

ロボットシミュレータ  
**Webots**を使って  
オンライン研修やってみた

LED-Camp実行委員会

祐源英俊

# 発表者自己紹介

祐源英俊（ゆうげんひでとし）

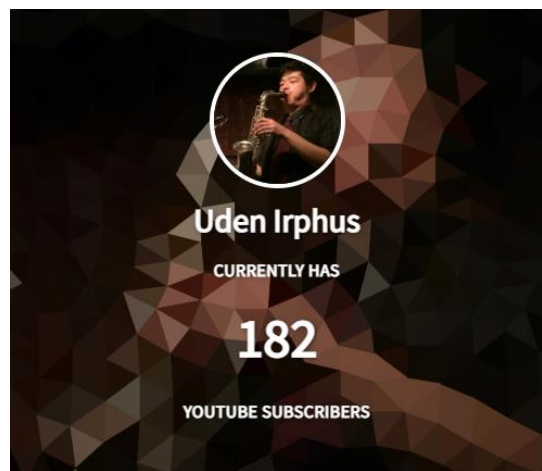
- 2018年 LED-Camp6参加
- 2018年より実行委員（3年目）
- 教材開発グループに所属
- オムロン株式会社 勤務(社会人1年目)
- ジャズの演奏が趣味



@Ke\_N\_551



@ken551



Uptown Jazz Underground



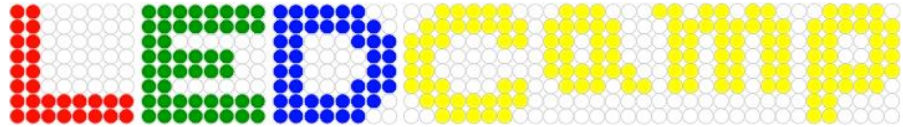
# 発表目次

1. はじめに
2. LED-Campってどんなイベント？
3. ロボットシミュレータwebotsについて
4. LED-Camp 9 教材開発裏話
5. Webotsプチハンズオン
6. Camp9開催報告・振り返り

# 発表目次

## 1. はじめに

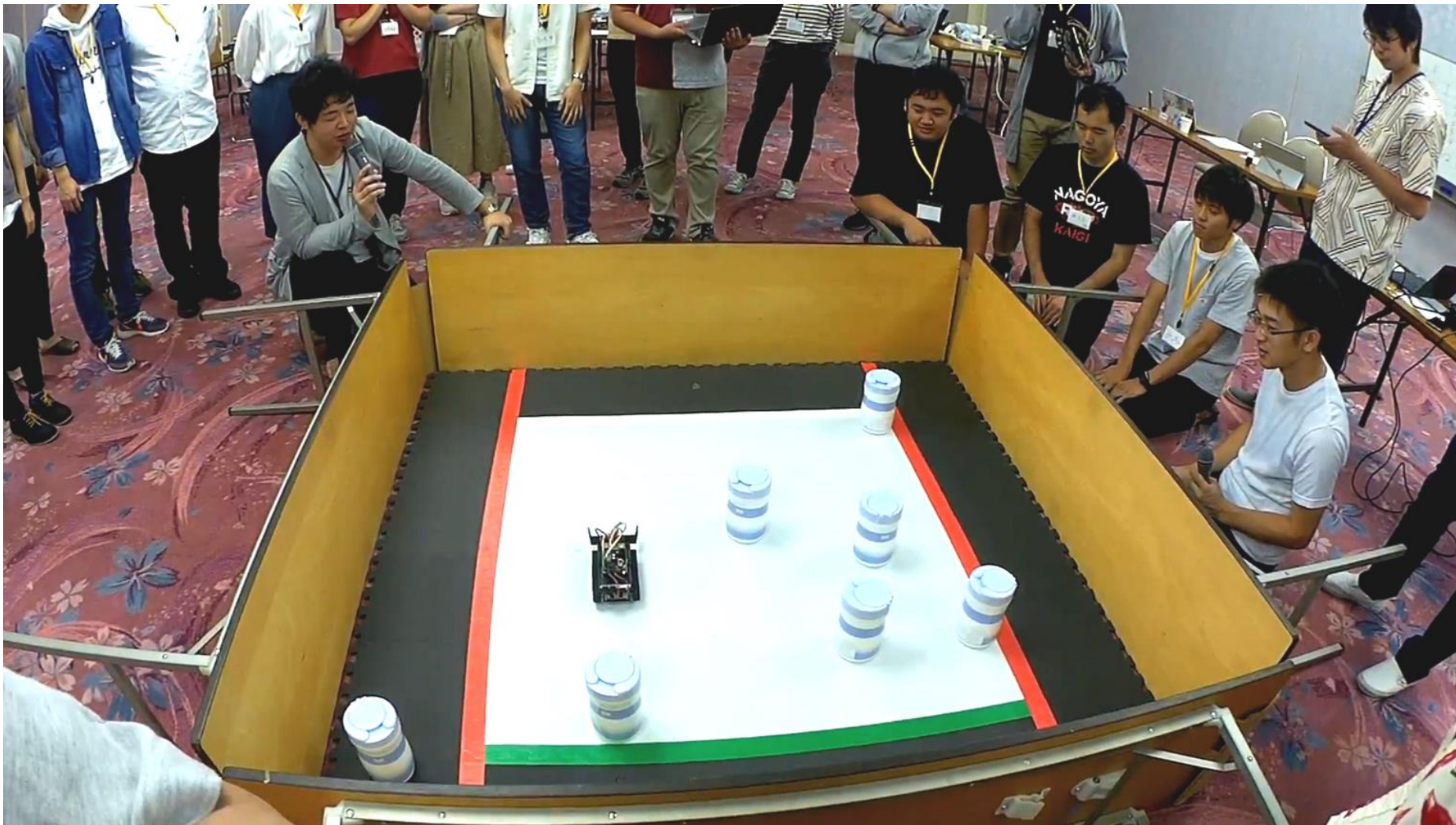
- 2. LED-Campってどんなイベント？
- 3. ロボットシミュレータwebotsについて
- 4. LED-Camp 9 教材開発裏話
- 5. Webotsプチハンズオン
- 6. Camp9開催報告・振り返り

そもそも、ってなんなん？

- **組込み初学者**（学生，社会人1,2年目など）  
を対象とする**合宿形式**の研修
- **アジャイル，スクラム，MDD**など開発手法の講義・実践
- **競技会**に向けたソフトウェアを**チーム**で開発する実習  
→**チーム開発のコツ・難しさなどを経験的に学べる！**



