

SWEST23 9/3(金) 11:30～12:40 セッションs3b

<https://swest.toppers.jp/SWEST23/program/s3b.html#s3>



ETロボコンシミュレータの開発

ETロボコン東京地区 実行委員長
新 吉高（あたらし よしたか）

1.1 講演者紹介

- 2015年よりETロボコン実行委員会に東京地区実行委員として参加
- 2017年東京地区実行副委員長を経て、2018年より東京地区実行委員長
- 2020年大会で用いられたETロボコンシミュレータ(Unity)の設計・実装を担当
- 本業は某企業で組み込みソフトからクラウド上でのデータ分析まで薄く広く対応するエンジニア兼マネージャ。
- 技術士(情報工学部門)



東京地区大会懇親会後の集合写真

1.2 ETRobotとは

Embedded Technology Software Design Robot Contestの通称で、LEGO社のLEGO Mindstormsを使用した自律型ライトレースロボットの設計とタイムを競うコンテストです。

総合部門

モデル部門

システム分析、
ソフトウェア設計モデル

+

競技部門

走行競技による性能



HackEV

1.3 ETOボコン 2021 プロモーションビデオ

https://www.youtube.com/watch?v=W4U41Jk_b38



2.1 コロナ禍でのシミュレータ開発の決断

- 2020年3月まではリアル大会を実施するつもりで動いていた。
- コロナの状況が悪化したため、4/12の全国連絡会議で中止の連絡があるのだろうと思っていたが、シミュレータ大会を開催すると連絡があった。
- 救世主はTOPPERS/箱庭WGだった。
- 彼らは組込みCPUシミュレータAthrillを開発し、これと連携するUnity側の単体ロボットシミュレータを開発していた。たしかに、これが使い物になるならシミュレータ大会はできなくはない。

「でも、どう見積もっても10人月くらいかかるのだが…」

2.2 要求分析

- 運営の要求： 学びの歩みを止めない！
 - シミュレータ大会を実施すること。
 - 箱庭WGの成果物を活用すること。
 - ライントレースのみの簡易なものでも十分。
 - 競技判定は目視でもよい。
- 箱庭WGの要求： 成果物を多くの参加者にアピールしたい！
 - Athrillと単体ロボットシミュレータを大会で利用すること。
 - 箱庭WGにHackEV走行体の3Dモデルを提供すること。
 - ETロボコン大会のために十分な開発時間を用意することはできない。
- 新の狙い： リアル大会の雰囲気やシミュレータ大会でも再現したい！
 - エントリー、プライマリー、アドバンスの3クラスをすべてサポートする。
 - 音や画面以外のHackEV走行体の動作をすべてAPIで制御できるようにする。
 - Unityの物理エンジンで作り切る。
 - 競技判定はシミュレータが行う。

2.3 要求分析（コンフリクトの理解と解消）

- 運営の要求： 学びの歩みを止めない！
 - シミュレータ大会を実施すること。→ **OK**
 - 箱庭WGの成果物を活用すること。→ **OK**
 - ライントレースのみの簡易なものでも十分。→ **NG**
 - 競技判定は目視でもよい。→ **NG**
- 箱庭WGの要求： 成果物を多くの参加者にアピールしたい！
 - Athrillと単体ロボットシミュレータを大会で利用すること。→ **Athrillのみ**
 - 箱庭WGにHackEV走行体の3Dモデルを提供すること。→ **OK**
 - 十分な開発時間を用意することはできない。→ **最低限できることは何か？**
- 新の狙い： リアル大会の雰囲気やシミュレータ大会でも再現したい！
 - エントリー、プライマリー、アドバンスの3クラスをすべてサポートする。
 - 音や画面以外のHackEV走行体の動作をすべてAPIで制御できるようにする。
 - Unityの物理エンジンで作り切る。
 - 競技判定はシミュレータが行う。

2.4 箱庭との疎結合によるコンフリクトの解消

- Athrillとシミュレータの間の通信仕様を1024バイトのUDP通信と決定し、ETロボコン実行委員会とTOPPERS/箱庭WGが独立で開発できる状態を維持することを合意した。

参加チーム開発

走行体制御ソフトウェア

プログラミング言語：C/C++

新が開発した部分

ETロボコンシミュレータ
(参加者限定)



棚橋先生の開発

etrobo環境

TOPPERSの成果物

EV3RT/ASP3

エミュレータ向け移植版

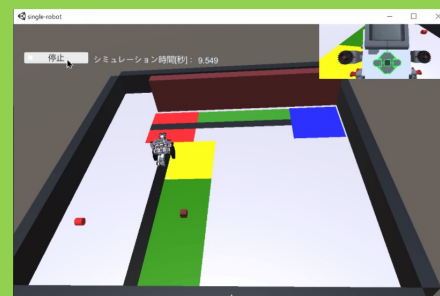
箱庭WGの成果物 + 土樋さんの改善

Athrill

CPU エミュレータ

箱庭WGの成果物

単体ロボット向け
シミュレータ(一般公開)



同期・通信(UDPのみ)

ETロボコンの独自拡張あり

同期・通信(UDP, ROS ...)

