speed=150,
)

```

После рибанања које је доживео од стране пуковника, Хулк одлази.

Сад следи последња, седма целина функције петље:

```

if (not mc.mission_completed and mc.objective_patrol_x and
    zc.fifth_conv and zc.world.stopwatch("countdown_complete") > 8):
    zc.player.update_navpoint("home", active=True)
    mission_completed(zc, mc, gc)

```

У овој последњој целини, између осталог, чисто формално проверавамо да ли је главни објектив мисије извршен, `mc.objective_patrol_x`, затим и да ли су испуњени сви остали услови који је требало да буду испуњени до овог момента. Ако јесу, најпре откључавамо навигациону тачку која води назад у зону "sanary", тј. кући, а затим проглашавамо и да је мисија успешно обављена, `mission_completed(zc, mc, gc)`. У питању је још једна функција из фајла `src/blocks/missiontools.py`.

У наставку следи излазна функција зоне "wsahara", у коју се улази оног тренутка када играч покрене аутопилота:

```

def cr060_wsahara_exit (zc, mc, gc):

    zc.visited_before = True

    if zc.player and zc.player.alive:
        store_player_state(mc, zc.player)
        yield zc.world, zone_flyout(zc)

    zc.world.destroy()

```

Као што се може приметити, ништа ново се не дешава у овој функцији.

Овим долазимо до краја скриптовања саме мисије, међутим остаје још -- позорница:

```
# =====  
# Background.  
  
mission_skipmenu = False  
mission_skipconfirm = False  
mission_menumusic = "audio/music/cr-menu.ogg"  
mission_debriefing = "late"  
mission_mustdrink = False  
  
def mission_setbg (mc, gc):  
    pass
```

У кампањи, позорница је место где играч проводи време између мисија. Функција `mission_setbg` је место где се конструише позорница (сценографија, и још што шта). Међутим, у овом тренутку, код за конструисање позорнице је још увек у фази нацрта, а позорница се састоји практично од само једног менија са четири дугмета: „мисија“, „пиће“, „архива“ (дугме које не води никуда за сад) и „излаз“. Зато се овом приликом нећемо бавити конструкцијом позорнице, већ ћемо је једноставно прескочити:

```
def mission_setbg (mc, gc):  
    pass
```

Сад следи објашњење три основна дијалога позорнице:

- `mission_drinkconv` (dc, mc, gc)
- `mission_inconv` (dc, mc, gc)
- `mission_outconv` (dc, mc, gc)

Одмах се може запазити да уместо `zc` позивамо такозвани `dc` објекат. Тај објекат служи да би се у функцијама из заглавља дијалога разни објекти могли преносити између функција.

`mission_drinkconv` је (обично) опциони дијалог, доступан између мисија на позорници, који играч може и не мора да одгледа. Подразумевана намена му је необавезна прича између ликова, у којој се опуштају, размењују утиске, савете и слично. То је дијалог који дизајнер кампање треба да користи пре свега да боље дочара атмосферу кампање и ликове који учествују у причи. У нашем примеру изгледа овако:

```
def mission_drinkconv (dc, mc, gc):  
  
    def bg_snd1_start():  
        dc.sound1 = Sound2D("audio/sounds/_bg-cantina-crowd.ogg",  
                             pnode=dc.node, volume=0.8, loop=True,  
                             play=True)  
        dc.sound2 = Sound2D("audio/sounds/_bg-bar-music-2.ogg",  
                             pnode=dc.node, volume=0.4, loop=True,  
                             play=True)  
  
    return Dialog(  
        pnode=dc.fgnode,  
        characters={  
            "arend": Character(shortdes=_("Arend"),
```