競技会ではこんなことをします

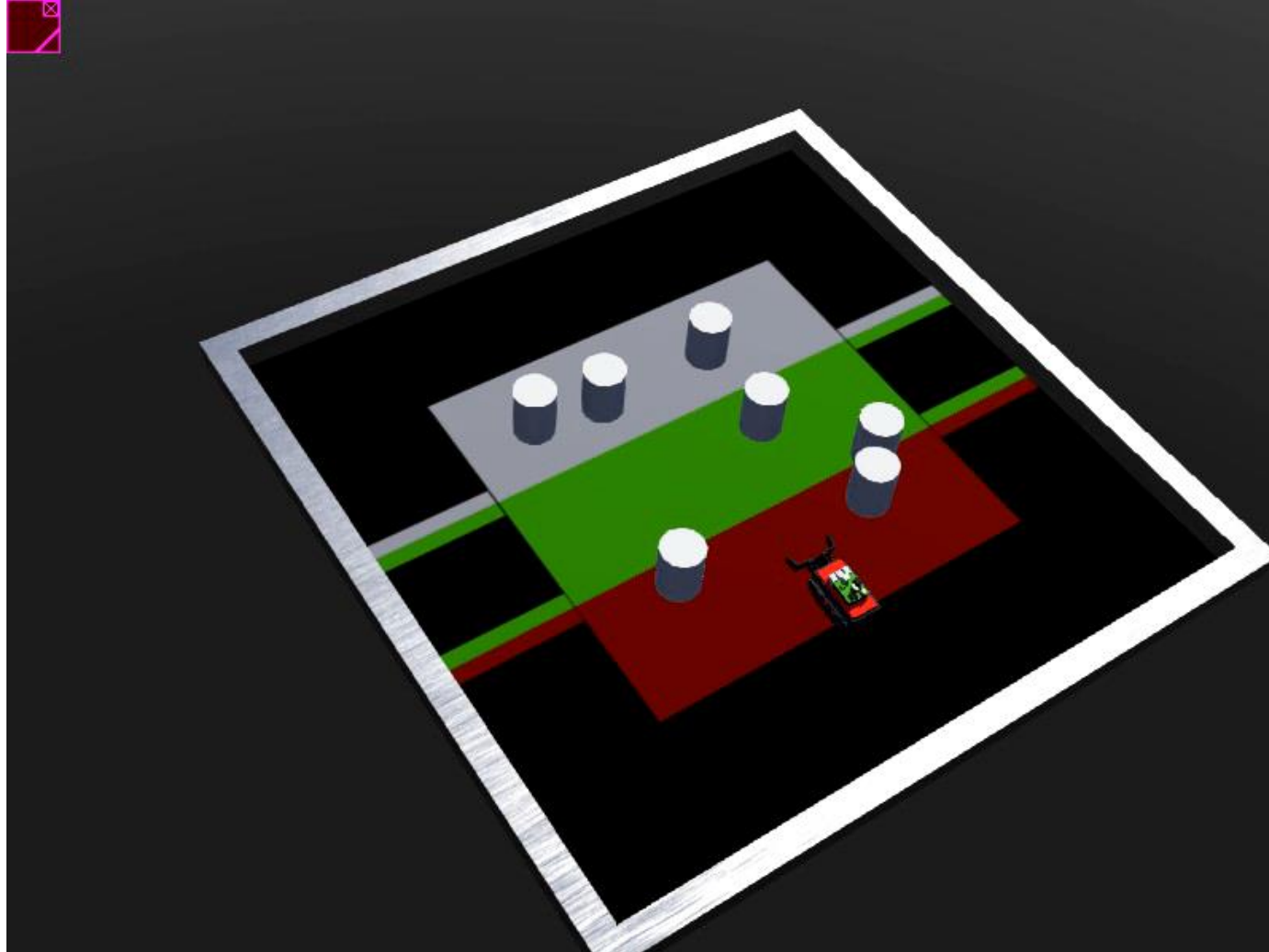




競技会ではこんなことをします



今年はこんなことをしました





# 発表目次

1. はじめに

**2. LED-Campってどんなイベント？**

3. ロボットシミュレータwebotsについて

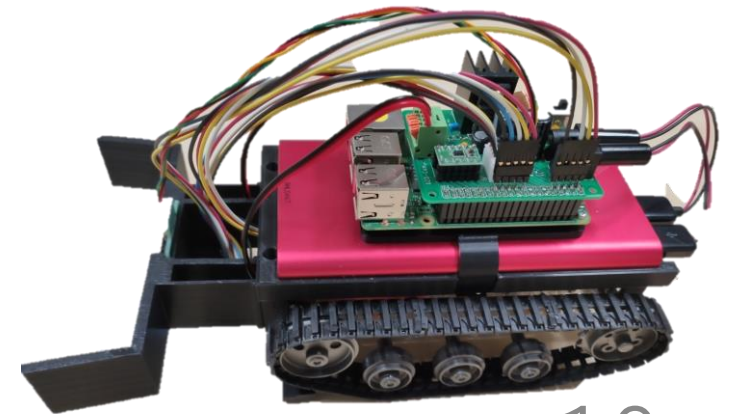
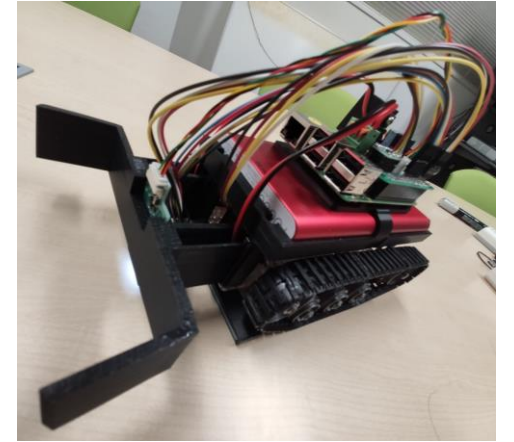
4. LED-Camp 9 教材開発裏話

5. Webotsプチハンズオン

6. Camp9開催報告・振り返り

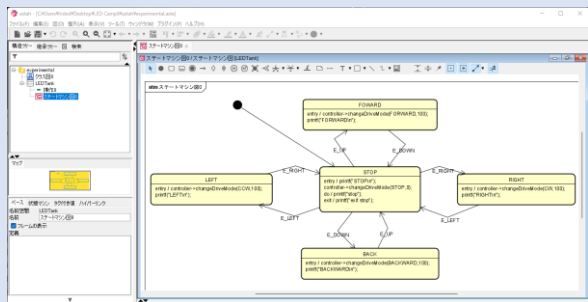
# オンサイト形式のLED-Camp

- 下呂温泉（山形屋）での**三泊四日の合宿**
- 講義，開発実習，発表会など密なスケジュール
- ナイトセッションを始めとする交流の機会も  
（LT大会，グループワークなど）
- Camp7（2019年）では  
**LED-Tank改**（右写真）を教材に使用