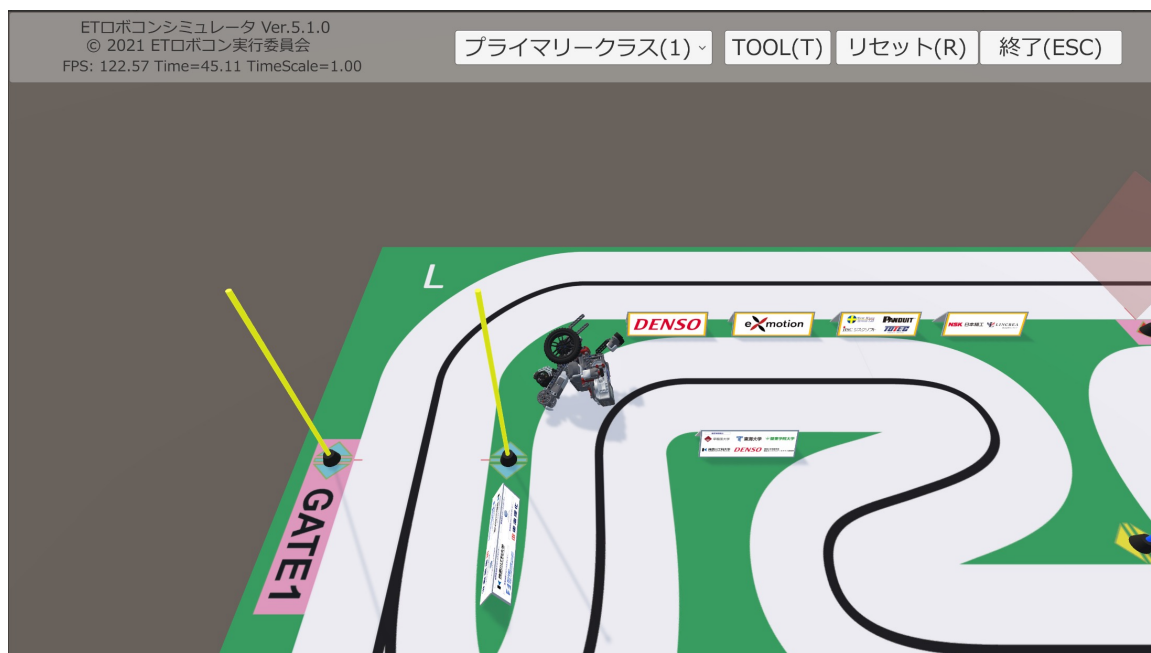
箱庭WGの独自拡張あり

<https://github.com/ETRobocon/etrobo>

参照: <https://toppers.github.io/hakoniwa/prototypes/single-robot/>

2.5 技術課題：走行体調整

- 本物のHackEVの重心は割と高い。
 - そのためシミュレータ上で全力で走るとカーブを曲がれず転倒する。（下図）
 - 重心を低くすると安定するが、HackEVらしい挙動が再現できなくなる。（難所攻略の再現性が低くなる）
- Unityのパラメータ調整だけでは解決できず、速度に応じて重心を下げるスクリプトで解決。



2.6 技術課題：アーム調整

- ブロックを掴むときにはアームをばねで広げる必要がある。
 - ばねが強すぎると掴めない。弱すぎると運べない。
 - 摩擦が強すぎると動かない。弱すぎるとすっぽ抜ける。
- Unityのパラメータ調整だけでは解決できず、ブロック保持スクリプトで解決。



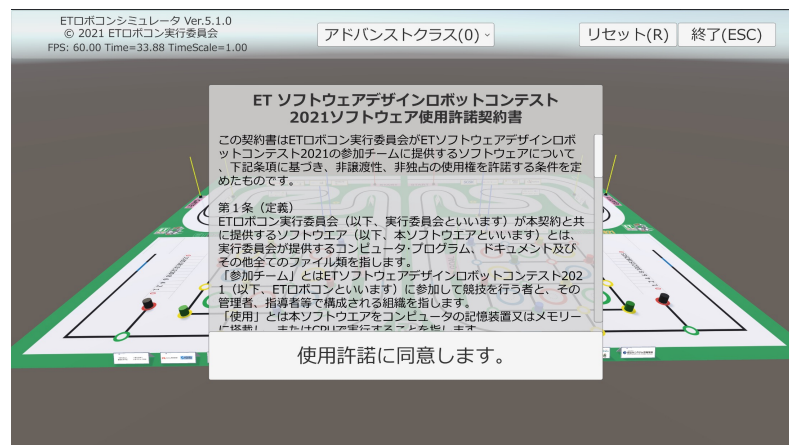
ブロック保持時のアーム



パラメータ調整で何とかしようとしていた頃の調整依頼書

2.7 参加者へのv1.0リリース (4/27~7/6)