```

        portrait="eagle.png", prtsize=0.6,
        width=1.4, pos=(0.7, 0.6), size=14,
        color=rgba(255, 200, 0, 1.0),
        align="l", anchor="tr",
        unfoldfac=0.5,
        played=True),
    "hulk": Character(shortdes=_("Hulk"),
        portrait="eagle.png", prtsize=0.6,
        width=1.4, pos=(-0.7, -0.2), size=14,
        color=rgba(200, 125, 0, 1.0),
        align="l", anchor="tl",
        unfoldfac=0.5),
    "narrator": Character(width=2.0, pos=(-1.15, 0.80),
        font=FONT_RUS, size=16,
        color=rgba(40, 40, 255, 1.0),
        shcolor=None,
        olcolor=rgba(0, 0, 0, 0.5), olwidth=0.5, olfeather=0.2,
        align="l", anchor="tl", swipe=0,
        unfoldfac=0.0,
        wpspeed=150),
},
branches={
    "start": [
        Speech("narrator",
            Line(_("Las Palmas bar, evening"),
                time=2.0, startf=bg_snd1_start)),
        Pause(time=2.0),
        Entry(["arend", "hulk"]),
        Speech("hulk",
            Line(_("*sneezes while filling his glass*"))),
        Speech("hulk",
            Line(_("Locals are getting on my nerves."))),
        Speech("hulk",
            Line(_("They are constantly snooping around "
                "the airfield."))),
        Speech("hulk",
            Line(_("These civilians think of us as some sort "
                "of a goddamn attraction. *swigs*"))),
        Speech("arend",
            Line(_("I believe that others are enjoying that "
                "attention."))),
        Speech("hulk",
            Line(_("*snorts...* Not I. *...clears his throat*"))),
        Speech("arend",
            Line(_("*leans his elbows at the bar* What is it "
                "really that is bothering you, captain?"))),
        Speech("arend",
            Line(_("Franz?"))),
        Speech("hulk",
            Line(_("Look, Colonel..."))),
        Speech("hulk",
            Line(_("There is no way those MiGs shot at us."))),
        Speech("hulk",
            Line(_("We shot them down long before they had "
                "a chance to do anything."))),
        Speech("hulk",

```

```

        Line(_("*knocks the bar with his finger* "
              "Those Dark Water spooks, they did."))),
    Speech("arend",
          Line(_("I have reviewed you report, twice."))),
    Speech("arend",
          Line(_("You said you hadn't really seen that."))),
    Speech("hulk",
          Line(_("*swigs and shakes his head* "
                "I didn't, but I am sure of it!"))),
    Speech("arend",
          Line(_("You are not offering any solid proof "
                "to your claims, Hulk."))),
    Speech("arend",
          Line(_("Little can be done, about it. "
                "You do understand?"))),
    Speech("hulk",
          Line(_("Just... *lifts his elbow, irritated*"))),
    Speech("hulk",
          Line(_("Just keep the Dark Water away from us, "
                "Arend. Please."))),
    Speech("arend",
          Line(_("*pours drink into a glass*"))),
    Speech("arend",
          Line(_("I am not a fan of them either..."))),
    Speech("arend",
          Line(_("...I will see what I can do."))),
    Speech("arend",
          Line(_("And don't worry about Franz, "
                "we will find him."))),
    Speech("hulk",
          Line(_("*snorts* I am sure we will..."))),
    Speech("hulk",
          Line(_("*swigs* ...unless Warlord's death squad "
                "finds him first, that is."))),
    Pause(time=1.0),
    Exit(),
    Pause(time=1.0),
    ],
    },
)

```

Видимо да намђораста Хулк и његов претпостављени, пуковник Аренд, ван дужности разговарају у неком кафићу у граду Лас Палмас, изван ваздухопловне базе. Хулк делује мало нервозно због скорашњих догађаја, док га Аренд уверава да ће размотрити његове захтеве.

mission_inconv је дијалог увода у следећу мисију. У овом дијалогу је битно објаснити наредну мисију и њене објективе, а наравно може се употребити и за даље напредовање приче у кампањи. Ма какав наративни контекст да је дизајнер одабрао за овај дијалог, увек је битно да играч на крају дијалога има јасну представу шта му је задатак на следећој мисији. Под претпоставком да тај задатак треба да буде јасно одређен, јер нису необичне ни мисије које почињу без јасног циљ. У ту сврху планиран је и додатни алат за сликовито шематизовање корака наступајуће мисије, који међутим још није направљен. У овом дијалогу је такође смислено да и играч доноси неке одлуке, ако такве постоје у контексту приче кампање. У нашем уводном дијалогу мисије:

```

def mission_inconv (dc, mc, gc):

    def bg_snd1_start():
        dc.sound1 = Sound2D("audio/sounds/_bg-empty.ogg",
                            pnode=dc.node, volume=0.8, loop=True,
                            play=True)
        dc.sound2 = Sound2D("audio/sounds/_bg-morning-1.ogg",
                            pnode=dc.node, volume=0.2, loop=True,
                            play=True)

    def set_whiplash_and_hurricane_south():
        mc.whiplash_and_hurricane_south = True

    def set_pyro_andPainter_south():
        mc.pyro_andPainter_south = True

    return Dialog(
        pnode=dc.fgnode,
        characters={
            "arend": Character(shortdes=_("Arend"),
                                portrait="eagle.png", prtsize=0.3,
                                width=0.9, pos=(0.0, 0.45), size=12,
                                color=rgba(255, 200, 0, 1.0),
                                align="l", anchor="tc",
                                unfoldfac=0.5,
                                played=True),
            "draggon": Character(shortdes=_("Draggon"),
                                portrait="eagle.png", prtsize=0.3,
                                width=0.9, pos=(1.15, -0.45), size=12,
                                color=rgba(230, 130, 30, 1.0),
                                align="l", anchor="tc",
                                unfoldfac=0.5),
            "hulk": Character(shortdes=_("Hulk"),
                                portrait="eagle.png", prtsize=0.3,
                                width=0.9, pos=(-0.7, -0.55), size=12,
                                color=rgba(200, 125, 0, 1.0),
                                align="l", anchor="tc",
                                unfoldfac=0.5),
            "whiplash": Character(shortdes=_("Whiplash"),
                                portrait="eagle.png", prtsize=0.3,
                                width=0.9, pos=(-0.25, -0.65), size=12,
                                color=rgba(200, 150, 20, 1.0),
                                align="l", anchor="tc",
                                unfoldfac=0.5),
            "hurricane": Character(shortdes=_("Hurricane"),
                                portrait="eagle.png", prtsize=0.3,
                                width=0.9, pos=(-1.15, -0.45), size=12,
                                color=rgba(225, 210, 25, 1.0),
                                align="l", anchor="tc",
                                unfoldfac=0.5),
            "pyro": Character(shortdes=_("Pyro"),
                                portrait="eagle.png", prtsize=0.3,
                                width=0.9, pos=(0.7, -0.55), size=12,
                                color=rgba(210, 110, 10, 1.0),
                                align="l", anchor="tc",
                                unfoldfac=0.5),

```

```

"painter": Character(shortdes=_("Painter"),
    portrait="eagle.png", prtsize=0.3,
    width=0.9, pos=(0.25, -0.65), size=12,
    color=rgba(255, 175, 40, 1.0),
    align="l", anchor="tc",
    unfoldfac=0.5),
"narrator": Character(width=2.0, pos=(-1.15, 0.80),
    font=FONT_RUS, size=16,
    color=rgba(40, 40, 255, 1.0),
    shcolor=None,
    olcolor=rgba(0, 0, 0, 0.5), olwidth=0.5, olfeather=0.2,
    align="l", anchor="tl", swipe=0,
    unfoldfac=0.0,
    wpspeed=150),
},
branches={
    "start": [
        Speech("narrator",
            Line(_("Las Palmas airbase, early morning"),
                time=2.0, startf=bg_snd1_start)),
        Pause(time=2.0),
        Entry(["arend", "draggon", "hulk", "whiplash",
            "hurricane", "pyro", "painter"]),
        Speech("arend",
            Line(_("Alright boys and girls, pay attention..."))),
        Speech("arend",
            Line(_("This morning, we received the latest "
                "schedule from the command."))),
        Speech("arend",
            Line(_("We will be doing the third shift until "
                "the end of this week."))),
        Speech("painter",
            Line(_("Third shift. Very good."))),
        Speech("arend",
            Line(_("*glances briefly at Painter* "
                "As you all know, captain Pinote has gone "
                "missing on his last mission."))),
        Speech("arend",
            Line(_("Command will give the green light "
                "to the search party soon."))),
        Speech("arend",
            Line(_("And we need to keep the enemy pinned down."))),
        Speech("arend",
            Line(_("Today's mission will be an air patrol."))),
        Speech("arend",
            Line(_("This is an overview of the whole task. "
                "*points at the display*"))),
        Speech("whiplash",
            Line(_("Now that there is seven of us... "
                "who's going to catch a break?"))),
        Speech("arend",
            Line(_("That's sorted, Major. Noone will "
                "be sitting idle."), branch="choice")),
    ],
    "choice": [
        Speech("arend", [

```