# Camp7での環境（既存資産）

参加者PC



UML図（astah\*上）

コード生成

C++

Tankコントローラ

既存の資産：

- コントローラ用テンプレート
- RasPi向けTankドライバ
- Astah\*に関する知識
- Tankの筐体
- 競技会のルール

Tank上RasPi



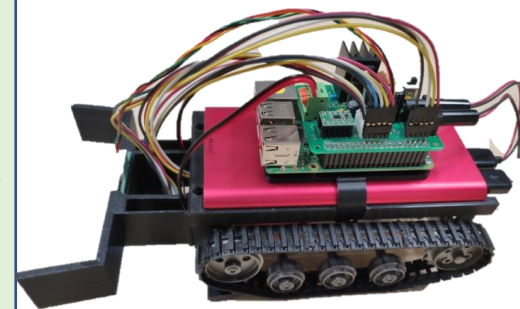
C++

Tankドライバ

コンパイル・  
リンク

.a

実行ファイル



実行！

# 今年のLED-Camp (Camp9) は . . . ?

- LED-Camp史上初の**完全オンライン開催**
- これまでの内容に加え, **オンラインコミュニケーション**および**オンラインチーム開発**の経験を目的に
- ロボットシミュレータWebotsを用いたバーチャル競技会を実施
- 今年度の開催報告はのちほど！





# Camp9の教育目標

- **組込みシステムの開発技術を習得する**
  - UMLを用いたモデル駆動開発
  - アジャイル開発手法（スクラムフレームワーク）
- **チーム開発を円滑にするコミュニケーション技術を習得する**
  - 目標に向けた合意形成
  - メンバーの考えを引き出す話し方・聴き方
  - チーム状況を適切に把握するフォローアップ

# Camp9の教育目標

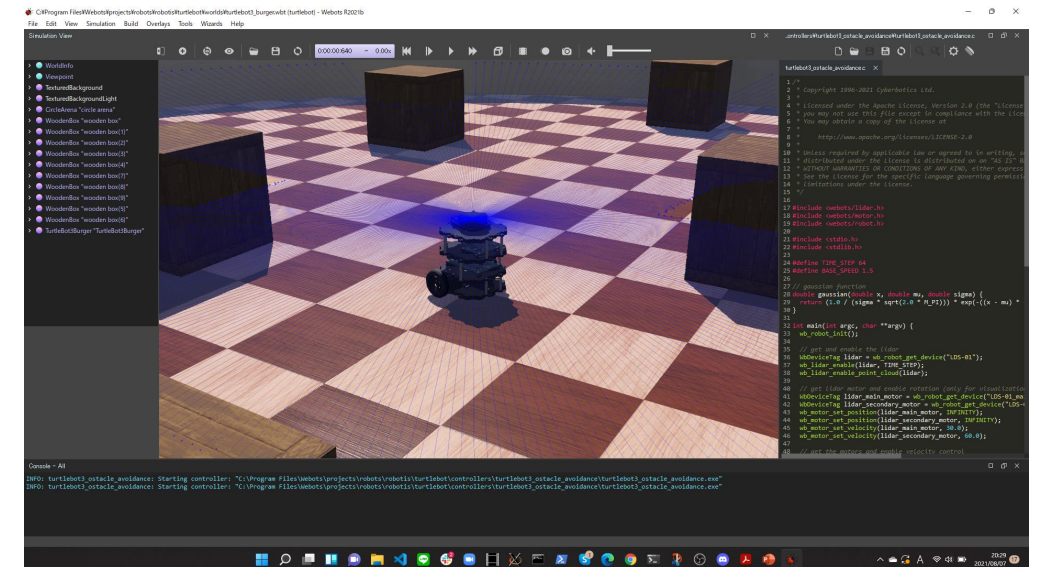
- **組込みシステムの開発技術を習得する**
  - UMLを用いたモデル駆動開発
  - アジャイル開発手法（スクラムフレームワーク）
- チーム開発を円滑にするコミュニケーション技術を習得する
  - 目標に向けた合意形成
  - メンバーの考えを引き出す話し方・聴き方
  - チーム状況を適切に把握するフォローアップ

# 発表目次

1. はじめに
2. LED-Campってどんなイベント？
- 3. ロボットシミュレータwebotsについて**
4. LED-Camp 9 教材開発裏話
5. Webotsプチハンズオン
6. Camp9開催報告・振り返り

# Webots

- Cyberbotics社（スイス）が開発するロボットシミュレータ
- バージョン2019aより**完全オープンソース**に※1（Apache 2.0）
- 物理エンジンは  
**ODEの改良版（Fork）**を使用
- サポート付きの有償プランあり
- 日本企業での使用実績もあり



※1:

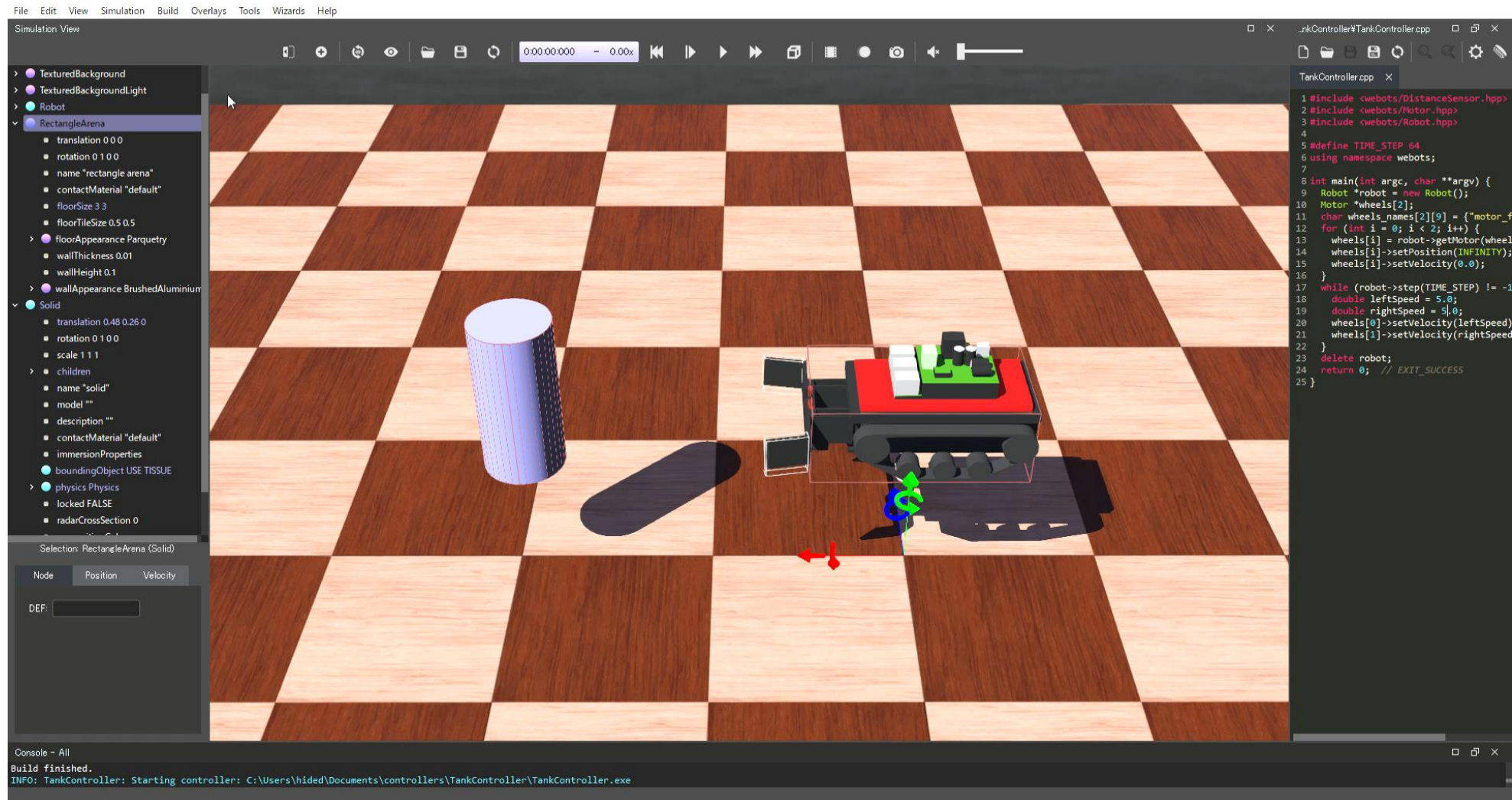
Version R2019a - Webots Goes Open Source – Webots Blog  
<https://www.cyberbotics.com/doc/blog/Webots-2019a-release>



