Examen 2019 juin - Corrigé

```
from random import randint
import pygame, sys
from pygame.locals import *

pygame.init()
size = (600, 200)
screen = pygame.display.set_mode(size)
pygame.display.set_caption("Grenouille - XY - 7")
screen.fill(Color("black"))

FPS = 25
clock = pygame.time.Clock()


class Frog:

    def __init__(self, x=300, y=180, color=Color("white")):
        self.x = x
        self.y = y
        self.color = color

    def reset_frog(self):
        self.x = 300
        self.y = 180
        self.color = Color("white")
        print("frog reset done")

    def draw_frog(self):
        pygame.draw.circle(screen, self.color, (self.x, self.y), 3, 1)


class Car:

    def __init__(self, x=0, y=200, width=20, height=10):
        self.x = x
        self.y = y
        self.width = width
        self.height = height
        self.speed = randint(-20, 20)

    def reset_car(self):
        self.x = 0
        self.speed = randint(-20, 20)
        print("car reset done")

    def drive_car(self):
        self.x += self.speed
        if self.x + self.width < 0:
            self.x = 600
        elif self.x > 600:
            self.x = - self.width

    def contains_pixel(self, x, y):
        if self.x <= x <= self.x + self.width and self.y <= y <= self.y +
self.height:
            return True
        return False

    def draw_car(self):
        pygame.draw.rect(screen, Color("yellow"), (self.x, self.y, self.width,
self.height), 1)


class Street:
```

```
    def __init__(self):
        self.list_of_cars = []

    def append_car(self, car):
        self.list_of_cars.append(car)

    def speed_up_cars(self, amount):
        if len(self.list_of_cars) > 0:
            for car in self.list_of_cars:
                if car.speed > 0:
                    car.speed += amount
                elif car.speed < 0:
                    car.speed -= amount
                elif car.speed == 0:
                    for i in randint(0,1):
                        if i == 0:
                            car.speed += amount
                        elif i == 1:
                            car.speed -= amount

    def slow_down_cars(self, amount):
        if len(self.list_of_cars) > 0:
            for car in self.list_of_cars:
                if amount > abs(car.speed):
                    car.speed = 0
                elif car.speed > 0:
                    car.speed -= amount
                elif car.speed < 0:
                    car.speed += amount

    def stop_cars(self):
        if len(self.list_of_cars) > 0:
            for car in self.list_of_cars:
                car.speed = 0

    def drive_cars(self):
        if len(self.list_of_cars) > 0:
            for car in self.list_of_cars:
                car.drive_car()

    def draw_cars(self):
        if len(self.list_of_cars) > 0:
            for car in self.list_of_cars:
                car.draw_car()

    def reset_cars(self):
        if len(self.list_of_cars) > 0:
            for car in self.list_of_cars:
                car.reset_car()

    def check_collision(self, frog):
        if len(self.list_of_cars) > 0:
            for car in self.list_of_cars:
                if car.contains_pixel(frog.x, frog.y - 3) or
car.contains_pixel(frog.x, frog.y + 3) or car.contains_pixel(frog.x + 3, frog.y) or
car.contains_pixel(frog.x - 3, frog.y):
                    return True
        return False


frog = Frog()
street = Street()
street.append_car(Car(0, 40))
street.append_car(Car(0, 80))
street.append_car(Car(0, 120))
```

```
street.append_car(Car(0, 160))

done = False
pushed = KMOD_NONE
game_stopped = False
while not done:
    while not game_stopped:
        for event in pygame.event.get():
            if event.type == QUIT:
                done = True
                game_stopped = True
            if event.type == KEYDOWN:
                pushed = event.key
                if event.key == K_SPACE:
                    screen.fill(Color("black"))
                    frog.reset_frog()
                    street.reset_cars()
            if event.type == KEYUP:
                pushed = KMOD_NONE

        if pushed == K_UP:
            x = frog.x
            y = frog.y
            frog = Frog(x, y - 2)
        if pushed == K_DOWN:
            x = frog.x
            y = frog.y
            frog = Frog(x, y + 2)
        if pushed == K_RIGHT:
            x = frog.x
            y = frog.y
            frog = Frog(x + 2, y)
        if pushed == K_LEFT:
            x = frog.x
            y = frog.y
            frog = Frog(x - 2, y)
        if pushed == K_a:
            street.speed_up_cars(1)
        if pushed == K_s:
            street.slow_down_cars(1)

        if street.check_collision(frog):
            street.stop_cars()
            x = frog.x
            y = frog.y
            frog = Frog(x, y, Color("red"))
            game_stopped = True
        if frog.y - 3 <= 0:
            street.stop_cars()
            x = frog.x
            y = frog.y
            frog = Frog(x, y, Color("green"))
            game_stopped = True

        screen.fill(Color("black"))
        frog.draw_frog()
        street.drive_cars()
        street.draw_cars()
        pygame.display.update()
        clock.tick(FPS)

    for event in pygame.event.get():
            if event.type == QUIT:
                done = True
            if event.type == KEYDOWN:
                pushed = event.key
```

```
                if event.key == K_SPACE:
                    screen.fill(Color("black"))
                    frog.reset_frog()
                    street.reset_cars()
                    game_stopped = False


pygame.quit()
sys.exit()
```