**LAPORAN PRAKTIKUM HURDLE 1,2,3,4 & MAZE**

****

Disusun oleh:

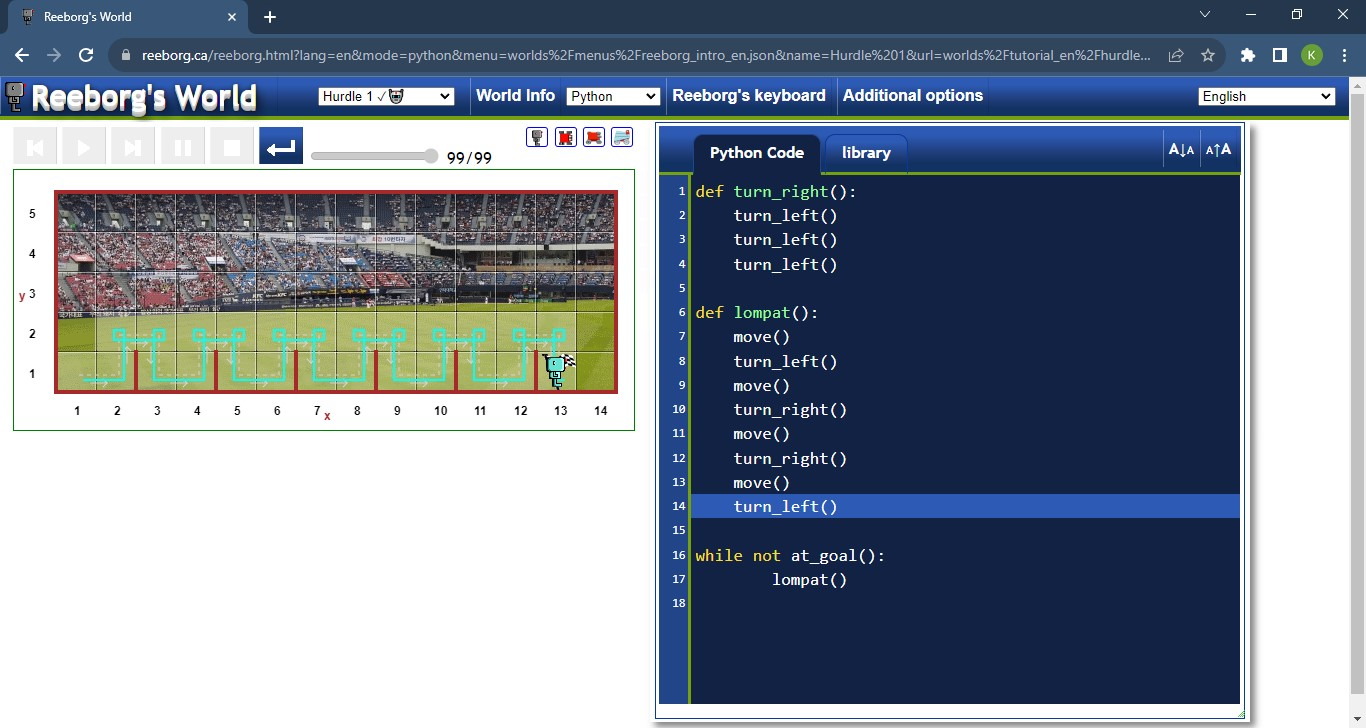
ADZKIA MIFTACHUZZAKIY / 23091397073

KUKUH ADJI PRAMUKTILAKSONO / 23091397103

M. SYAFIUL MASRURI / 23091397085

**PROGRAM STUDI D4 MANAJEMEN INFORMATIKA PROGRAM VOKASI UNIVERSITAS NEGERI SURABAYA 2023**

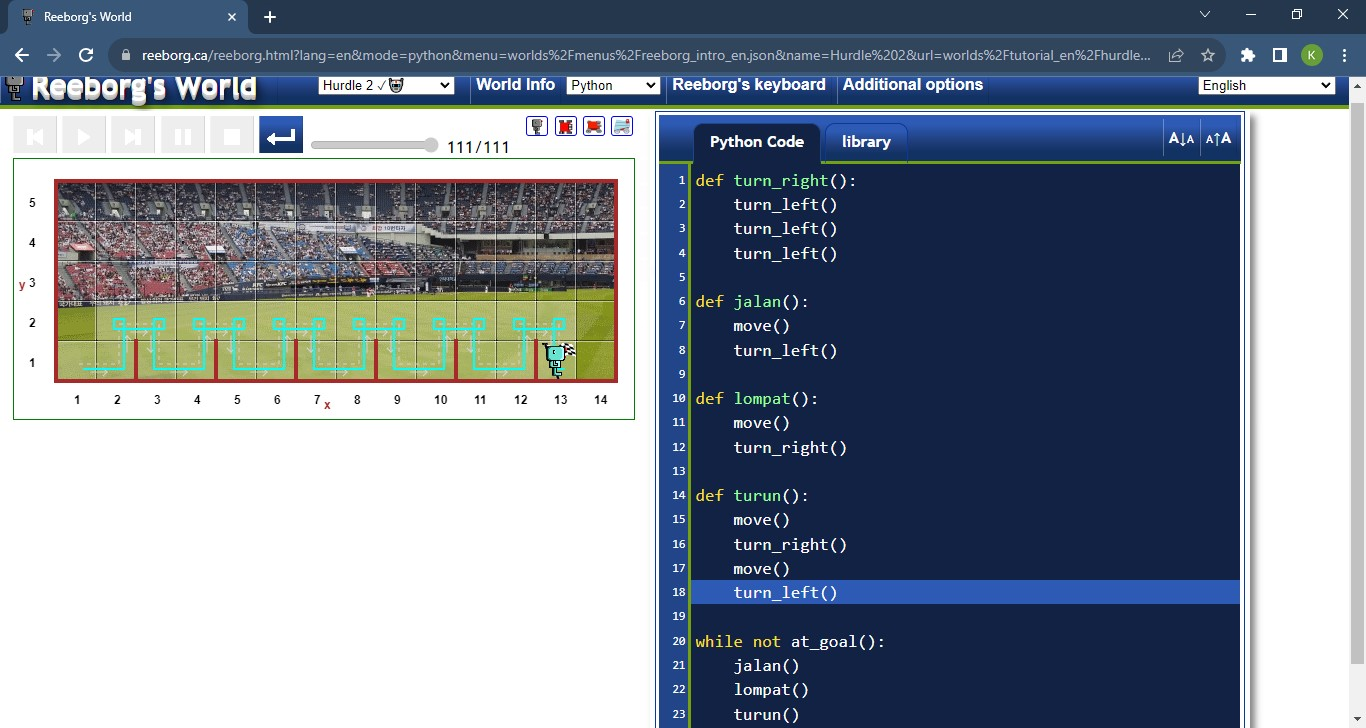
**HURDLE 1**



Penjelasan:

* Baris pertama mendefinisikan perintah untuk hadap kanan.
* Baris kedua sampai keempat berisi kode turn\_left() sebanyak 3 kali agar robot dapat menghadap kanan.
* Baris ke lima mendefinisikan lompat
* Baris ketujuh sampai empat belas berisi perintah agar Karell bisa berjalan sampai finish. Terdapat kode move() untuk melangkah, turn\_left() untuk menghadap kiri, dan fungsi turn\_right() yang berisi kode turn\_left() tiga kali untuk menghadap kanan.
* Baris keenam belas berfungsi agar Karell bisa bergerak hingga sampai di garis finish baru berhenti.

**HURDLE 2**



Penjelasan:

* Baris pertama sampai ke empat mendefinisikan fungsi untuk hadap kanan, yang berisi turn\_left() sebanyak tiga kali
* Baris ke enam mendefinisikan fungsi untuk jalan
* Baris ke tujuh dan delapan berisi kode untuk fungsi jalan yakni move() dan turn\_left()
* Baris ke sepuluh mendefinisikan fungsi untuk lompat
* Baris ke sebelas dan dua belas merupakan kode dari definisi lompat, berisi move() untuk bergerak, turn\_right() agar Karell menghadap kanan setelah bergerak
* Baris ke empat belas mendefinisikan fungsi turun
* Baris ke lima belas sampai delapan belas merupakan kode untuk fungsi turun, yang berisi move() untuk melangkah, turn\_right() untuk menghadap kanan, dan turn\_left() untuk menghadap kiri
* Baris ke dua puluh sampai dua puluh tiga adalah looping agar Ketika belum sampai di finish Karell melakukan fungsi jalan(), lompat(), dan turun() sampai pada garis finish dan berhenti