# Webots概要



# Webotsによるプロトタイプ





# ちょっとしたデモ

お手元で体験していただけます

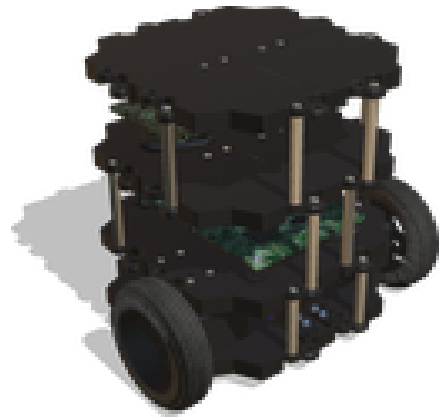
- 以下のリポジトリからclone（あるいはダウンロード）  
[https://github.com/LED-Camp/LED-Camp9\\_SWEST23\\_demo](https://github.com/LED-Camp/LED-Camp9_SWEST23_demo)
- リポジトリ内に含まれるWorldを開き，ビルド＆実行
- WASDでTankを操作できます（E押下で止まります）
- コード上部“USE\_〇〇\_SENSOR”のdefineで  
センサ使用/不使用を切り替え

# 特徴 1 : アセットのロボットが豊富

UserGuideによると車型・人型・動物型など**62種類**



**Jetbot**  
(NVIDIA)



**Turtlebot 3  
Burger**  
(Robotis)



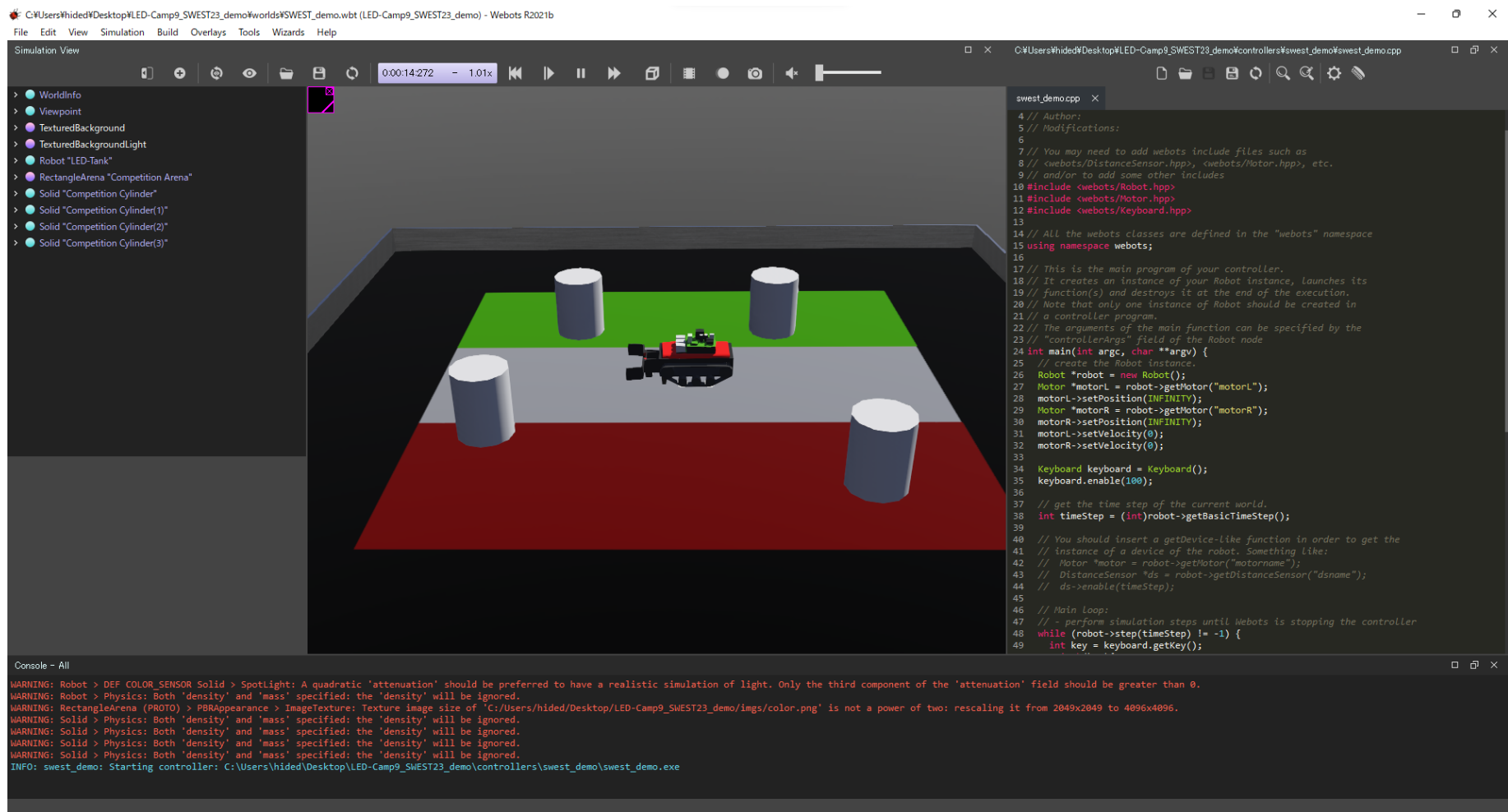
**Create**  
(iRobot)