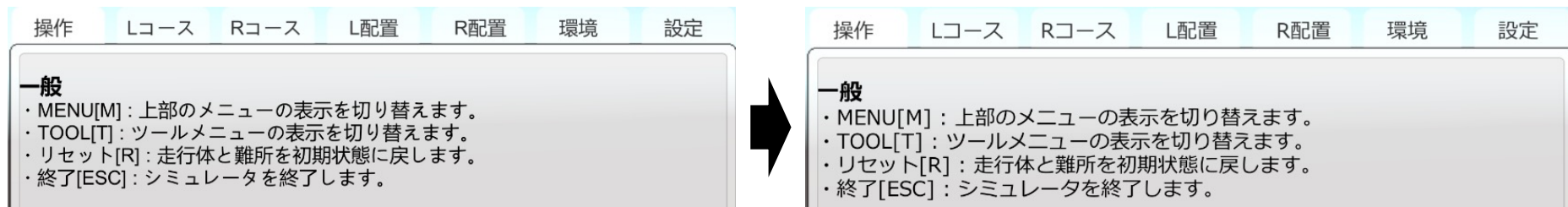
- 3クラスの難所の実装完了
- 三角柱（スポンサーロゴ）対応
 - 定期的にロゴが切り替わる仕組み
- 使用許諾画面を実装
- 7/6 無事、v1.0をリリース！
- ギリギリ技術教育に間に合った。



7月-8月	9/16-22	10/9	10/17-10/25	10/26 23:59まで	11/13	11/22	11/23
オンライン 技術教育	試走会 1	モデル 提出	試走会2	プログラム 提出	CS大会 選出発表	CS大会 (競技会)	モデリング ワークショップ

2.8 試走会向けv2.0の開発(7/7~8/3)

- 日本語フォントの見栄えを変更した。



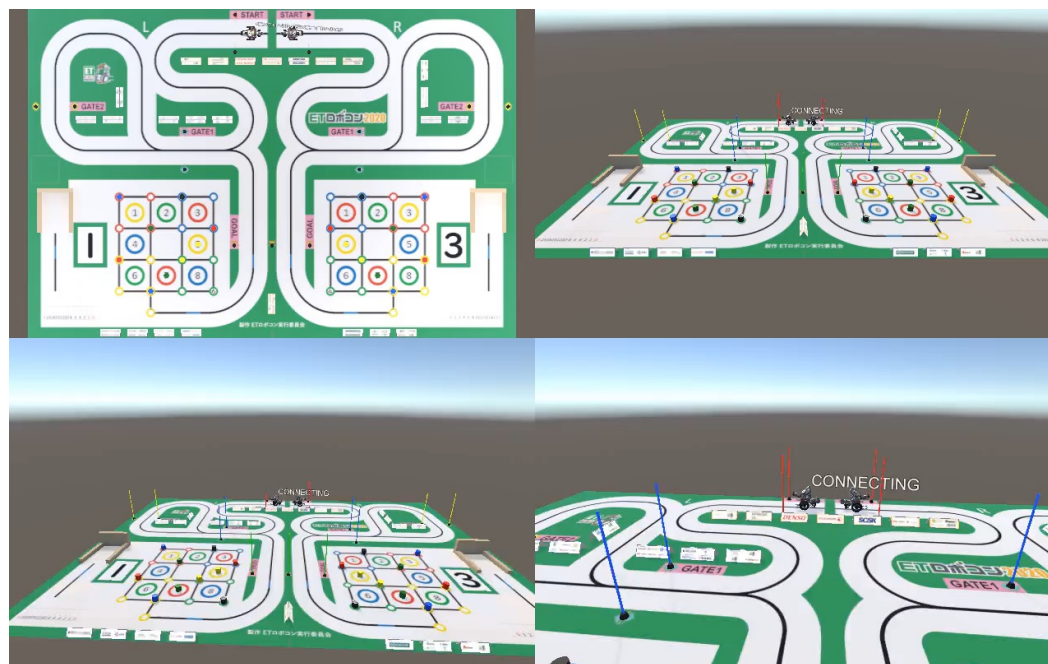
- 小田島さんがAzure上での動作確認をした。
 - [Azure VM で ET ロボコン EV3/シミュレータ開発環境を構築しよう \(準備編\) - Qiita](#)
- 走行体の初期位置を変更できるようにした。
 - 数値変更後に走行体をリスポーン
 - コース選択機能
- 8/3 v2.0をリリースした！