**HURLDE 3**

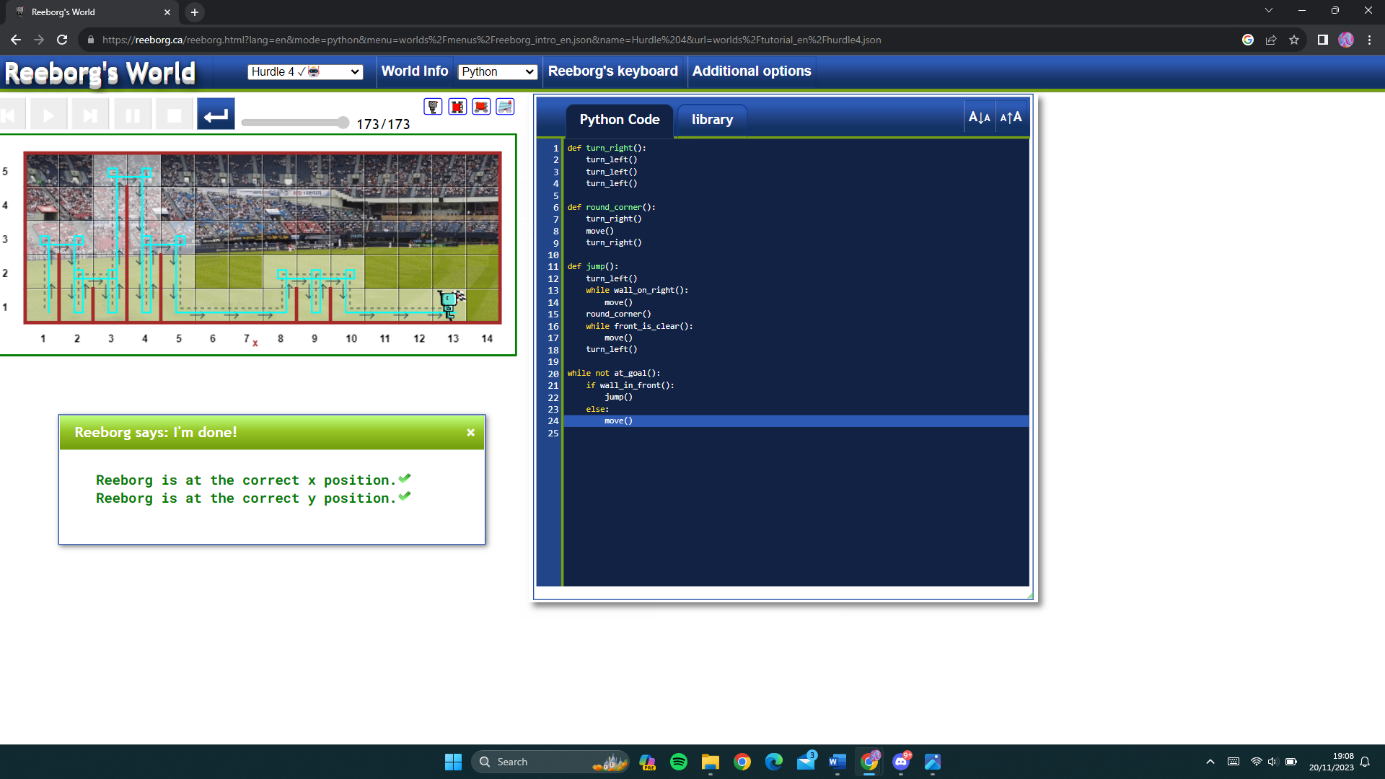
**A computer screen shot of a computer screen

Description automatically generated**

Penjelasan:

* Baris pertama mendefinisikan fungsi turn\_right untuk menghadap kanan
* Baris ke dua sampai empat berisi kode turn\_left() sebanyak tiga kali agar Karell bisa menghadap kanan
* Baris ke enam mendefinisikan fungsi loncat()
* Baris ke tujuh sampai tiga belas berisi kode untuk definisi fungsi loncat(), berisi kode turn\_left() untuk menghadap kiri, move() untuk melangkah, turn\_right() untuk menghadap kanan. Urutan kode seperti pada gambar
* Baris ke lima belas merupakan looping jika Karell belum mencapai finish
* Baris ke enam belas sampai 19 merupakan kode perintah yang akan dilakukan Karell jika belum mencapai garis finish. If wall\_in\_front() berfungsi jika ada tembok di depan maka karell akan melaksanakan kode loncat(), else move() berarti selain itu karel akan melaksanakan kode move()