**Spot**  
(Boston  
Dynamics)



# ちょっとしたデモ

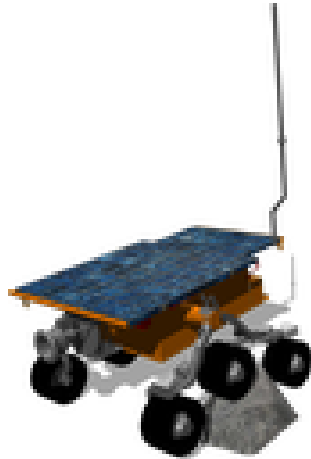


# 特徴 1 : アセットのロボットが豊富

こんなもので・・・



**Nao**  
(Softbank  
Robotics)



**Sojourner**  
(NASA)



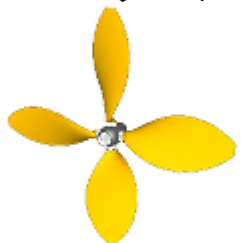
**AIBO**  
(Sony)



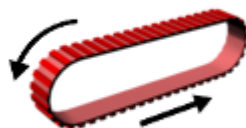
**BB-8**  
(Sphero)

## 特徴2：アセットのデバイスが豊富

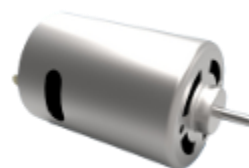
- アクチュエータ



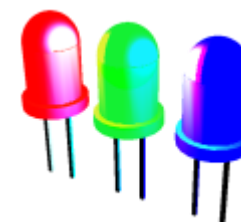
プロペラ



履帯



モータ

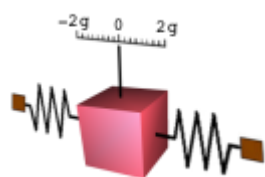


LED

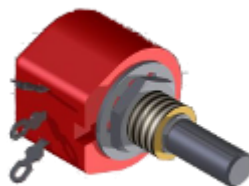


ペン

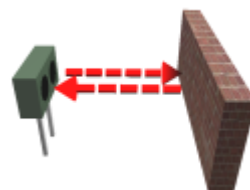
- センサ



加速度センサ



回転角センサ



距離センサ



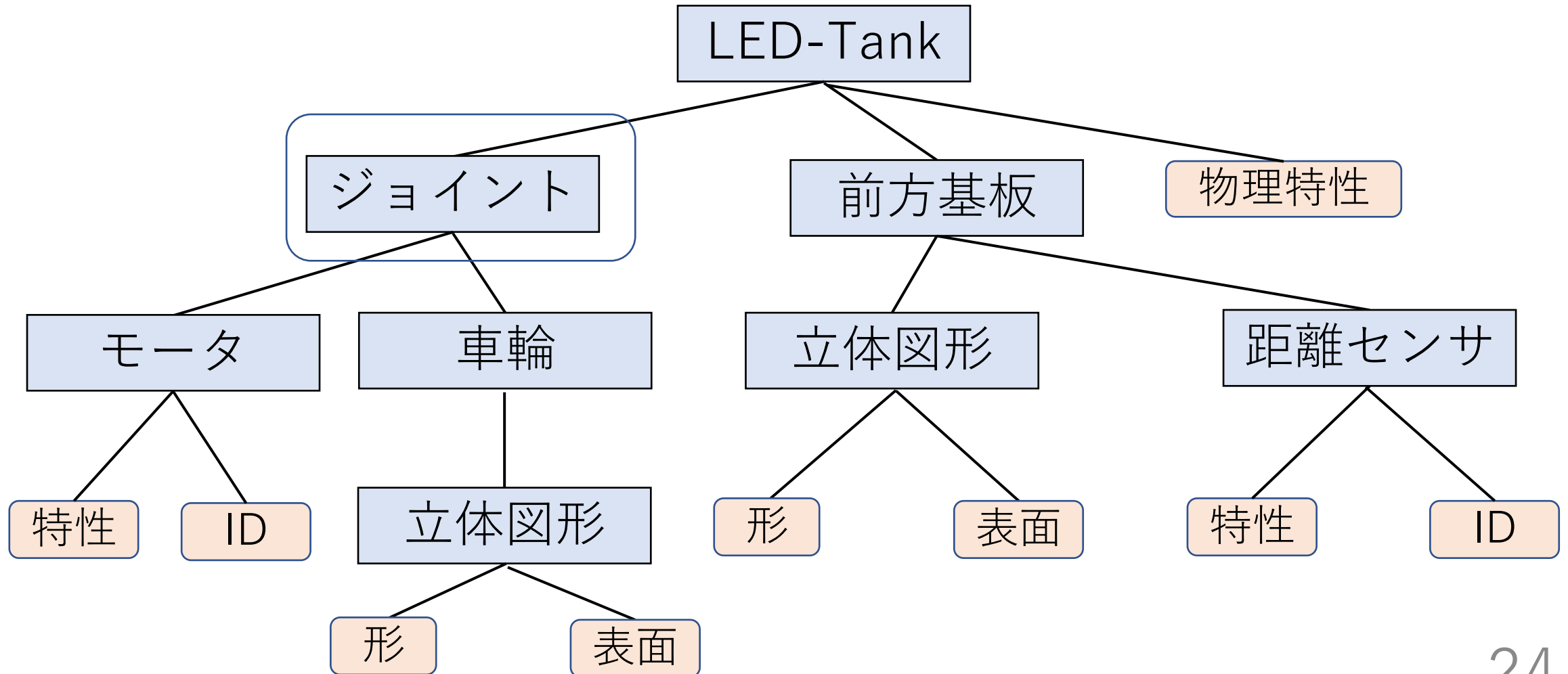
カメラ



LiDAR

※有償プランのみ使えるデバイスあり（例：HOKUYO製LiDAR, Kinectなど）

# 特徴3：ロボットは木構造で管理





## 特徴 4 : ロボットはControllerから制御

- ロボットに搭載されるデバイスをControllerから利用
- ControllerはC, C++, Java, Python, MATLABがサポート
- 1ロボットには1コントローラが紐づけ可能

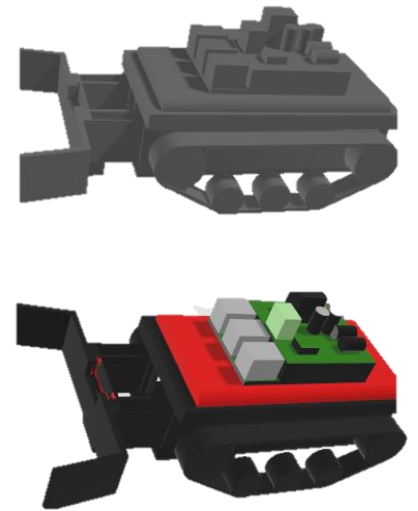
```
31 int main(int argc, char **argv) {  
32     // create the Robot instance.  
33     Robot *robot = new Robot();  
34     Motor *motorL = robot->getMotor("motorL");  
35     motorL->setPosition(INFINITY);  
36     Motor *motorR = robot->getMotor("motorR");  
37     motorR->setPosition(INFINITY);  
38     motorL->setVelocity(0);  
39     motorR->setVelocity(0);  
40     // 5.5.5. USE CAMERA SENSOR
```