```

        Line(_("You and Hurricane will take patrol routes "
            "to the south."), branch="south"),
        Line(_("Northern routes will be your and "
            "Hurricane's task."), branch="north"),
    ]),
],
"south": [
    Speech("arend",
        Line(_("*looks to the left* Pyro and Painter will "
            "take the northern routes."),
            startf=set_whiplash_and_hurricane_south,
            branch="continue")),
],
"north": [
    Speech("arend",
        Line(_("*looks to the left* Pyro and Painter will "
            "take the southern routes."),
            startf=set_pyro_andPainter_south,
            branch="continue")),
],
"continue": [
    Speech("arend",
        Line(_("*focuses his gaze in front* "
            "And me, Hulk, and Draggon will take "
            "central patrol routes."))),
    Speech("draggon",
        Line(_("Formation of three? That's odd...?")),
    Speech("arend",
        Line(_("*nods faintly at Draggon* "
            "It will make more sense, Lieutenant."))),
    Speech("arend",
        Line(_("*eyes everyone* That's it. Questions?))),
    Speech("arend",
        Line(_("...")),
    Speech("arend",
        Line(_("Good."))),
    Speech("arend",
        Line(_("Take your mission instructions... "
            "*motions at the pile of tablets in front* "
            "...and prepare your jets."))),
    Speech("arend",
        Line(_("We are taking off in two hours."))),
    Speech("arend",
        Line(_("Dismissed."))),
    Exit(),
    UpdateChar("arend", CharMod(pos=(-0.6, -0.1))),
    UpdateChar("draggon", CharMod(pos=(0.5, 0.2))),
    UpdateChar("hulk", CharMod(pos=(0.6, -0.3))),
    Pause(time=2.0),
    Speech("arend",
        Line(_("Draggon, Hulk, remain for a moment."))),
    Speech("draggon",
        Line(_("Yes, Colonel?))),
    Speech("hulk",
        Line(_("*faintly absent* What?))),
    Speech("arend",

```

```

        Line(_("*eyes Hulk*")),
        Speech("arend",
            Line(_("When we enter the area of "
                "our patrol routes..."))),
        Speech("arend",
            Line(_("I will be flying with the two of you, "
                "up to the first patrol point."))),
        Speech("arend",
            Line(_("Then..."))),
        Speech("arend",
            Line(_("You and... *glances briefly at Draggon* "
                "...Draggon will finish the rest of the task, "
                "while I will be flying elsewhere."))),
        Speech("hulk",
            Line(_("*rises his eyebrow faintly* Elsewhere...?"))),
        Speech("arend",
            Line(_("Yes, I am going on a reconnaissance."))),
        Speech("arend",
            Line(_("To record a possible area of interest."))),
        Speech("hulk",
            Line(_("I see..."))),
        Speech("draggon",
            Line(_("Understood, Colonel."))),
        Pause(time=1.0),
        Exit(),
        Pause(time=1.0),
    ]
},
)

```

Пуковник Аренд окупљеним пилотима ескадриле објашњава тренутну ситуацију и наредну мисију. У току дијалога, играч добија избор:

```

"choice": [
    Speech("arend", [
        Line(_("You and Hurricane will take patrol routes "
            "to the south."), branch="south"),
        Line(_("Northern routes will be your and "
            "Hurricane's task."), branch="north"),
    ]),
]

```

у које подручје да пошаље два одређена пилота. Последице овог избора ће бити видљиве нешто касније.

mission_outconv је изводни дијалог који почиње по завршетку мисије. Овај дијалог има више намена. С једне стране, може се употребити за детаљно претресање утисака са саме мисије, односно бити уско везан за мисију која се управо завршила. А опет, може се употребити и за неко важно напредовање приче кампање, која нема никакве везе са претходном мисијом. У нашем примеру, бавимо се непосредним утисцима са управо завршене мисије:

```

def mission_outconv (dc, mc, gc):

    def bg_snd1_start():
        dc.sound1 = Sound2D("audio/sounds/_bg-empty.ogg",

```

```

        pnode=dc.node, volume=0.8, loop=True,
        play=True)
dc.sound2 = Sound2D("audio/sounds/_bg-morning-1.ogg",
        pnode=dc.node, volume=0.2, loop=True,
        play=True)

return Dialog(
    pnode=dc.fgnode,
    characters={
        "arend": Character(shortdes=_("Arend"),
            portrait="eagle.png", prtsize=0.6,
            width=1.4, pos=(-0.7, 0.6), size=14,
            color=rgba(255, 200, 0, 1.0),
            align="l", anchor="tl",
            unfoldfac=0.5,
            played=True),
        "whiplash": Character(shortdes=_("Whiplash"),
            portrait="eagle.png", prtsize=0.6,
            width=1.4, pos=(0.7, -0.2), size=14,
            color=rgba(200, 150, 20, 1.0),
            align="l", anchor="tr",
            unfoldfac=0.5),
        "painter": Character(shortdes=_("Painter"),
            portrait="eagle.png", prtsize=0.6,
            width=1.4, pos=(0.7, -0.2), size=14,
            color=rgba(255, 175, 40, 1.0),
            align="l", anchor="tr",
            unfoldfac=0.5),
        "narrator": Character(width=2.0, pos=(-1.15, 0.80),
            font=FONT_RUS, size=16,
            color=rgba(40, 40, 255, 1.0),
            shcolor=None,
            olcolor=rgba(0, 0, 0, 0.5), olwidth=0.5, olfeather=0.2,
            align="l", anchor="tl", swipe=0,
            unfoldfac=0.0,
            wpm speed=150),
    },
    branches={
        "start": [
            Speech("arend",
                Line(_("*climbs down the ladder to the ground*"))),
            Speech("arend",
                Line(_("..."),
                    cond=mc.whiplash_and_hurricane_south,
                    branch="painter")),
            Speech("arend",
                Line(_("..."),
                    cond=mc.pyro_and_painter_south,
                    branch="whiplash")),
        ],
        "painter": [
            Speech("painter",
                Line(_("*paces fast* Colonel, what happened to you!? "
                    "Draggon said--"), ctimefac=0.8)),
            Speech("arend",
                Line(_("*frowned* Shit, that's what happened, "

```



