

Smart Home League

جزوه‌ی آموزشی درس‌به‌درس – زیر ۱۹ سال (سکندری)

درس 4

نصب و اجرا

پوشه‌ی Smart Home League که دریافت کرده‌اید دو زیرپوشه دارد: **Windows** و **Mac**.

ویندوز

1. وارد پوشه‌ی **Windows** شوید و روی `Smart Home League.exe` دوبار کلیک کنید.
2. اگر بار اول ویندوز پیام امنیتی داد: **More info** و سپس **Run anyway** را بزنید.
3. پنجره‌ی بازی خودش باز می‌شود – همین!

مک

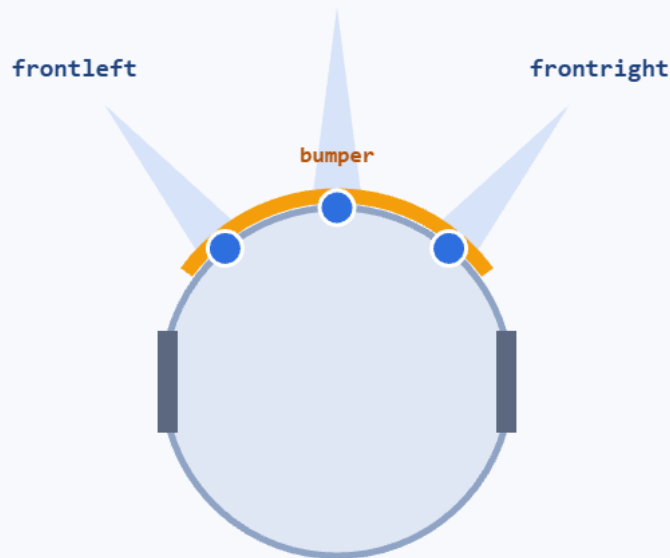
1. وارد پوشه‌ی **Mac** شوید و روی `Smart Home League (Mac).command` دوبار کلیک کنید.
 2. اگر مک اجازه نداد: روی همان فایل **راست‌کلیک** → **Open** → **Open** (فقط بار اول لازم است).
 3. اگر مک پیشنهاد نصب ابزار خط فرمان (python3) داد، **Install** را بزنید و بعد از پایان، دوباره دوبار کلیک کنید.
- در خود برنامه: زبان از منوی ≡ قابل تغییر است؛ «کد پایه»، «هلیپرها» و «حالت مسابقه» روی صفحه‌ی اول هر لیگ هستند و کتابچه‌ی کامل قوانین از دکمه‌ی «قوانین» باز می‌شود.

هلیپر مسیر (U19) – نقطه بزنید، کد تحویل بگیرید

1. از صفحه‌ی اول لیگ، «هلیپر مسیر» را بزنید – عکس واقعی از بالای خانه باز می‌شود.
2. اول در تنظیمات مشخص کنید **ربات شما قرمز است یا آبی** – نقطه‌ی شروع همان ربات روی عکس با برچسب START نشان داده می‌شود.
3. روی نقشه کلیک کنید تا نقطه‌های مسیر ساخته شوند (یا با حالت «کشیدن»، مسیر آزاد بکشید تا خودش به نقطه تبدیل شود).
4. هلیپر خودش نقطه‌های میانی رد شدن از درها را حساب می‌کند – چون `goto` فقط خط مستقیم می‌رود.
5. کنار هر نقطه یک چپ باتری می‌بینید: 🔋 شارژ تقریبی هنگام رسیدن، 🏠 هزینه‌ی راه برگشت تا شارژر – و اگر با تنظیمات فعلی به شارژر نرسد، ⚠ هشدار قرمز می‌گیرید.
6. در جعبه‌ی ⚙ تنظیمات: مسیر حلقه باشد یا یک‌بار؛ گارد باتری روشن/خاموش؛ آستانه‌ی «زیر چند % برو شارژر»؛ و اینکه روی پد «تا چند %» یا «چند ثانیه» بماند.
7. کد ساخته‌شده دو تابع دارد: `follow_route()` برای مسیر و `go_charge()` برای سفر شارژر – پایین فایل صدا زده می‌شوند و اسمشان را هم می‌توانید عوض کنید. اگر پد اشغال باشد، ربات آرام در صف می‌ماند.
8. در پایان: «کپی پایتون»، ⬇ فایل پایتون یا «▶» – با ▶ مسابقه خودش شروع می‌شود.

سنسورهای ربات در یک نگاه

- سه سنسور فاصله‌ی آلتراسونیک (US) – `frontleft` ، `front` (جلو-چپ)، `frontright` (جلو-راست)، بر حسب سانتی‌متر.
- بامپر – `bumperfront` و `bumperback` (۰ یا ۱).
- سنسور رنگ – `color`.
- قطب‌نما – `heading`.
- `x` – GPS و `y` (سانتی‌متر) به همراه `goto(x, y)` ، `distto` و `angleto`.
- باتری و ایستگاه شارژ – `battery` (%) و مختصات پد: `dockx` ، `docky`.



سه سنسور فاصله‌ی آلتراسونیک (US): `front` جلوی ربات، `frontleft` جلو-چپ، `frontright` جلو-راست – و حلقه‌ی نارنجی، بامپر (سپر لمسی) است

پیش از شروع

نکته: برنامه‌ی ربات ۱۰ بار در ثانیه از بالا تا پایین اجرا می‌شود؛ مقداری که در `wheelleft` و `wheelright` می‌گذارید تا اجرای بعدی می‌ماند. هر درس یک برنامه‌ی کامل و قابل اجراست – آن را در بخش «آموزش» داخل برنامه اجرا کنید یا در ادیتور مسابقه بگذارید.