デバイス指定・初期化

値（回転速度）セット

# 良かった・助かった点

- Windowsの場合**ビルドツールが付属する**  
(ビルド済みのMinGWコンパイラ)
- 比較的低スペックPCでもそこそこ動作する
- Fusion360から出力する色付き 3 Dモデル (.fbx) がそのまま読み込み**ルドツールが付属**  
(コンパイル済み・使える)
- 必要なセンサ類が揃っていてControllerから直感的に叩くことができる



# 困った点・躓いた点

- 英語の公式ドキュメント（と一部Youtube動画）が参考情報
- 光の反射は計算されない  
（＝**反射型カラーセンサは使えない**）

開発は比較的スムーズに進んだ印象

# 発表目次

1. はじめに
2. LED-Campってどんなイベント？
3. ロボットシミュレータwebotsについて
- 4. LED-Camp 9 教材開発裏話**
5. Webotsプチハンズオン
6. Camp9開催報告・振り返り

# Camp9開催の背景

- 感染症対策により**合宿形式のオフライン開催が不可能に**
- Camp1-7はすべてオンサイト開催  
→**既存のカリキュラム・プログラム資産では開催不可能に**



- **Camp8（2020年）は中止し，準備期間に**
  - 開催目的・要件・ニーズを整理
- 開催したい，けど集まりはできない・・・  
→**オンライン形式での開催を検討**



# 教材の要件

- そもそも動作がもっさりだとモチベーションが低下するのでは？  
→動作の軽さ
- 環境構築が複雑だと心折れそう & サポート大変  
→環境構築（の容易さ）
- 見た目がカッコいいとモチベーション上がりそう  
→見た目（の良さ）
- なるべく現実に近い方が面白いのでは？  
→シミュレーション性（Sim性）
- 教材開発が大変だと実行委員が大変なことになる  
→開発コスト



# 実機配布 or シミュレータ？

	実機配布	シミュレータ
利点	<ul style="list-style-type: none"><li>• 手元で動く楽しさ</li><li>• より「組込み」らしい</li></ul>	<ul style="list-style-type: none"><li>• 参加者全員に均一の競技環境</li><li>• コロナ禍もあり，ナウい</li><li>• これまでの知見を適用しやすい</li></ul>
欠点	<ul style="list-style-type: none"><li>• 故障時のフォローが難しい</li><li>• 配送など作業の手間がかかる</li><li>• 動かす場所を確保する必要</li><li>• これまでの知見が使いにくい</li></ul>	<ul style="list-style-type: none"><li>• 高スペックのPC環境が必要</li><li>• 組込み感が薄い</li></ul>

実機の故障と配送の手間，開発コストを心配し  
シミュレータを採用することに

# 2D or 3D ?

	2D (robocodeが候補)	3D
利点	<ul style="list-style-type: none"><li>• 要求スペックを抑えられる</li><li>• 開発コストを抑えられる</li></ul>	<ul style="list-style-type: none"><li>• 何より見た目がカッコいい</li><li>• いわゆる「シミュレータ」</li></ul>
欠点	<ul style="list-style-type: none"><li>• 見た目・魅力がイマイチ</li><li>• 現実から遠すぎるのでは？</li></ul>	<ul style="list-style-type: none"><li>• 高スペックのPC環境が必要</li><li>• 開発コストの高さ</li></ul>

見た目と現実の再現度を重視し、3Dシミュレータを用いる方針に

# 実行環境は？

	仮想環境 (WSL,Docker)	クラウド (AWSなど)	直接構築
利点	<ul style="list-style-type: none"><li>• Windows上でLinux</li><li>• ナウい</li><li>• Dockerならコンテナ配布できる</li></ul>	<ul style="list-style-type: none"><li>• 参加者環境に非依存</li><li>• 参加者環境に非影響</li></ul>	<ul style="list-style-type: none"><li>• 直接実行なので要求スペックは低い</li></ul>
欠点	<ul style="list-style-type: none"><li>• <b>使えるまでが複雑</b></li><li>• スペック要件が上がるのでは？</li></ul>	<ul style="list-style-type: none"><li>• 運営コストの高さ</li><li>• 予想以上にもっさり</li></ul>	<ul style="list-style-type: none"><li>• Windows向けはややこしい傾向</li></ul>

実行時の軽さを考え、**環境構築手順の制約を決めたうえで**  
直接構築してもらうことに

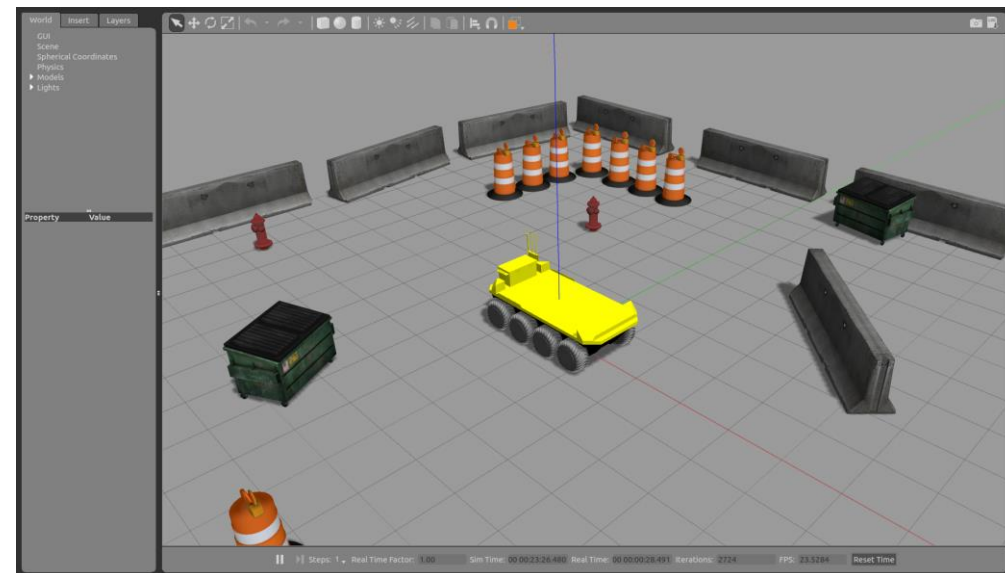
# 3Dシミュレータの選択

候補として，以下のものが挙げた

- Gazebo
- Unity
- 箱庭
- ODE（内製）
- Webots

# Gazebo

- Open Source Robotic Foundation (OSRF) が管理する 3Dロボットシミュレータ
- 物理エンジンにODEを使用
- ROSと連携して使われることが多い