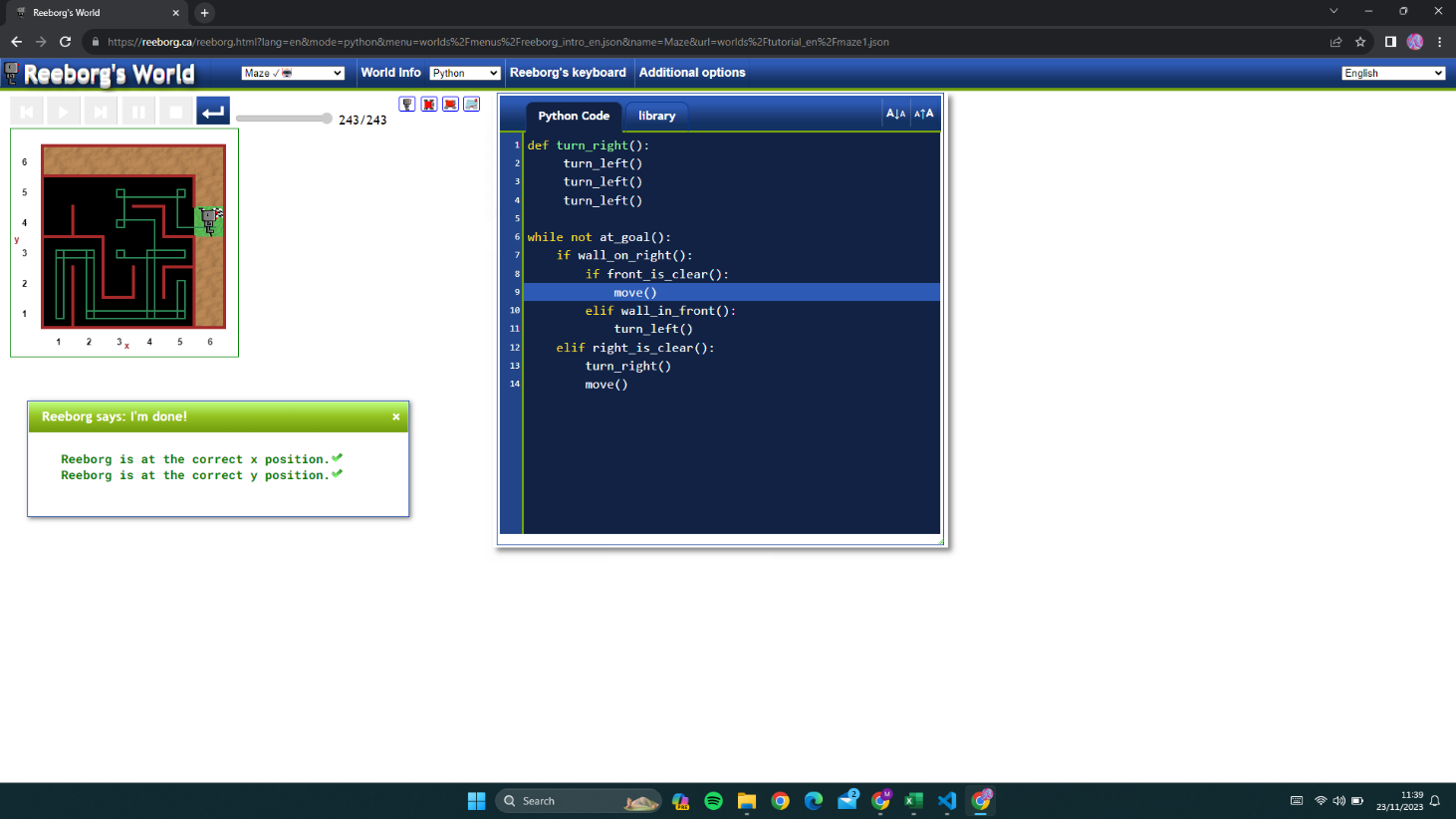
**HURDLE 4**

****

Penjelasan:

* Baris pertama sampai ke empat mendefinisikan fungsi untuk hadap kanan, yang berisi turn\_left() sebanyak tiga kali
* Baris enam sampai ke sembilan mendefinisikan fungsi untuk menavigasi belok ke kanan turn\_right() yang berisi turn\_left() tiga kali, setelah itu move() atau maju ke depan dan yang terakhir akan melakukan turn\_right() yang berisi turn\_left() sebanyak tiga kali
* Baris sebelas sampai delapan belas mendefinisikan kondisi untuk menghadap ke kiri yaitu turn\_left(), ketika terdapat tembok di depan maka akan menjalankan perintah move() jalan dan setelah itu akan melanjutkan function round\_corner(): yang berisi turn\_right() move() turn\_right(), dan selanjutnya ketika di depan kosong maka ankan menjalankan perintah move() jalan dan setelah itu menghadap ke kiri turn\_left()
* Baris dua puluh sampai dua puluh empat mendefinisikan kondisi jika belum mencapai finish akan menjalankan perintah if wall\_in\_front(): maka menjalankan function jump() dan selain itu false akan melakukan move()

**MAZE**

****

Penjelasan:

* Baris pertama sampai ke empat mendefinisikan fungsi untuk hadap kanan, yang berisi turn\_left() sebanyak tiga kali
* Baris enam sampai empat belas mendefinisikan kondisi jika belum mencapai finish akan melakukan perintah if wall\_on\_right(): if front\_is\_clear(): yang berisi move() atau jalan, dan jika ternyata kedua if tersebut tidak melakukan perintah dengan kata lain tidak memenuhi maka akan beralih ke elif wall\_in\_front(): yang berisi turn\_left(), dan jika lain lagi maka menjalankan perintah elif right\_is\_clear(): yaitu ada function turn\_right() yang berisi turn\_left() 3 kali dan move()

**LINK GITHUB TUGAS 2 KELOMPOK 3**

<https://github.com/23091397103-KukuhAdjiPramuktilaksono/Tugas-Alpro-Hurdle.com>