7月-8月	9/16-22	10/9	10/17-10/25	10/26 23:59まで	11/13	11/22	11/23
オンライン 技術教育	試走会 1	モデル 提出	試走会2	プログラム 提出	CS大会 選出発表	CS大会 (競技会)	モデリング ワークショップ

2.9 技術課題： 画面キャプチャ機能

- 4画面構成を決定。
 - GPUを使わず、メインスレッドで画像を取り込んでいたため2分の走行をキャプチャするのに5分かかっていた。
- 高速化への道： 余計な処理をメインスレッドから追い出す。
 - AsyncGPUReadbackを使って画面取り込み処理を並列化
 - PNG生成処理を並列化。これで、2フレーム遅延でのほぼリアルタイムキャプチャに成功。



2.10 試走会向けv3.0の開発(8/3～9/3)

- 画面キャプチャ機能
 - 15～60fpsで画面キャプチャしながら、競技結果をcsvに書き出す機能
 - これがなければ試走会も大会も成立しない！
- 計測モードを追加した。
 - 計測準備： 走行準備ができたことをシミュレータに通知する。
 - 計測開始： 3, 2, 1, GO! で自動的に走行体のタッチボタンが押される。走行体はタッチボタンを検知してスタートする。
 - 計測終了： 2分間経過する前に強制的に計測を停止する。
 - タッチボタン押下からゴールまでの時間計測
- その他、夏休みを利用して膨大な不具合修正を行った。
 - ジャイロセンサの調整とか、ブロック配置の読み込みミスとか…
- 9/3 v3.0をリリースした！ 試走会1に間に合った。

7月-8月	9/16-22	10/9	10/17-10/25	10/26 23:59まで	11/13	11/22	11/23
オンライン 技術教育	試走会 1	モデル 提出	試走会2	プログラム 提出	CS大会 選出発表	CS大会 (競技会)	モデリング ワークショップ

2.11 技術課題： スラローム板上の判定

- Unityは凸多角形のコライダーのみサポートしている。
- スラローム板のコライダーは2つのボックス型コライダーの組み合わせで作られている。
- コライダーの境界において、2つの不具合が報告された。
 - 走行体が1%程度の確率で小ジャンプする
 - ペットボトルの転倒判定を間違えることがある



2.12 大会向けv4.0の開発(9/4~10/11)

- ペットボトルの転倒判定の不具合を修正。
 - 当初はz軸とペットボトルの軸の内積を取って判定していた。しかし、ペットボトルが走行体の擦りによって回転した場合も転倒と判定されてしまった。
 - 検討の結果、ペットボトルの蓋とスラローム板との接触判定を用いることでほぼ解決した。
- ガレージ挑戦中に2分を超えた場合に判定を継続できるように修正。