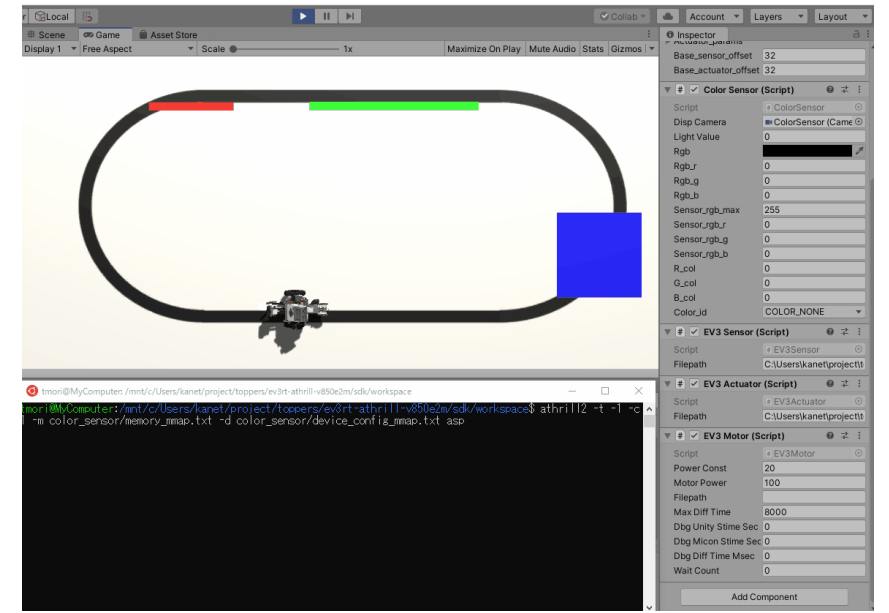


<http://www.clearpathrobotics.com/assets/guides/kinetic/moose/MooseSimulation.html>



# 箱庭

- 国産RTOSのTOPPERSプロジェクト内WG
- 仮想化技術を組み合わせたシミュレーションを実現
- 単体ロボット向けでは，ETロボコン向けシミュレータを実現，  
マイコンシミュレータとUnity 3D  
シミュレータの同期・連携



# Unity

- Unity Technologiesによるゲームエンジン・プラットフォーム
- IDE内蔵で，単体でC++・C#がビルド可能
- 有償・無償のアセットが豊富にあり，  
初心者でも手軽に本格的ゲームが作成可能



# ODE（内製）

- オープンソースの物理エンジン  
（BSD, LGPL）
- 無償利用可能な物理エンジンの中では軽量
- シミュレータ利用よりさらに軽量
- C++のサンプルもあり，手軽に(?) 3D描画・シミュレーション



# 他候補との比較

	参加者目線				実行委員目線
	動作の軽さ	環境構築	見た目	Sim性	開発コスト
Gazebo	×	×	○	○	△
Unity	△	○	○	?	×
箱庭	?	×	○	○	×
ODE(内製)	○	×	△	△	×
<b>Webots</b>	△	○	○	○	○

# 開催にあたっての心配事

- 教育目標は達成できるか？
- 「参加者目線」の4つの教材要件は満たせるか？  
動作の軽さ，環境構築，見た目，Sim性
- そもそも，参加・開発が楽しいか？

開催しての結果は次次章で発表



# 発表目次

1. はじめに
2. LED-Campってどんなイベント？
3. ロボットシミュレータwebotsについて
4. LED-Camp 9 教材開発裏話
- 5. Webotsプチハンズオン**
6. Camp9開催報告・振り返り

# プチハンズオン1:Controllerを作成しよう

- 目的：ロボットを制御するControllerの作り方を学ぶ
- ゴール：
  - Controllerの作成・追加ができる
  - デバイス類を制御する基本的なコードの書き方がわかる

# Controllerを作成しよう (1/2)

1. Wizards > New Robot Controller
2. “Language”を”C++”に, “IDE”を“Webots”に,  
“Controller Name”を“SWEST\_Handson”に指定
3. ”Open ‘SWEST\_Handson.cpp’ in Text Editor”に  
チェックを入れてFinish
4. 画面左のViewからRobot “LED-Tank” > controllerをクリック,  
”select”ボタンから”SWEST\_Handson”を選択

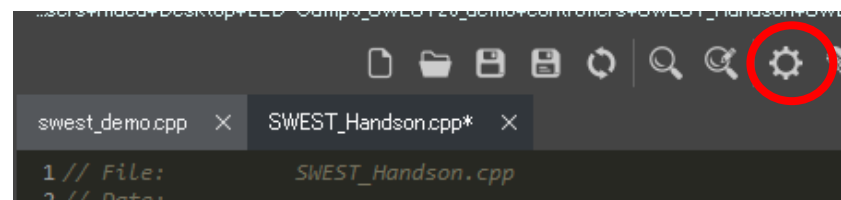
# Controllerを作成しよう(2/2)

5. 画面右側のエディタから, SWEST\_Handson.cppの一部に  
コードを追加  
(二か所↓→)

```
10 #include <webots/Robot.hpp>
11 #include <webots/Motor.hpp> // 追加
12
```

```
25 Robot *robot = new Robot();
26 // 以下を追加
27 Motor *motorL = robot->getMotor("motorL");
28 motorL->setPosition(INFINITY);
29 Motor *motorR = robot->getMotor("motorR");
30 motorR->setPosition(INFINITY);
31 motorL->setVelocity(10);
32 motorR->setVelocity(10);
33 // 以上を追加
```