درس ۱ – باتری: سوختِ مسابقه

در U19 هر چرخش چرخ، شارژ می‌سوزاند: **battery** از ۱۰۰ شروع می‌شود و با تمام‌سرعت حدود ۱/۷٪ در ثانیه پایین می‌آید – یعنی ۱۰۰٪ فقط ۶۰۰ ثانیه راندن است. در **صفر**، ربات همان‌جا برای همیشه خاموش می‌شود. این کد فقط می‌راند؛ شما عقبه را تماشا کنید.

```
wheelleft = 25
wheelright = 25

if front < 55:           # فقط دیوارها را رد کن
    wheelleft = 18
    wheelright = -18
elif frontleft < 45:
    wheelleft = 25
    wheelright = 8
elif frontright < 45:
    wheelleft = 8
    wheelright = 25
```

نتیجه‌ی مورد انتظار: پِنل پایین را نگاه کنید: battery پله‌پله پایین می‌آید. با این آهنگ، کل مسابقه‌ی ۳ دقیقه‌ای یعنی دست‌کم دو سفر شارژ. درس بعد: خودِ ایستگاه.

درس ۲ – ایستگاه شارژ و goto

جای ایستگاه ثابت است و به کد شما داده می‌شود: **dockx** و **docky** (سانتی‌متر). **goto(x, y)** خودش می‌چرخد، می‌راند و همان‌جا می‌ایستد؛ **atgoal** که ۱ شد یعنی رسیده‌اید. روی پد، شارژ +۲۵٪ در ثانیه پر می‌شود. توجه داشته باشید: goto مبل و سگ را دور نمی‌زند – مراقب سِپر باشید!

```
if state == 0:           # یک بار در شروع
    state = 1
    goto(dockx, docky)   # برو ایستگاه ->

if bumper == 1:          # goto را دور نمی‌زند
    stopgoto()
    backward(1)
    state = 0            # دوباره تلاش کن

if atgoal == 1:          # رسیدیم: بایست و بنوش
    wheelleft = 0
    wheelright = 0
```

نتیجه‌ی مورد انتظار: ربات رو به ایستگاه می‌چرخد، مسیر مستقیم را می‌راند و روی پد می‌ایستد – و در پِنل، battery شروع می‌کند به بالا رفتن. بعد از برخوردها هم دوباره راه می‌افتد.

درس ۳ – قانون بودجه

کی برگردیم؟ سفر خانه حدوداً $home / 25$ درصد شارژ می‌خواهد ($home$ = فاصله تا داک به سانتی‌متر). قهرمان ۲۲ واحد هم حاشیه می‌گذارد: $needed = home / 25 + 22$ – فرشی کند، دورزدن مبل و برخورد ها را هم حساب کرده. شارژ شما زیر $needed$ رفت؟ همان لحظه راه بیفتید.

```
home = distto(dockx, docky)
needed = home / 25 + 22

wheelleft = 25
wheelright = 25

if state == 1:           # در راه خانه
    if atgoal == 1:
        wheelleft = 0    # روی پد بمان تا پر شوی
        wheelright = 0
        if battery > 95:
            state = 0     # پر شد -> برگرد سر کار
    elif battery < needed: # وقت رفتن است!
        state = 1
        goto(dockx, docky)
    elif front < 55:
        wheelleft = 18
        wheelright = -18
```

نتیجه‌ی مورد انتظار: بگذارید بدود: هر بار که battery به زیر $needed$ می‌رسد خودش راهی خانه می‌شود، تا ۹۵٪ می‌نوشد و برمی‌گردد – چرخه‌ی واقعی مسابقه. حاشیه‌ی ۲۲ را ۵ کنید و ببینید چقدر خطرناک می‌شود!

درس ۴ – ماشین حالت

برنامه‌ی قابل‌اعتماد یعنی ربات همیشه بداند «وسط چه کاری» است. این را با **state** می‌نویسیم: 0 تمیزکاری، 1 در راه خانه، 2 روی پد. هر حالت فقط کار خودش را می‌کند و فقط در شرایط مشخص به حالت بعد می‌رود – این استخوان‌بندی کد کامل قهرمان U19 است.

```
home = distto(dockx, docky)

if state == 2:          # حالت ۲: نوشیدن
    wheelleft = 0
    wheelright = 0
    if battery > 95:
        state = 0

elif state == 1:        # حالت ۱: در راه خانه
    if bumper == 1:
        stopgoto()
        backward(1)
        goto(dockx, docky)
    elif home < 25:      # رسیدیم
        stopgoto()
        state = 2

else:                   # حالت ۰: تمیزکاری
    wheelleft = 25
    wheelright = 25
    if battery < home / 25 + 22:
        state = 1
        goto(dockx, docky)
    elif front < 55:
        wheelleft = 18
        wheelright = -18
```

نتیجه‌ی مورد انتظار: همان چرخه‌ی درس ۳، ولی این بار «هر حالت یک بلوک» – آن را بخوانید: هر بلوک فقط یک کار می‌کند. حالا کد کامل «🏆 قهرمان U19» را از منوی ربات‌ها باز کنید: همین ماشین است + مأموریت مخزن + فرار هوشمند.