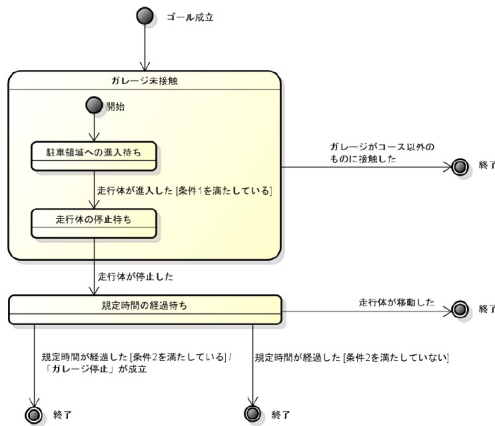


図 5-7 stm: ガレージ停止の判定方法

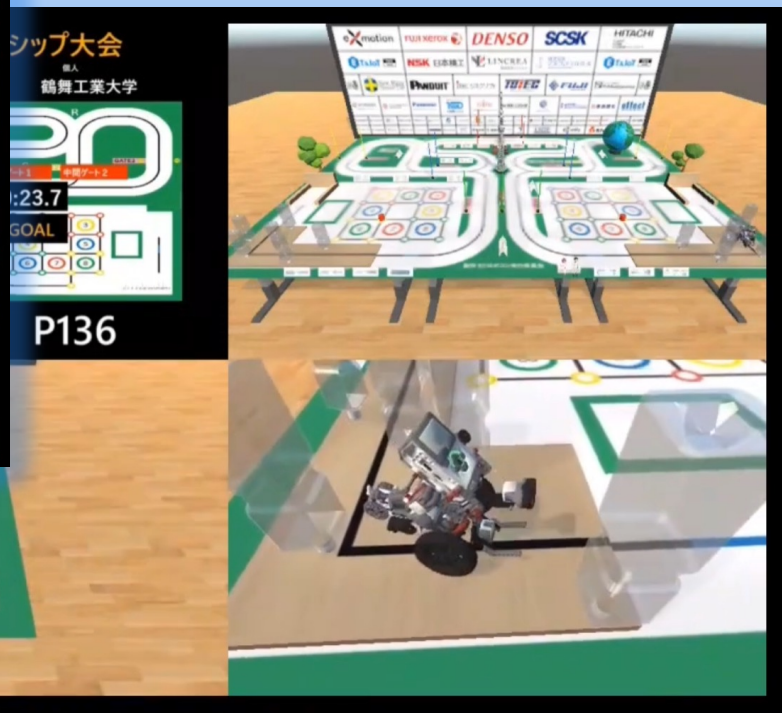


- 10/11 v4.0をリリースした！

7月-8月	9/16-22	10/9	10/17-10/25	10/26 23:59まで	11/13	11/22	11/23
オンライン 技術教育	試走会 1	モデル 提出	試走会2	プログラム 提出	CS大会 選出発表	CS大会 (競技会)	モデリング ワークショップ

2.13 CS大会

伝統的に社会人が強かったETロボコンだが、2020年は学生が活躍した！



3.1 ETロボコンシミュレータの課題と解決案

多くの制約の中で開発したため、様々な課題が残っている。

ETロボコンシミュレータの課題	解決案
複数OS対応によりインストーラが複雑化し、バージョンアップ毎に不具合報告がある (ruby、M1 Mac、WSL関係が多い)	WebGLによるOS非依存化、インストールレス環境の実現
参加者メリットの維持のため、シミュレータのソースコードを公開できない	ソースコードを公開可能なバージョンを別に開発する。
マイコンシミュレータAthrillとのUDP通信周期を短くできず、倒立振子制御の実現が難しい	Athrillの代わりに軽量なスクリプト言語を用いる（この場合、EV3用コード生成のメリットは無くなる）
JSONファイルでコースをカスタマイズできるものの、難所の配置の自由度が低い	スクリプト言語で自由に配置できるようにする

3.2 WebGL版ETロボコンシミュレータの開発

- WebGLによるOS非依存化、インストールレス環境の実現
- Athrillの代わりに軽量なスクリプト言語(MiniScript)を用いる
⇒ SWEST23参加者向けに作ってみた。

<https://webglsim.etrobo.jp>

API一覧

操作	機能
カーソルキー	走行体を動かす (スクリプト非実行時)
SHIFTキー	アームを上げる (スクリプト非実行時)
ALTキー	しっぽを伸ばす (スクリプト非実行時)
Vキー	カメラを走行体追従モード (正面⇄側面) にする
Cキー	カメラをマウス操作モード⇒パノラマモード⇒コクピットモードに変更する
カンマ(,)キー	走行体の位置を変更する

MiniScript仕様

Aceエディタ
(MiniScript用)