```

        "Major."))),
    Speech("arend",
        Line(_("From now on, Dark Water is to be considered "
            "a hostile force!"))),
    Speech("painter",
        Line(_("Dark Wat-- hostile?"), ctime="cut")),
    Speech("arend",
        Line(_("Status on all groups. Report."))),
    Speech("painter",
        Line(_("*scratches back of the head* "
            "Disturbing news, Colonel."))),
    Speech("painter",
        Line(_("Southern group... both Hurricane and "
            "Whiplash have gone missing."))),
    Speech("arend",
        Line(_("Missing?"))),
    Speech("painter",
        Line(_("Tower confirmed, they were shot down."))),
    Speech("arend",
        Line(_("*palms his face*"))),
    Speech("painter",
        Line(_("Northern group, Pyro and I, returned from "
            "our patrol sortie with little to report."),
            ctime="cut")),
    Speech("arend",
        Line(_("I have to speak with the command, "
            "immediately."))),
    Speech("painter",
        Line(_("*paces off*"))),
    Exit("painter"),
    Pause(time=1.0),
    Exit(),
    Pause(time=1.0),
],
"whiplash": [
    Speech("painter",
        Line(_("*paces fast* Colonel! "
            "What happened out there!?"))),
    Speech("arend",
        Line(_("*frowned* Shit, that's what happened, "
            "Major."))),
    Speech("arend",
        Line(_("From now on, Dark Water is to be considered "
            "a hostile force!"))),
    Speech("painter",
        Line(_("Fuck, Hulk was right."))),
    Speech("arend",
        Line(_("Status on all groups. Report."))),
    Speech("painter",
        Line(_("*frowns* Bad news, Colonel."))),
    Speech("painter",
        Line(_("Both Pyro and Painter have gone missing "
            "on their southern sortie."))),
    Speech("arend",
        Line(_("Missing?"))),
    Speech("painter",

```

```

        Line(_("It appears they were shot down.")),
        Speech("arend",
            Line(_("*palms his face*"))),
        Speech("painter",
            Line(_("Me and Hurricane, we finished our sortie "
                "with little to report."), ctime="cut")),
        Speech("arend",
            Line(_("I have to speak with the command, "
                "immediately."))),
        Speech("painter",
            Line(_("*paces off*"))),
        Exit("painter"),
        Pause(time=1.0),
        Exit(),
        Pause(time=1.0),
    ],
},
)

```

Пуковник Аренд није нимало задовољан оним што му се догодило током мисије. Сад међутим видимо и последице претходног избора у уводном дијалогу мисије. Као што се може приметити, у зависности од тога која су два пилота била отишла да патролирају у јужни сектор, та два ће и нестати.

Што се тиче генералних опција мисије, или заставица како их зовемо, у овом тренутку постоје:

- `mission_skipmenu`, ако је на `True`, прескаче позорницу и све меније позорнице, и одмах води на `mission_inconv`.
- `mission_skipconfirm`, на крају сваког дијалог позорнице играчу су понуђена два дугмета, да се дијалог понови или да настави даље. Ако је ова опција на `True`, не показују се та два дугмета.
- `mission_menumusic`, музика која се пушта у главном менију позорнице.
- `mission_debriefing`, **може имати три стринга као вредност:**
 - `"early"`, извештај о мисији (са пропратном статистиком) појављује се пре `mission_outconv` дијалога.
 - `"late"`, извештај о мисији (са пропратном статистиком) појављује се после `mission_outconv` дијалога.
 - `"skip"`, извештај о мисији (са пропратном статистиком) не појављује се уопште.
- `mission_mustdrink`, тиче се дијалога `mission_drinkconv`. Овај дијалог између мисија је подразумевано опциони, међутим ако дизајнер мисије процени да играч из неког разлога мора да га види пре него што пређе на увод у следећу мисију тј. `mission_inconv`, може поставити ову заставицу на `True`. У том случају уводни дијалог ће бити трајно закључан док играч најпре бар једном не одврти овај опциони.

Важна напомена: број опција или заставица, као и начин конструкције позорнице, нису још увек коначни, и врло је могуће да ће се доста тога мењати у будућности.

Овим закључујемо пример 2. У односу на пример 1, који је покрио основу скриптовања мисије и који ће мање-више остати прецизно упутство, пример 2 је знатно замршенији. Посебно зато што разни елементи још нису саграђени, а за неке, овде описане, постоји

реална могућност да ће бити измењени у будућности. Зато пример 2 треба узети више као опис концепта него као прецизно упутство.

Упутство приредио Стефан Илић <stef@sezampro.rs>