スクリプト実行

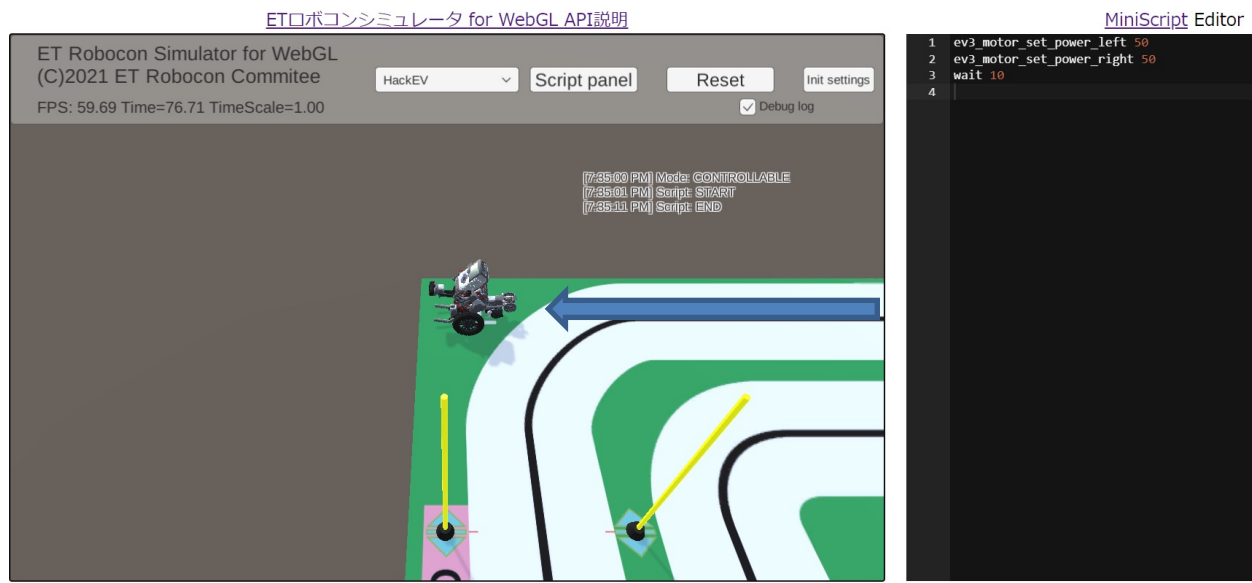
シミュレータ操作

練習1: HackEV(ロボット) を動かそう

- ① WebGL版ETロボコンシミュレータをブラウザで起動する。
- ② エディタに下記プログラムをコピーする。

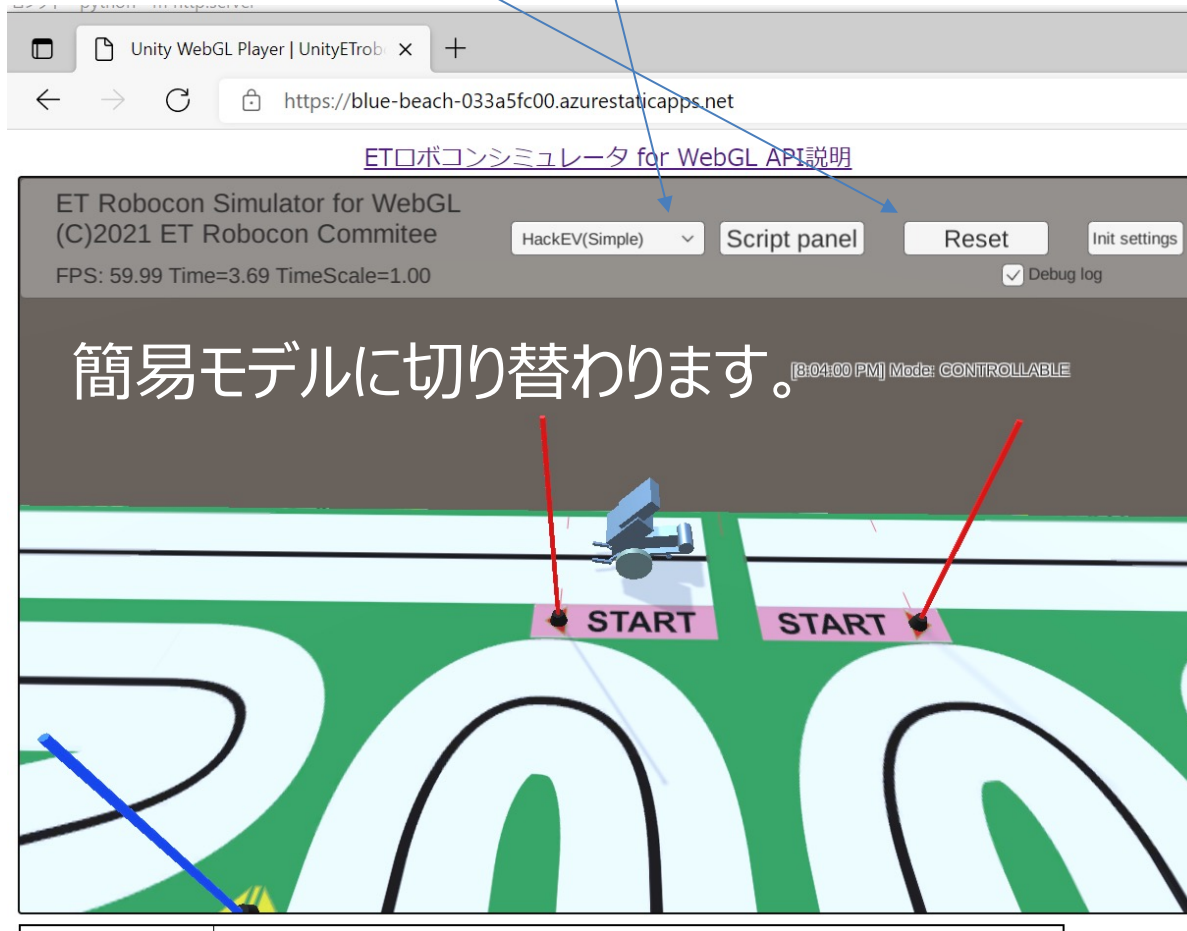
```
ev3_motor_set_power_left 50  
ev3_motor_set_power_right 50  
wait 10
```

- ③ シミュレータ部分をマウスでクリックし、Vキーを押す。
- ④ Run Scriptボタンを押す。



ヒント1 : FPSが20以下の場合(遅いPCの場合)

- ① HackEV(Simple)を選択する。
- ② Resetを押す。



実習1: HackEV(ロボット) を自由に動かそう

- ① シミュレータ上のResetを押す。
- ② エディタ上でev3_motor_set_power_left、ev3_motor_set_power_right、waitを使って、自由にプログラムを書く。for 文を使って動きを繰り返してみるもよし。

```
for i in range(0,10)
    ev3_motor_set_power_left 50
    ev3_motor_set_power_right 50
    wait 2
    ev3_motor_set_power_left -50
    ev3_motor_set_power_right 50
    wait 0.3
end for
```

- ③ シミュレータ部分をマウスでクリックし、Vキーを押す。
- ④ Run Scriptボタンを押す。

この方法でゲート1と2を通過し、ゴールまで走らせてみよう。

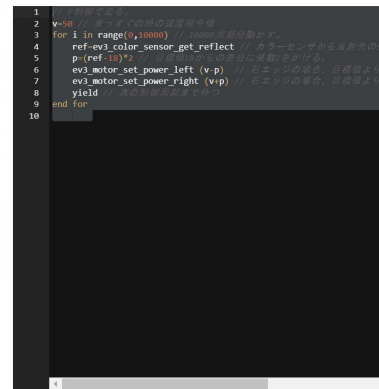
練習2: HackEV(ロボット) でライトレース

- ① WebGL版ETロボコンシミュレータをブラウザで起動する。
- ② エディタに下記プログラムをコピペする。

```
// P制御で走る。  
v=50 // まっすぐの時の速度指令値  
for i in range(0,10000) // 10000周期分動かす。  
    ref=ev3_color_sensor_get_reflect // カラーセンサから反射光の強さを取得する。  
    p=(ref-18)*2 // 目標値18からの差分に係数2をかける。  
    ev3_motor_set_power_left (v-p) // 右エッジの場合、目標値より暗い場合は右に曲がるように制御する。  
    ev3_motor_set_power_right (v+p) // 右エッジの場合、目標値より暗い場合は右に曲がるように制御する。  
    yield // 次の制御周期(5ms)まで待つ  
end for
```

- ③ シミュレータ部分をマウスでクリックし、Vキーを押す。
- ④ Run Scriptボタンを押す。

黒いラインの右側に沿って
走っている。



ヒント2： センサ値の確認方法

- ① ブラウザでF12キーを押す
- ② アプリケーション> ストレージ> セッションストレージの中のURLをクリックする。
- ③ スクリプト実行中は、MiniScriptのAPIによって取得可能な値を観測できる。

The screenshot displays the ET Robocon Simulator for WebGL API running in a browser. The browser's developer tools are open, showing the 'Applications' tab with 'Session Storage' expanded. A blue arrow points from the third step of the hint to the 'Session Storage' section. The 'MiniScript Editor' is also visible, showing a script for a robot competition simulation.

キー	値
ev3_motor_get_counts_arm	-46
ev3_motor_get_counts_left	1008
ev3_motor_stop_left	0
ev3_gyro_sensor_get_rate	2
ev3_color_sensor_get_r	51
ev3_color_sensor_get_reflect	18
ev3_motor_stop_tail	1
ev3_led_set_color	2
Time	3.315063
ev3_color_sensor_get_b	81
ev3_motor_set_power_tail	0
ev3_gyro_sensor_get_angle	0
ev3_motor_get_counts_right	950
ev3_color_sensor_get_ambient	0
ev3_motor_get_counts_tail	4
ev3_color_sensor_get_color	6
ev3_color_sensor_get_g	54
ev3_motor_set_power_right	50
ev3_motor_set_power_arm	0
ev3_motor_stop_right	0
ev3_ultrasonic_sensor_listen	0
ev3_motor_stop_arm	1
ev3_motor_set_power_left	50
ev3_ultrasonic_sensor_get_distance	2550

実習2: HackEV(ロボット) で左エッジライントレース

- ① WebGL版ETロボコンシミュレータをブラウザで起動する。
- ② 練習2のプログラムを改造して、左エッジをライントレースするように変更する。
- ③ シミュレータ部分をマウスでクリックし、Vキーを押す。
- ④ Run Scriptボタンを押す。



コース形状が複雑なので、途中でライントレースに失敗するかも。

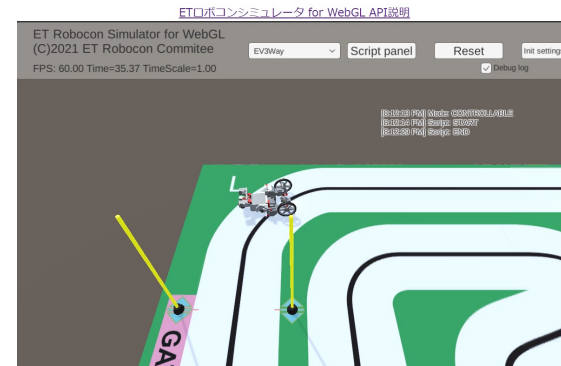
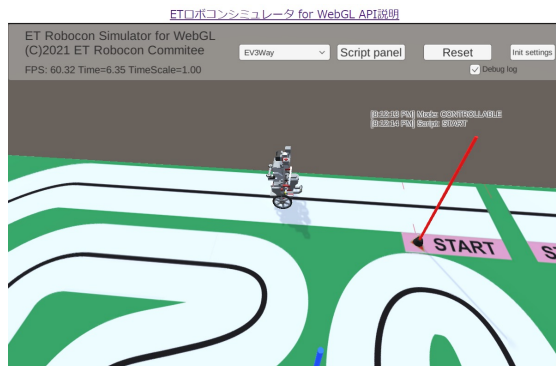
ゴールまで完走するにはどうすればよいか？

練習3: EV3way(倒立振子ロボット) を動かす

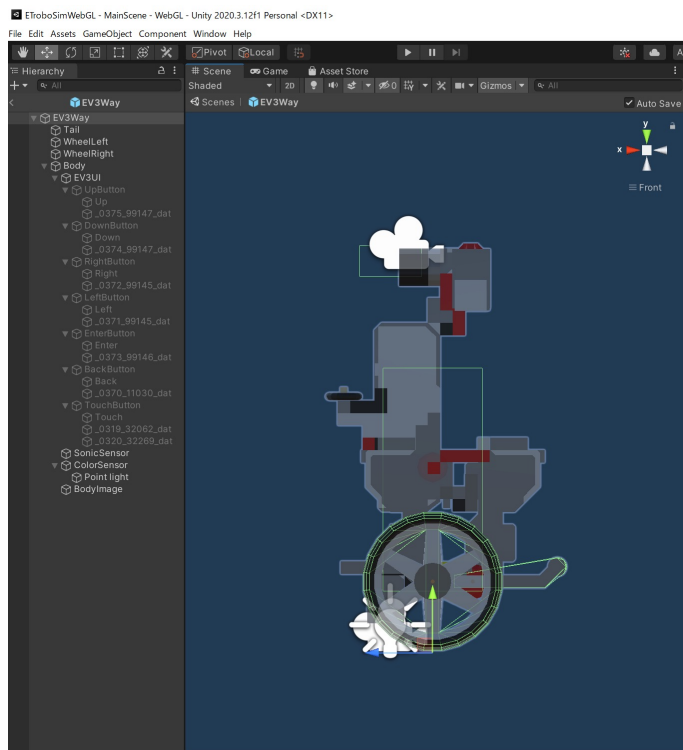
- ① 上のメニューでEV3Wayを選択し、リセットする。
- ② エディタに下記プログラムをコピーする。

```
speed=40 //前進速度
str=0 //旋回速度
balance_init // 倒立振子初期化
for i in range(0,3000)
    // 倒立振子制御
    balance_control speed,str, ev3_gyro_sensor_get_rate,0,ev3_motor_get_counts_left,ev3_motor_get_counts_right
    ev3_motor_set_power_left balance_pwm_l
    ev3_motor_set_power_right balance_pwm_r
    yield
end for
```

- ③ シミュレータ部分をマウスでクリックし、Vキーを押す。
- ④ Run Scriptボタンを押す。



ヒント 3 : 倒立振り子とは



ETロボコンで長く愛されてきた倒立振り子。
タイヤの回転数と回転速度、車体の角度と角速度
を入力として、倒立を維持するようにモータの制御値を
計算する。

今回は2017年のEV3Wayのモデル(塚本さん作)を
利用し、下記APIを提供している。頑張れば
MiniScriptを使って自分で実装することもできる。

- balance_init
- balance_control
- balance_pwm_l
- balance_pwm_r

大本のロジック : NXTway-GS(2輪型倒立振り子ロボット) C API

http://lejos-osek.sourceforge.net/jp/nxtway_gs.htm

参考にしたパラメータ : EV3way ETロボコンで使用している倒立振り子APIパラメーター
の再計算手順

<https://qiita.com/pulmaster2/items/08edce1ab097539b2deb>

実習3: EV3way(倒立振子ロボット) でライトレース

- ① 上のメニューでEV3Wayを選択し、リセットする。
- ② 練習3と、練習2or実習2のプログラム参考にして右or左エッジをライトレースするように変更する。
- ③ シミュレータ部分をマウスでクリックし、Vキーを押す。
- ④ Run Scriptボタンを押す。

ライトレースだけで、右エッジを30秒台、左エッジを40秒台でゴールまで走ればまあまあ速い。

4. さいごに

- 多くのETロボコン実行委員およびTOPPERS/箱庭WGの努力によって、2020年のリモートCS大会は無事終了した。リアル大会さながらの興奮を味わうことができた、評判は上々であった。
- YoutubeのETロボコンチャンネルにも色々と情報があるので、興味のある方はこちらも見ていただきたい。





終わり