6. エディタ上部の歯車ボタンを  
押下し, ビルド

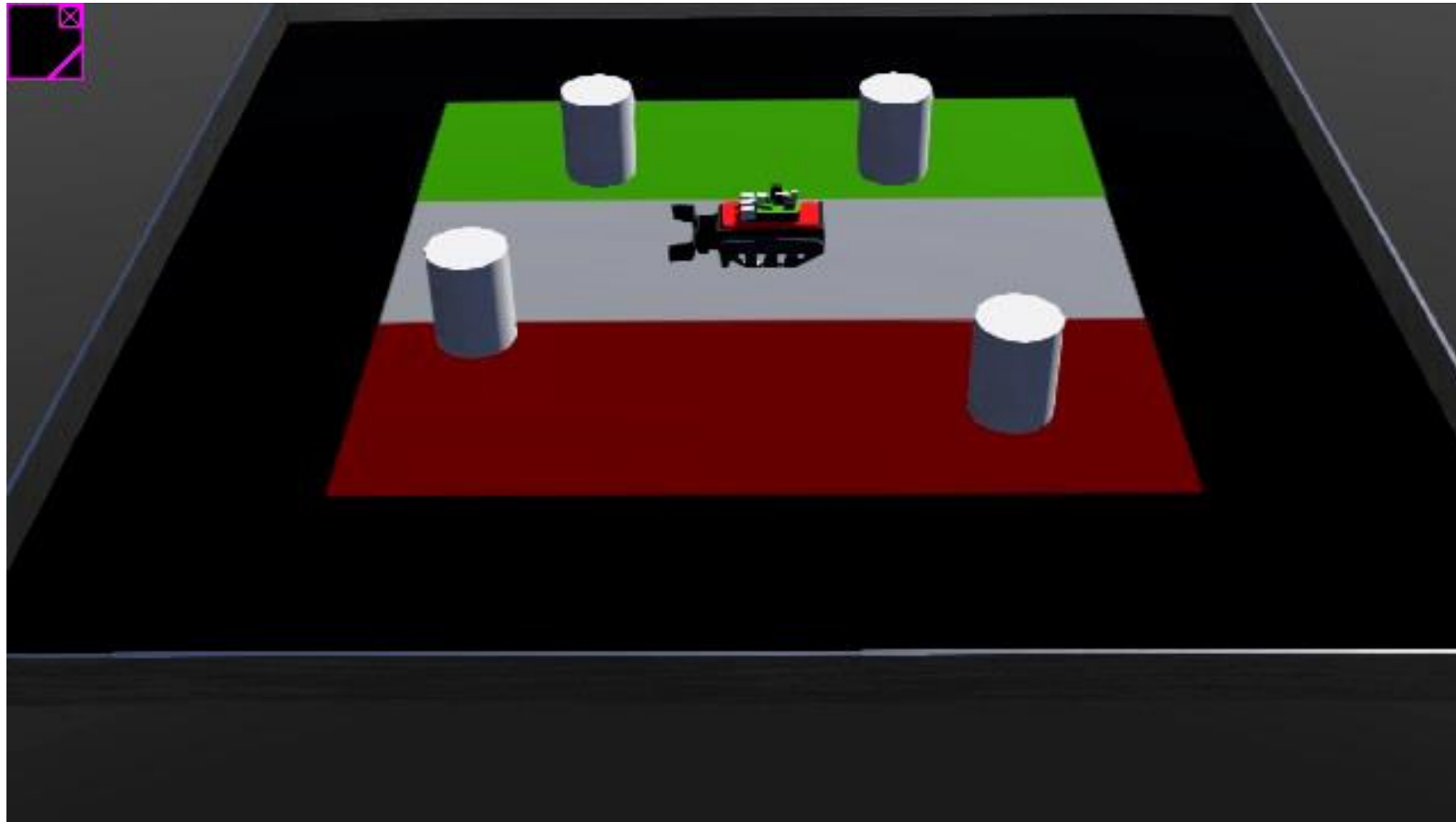


7. シミュレーション画面上部の  
再生ボタンを押し, シミュレーション実行



# プチハンズオン1:Controllerを作成しよう

- このようにLED-Tankが前進します



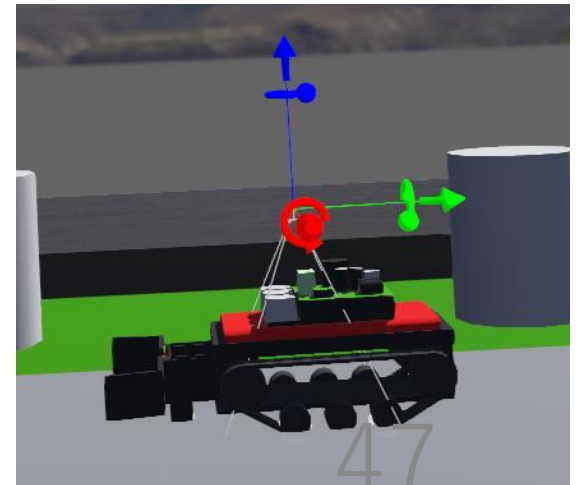


# プチハンズオン2:Cameraを取り付けよう

- 目的：ロボットにデバイスを取り付けて利用する方法を学ぶ
- ゴール：
  - ロボットにデバイスを追加することができる
  - Controllerから追加したデバイスにアクセスすることができる

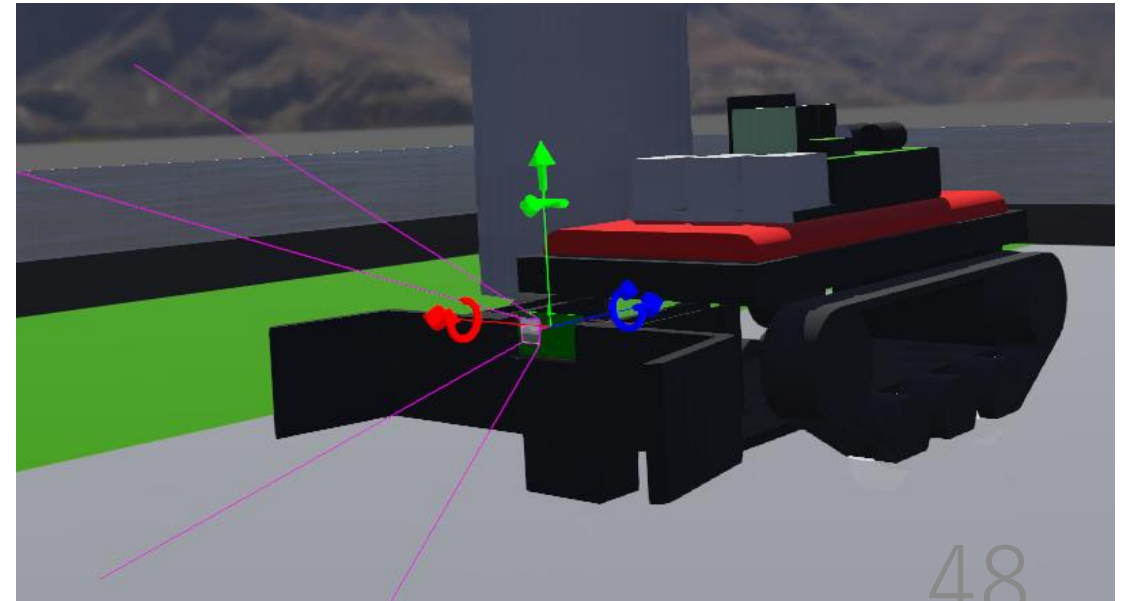
# Cameraを取り付けよう(1/3)

- Robot “LED-Tank”のchildrenを右クリックし, “Add New”
- Base Nodes > Camera を選択してadd
- Cameraのnameを”camera\_swtest”に変更
- View > Optional Rendering > Show Camera Frustumsからカメラの撮影範囲描画をOnに
- Camera選択→青矢印をドラッグして車体上部へ  
(カメラの撮影範囲描画が確認できます)



# Cameraを取り付けよう(2/3)

- 各色矢印をドラッグor座標直接入力でTank正面にCamera設置
  - Transition: 0, -0.13, 0.02
  - Rotation: 0, 0.707105, 0.707105, 3.14159
- 取得, 有効化のコードをControllerに追加  
(詳細は次ページ)



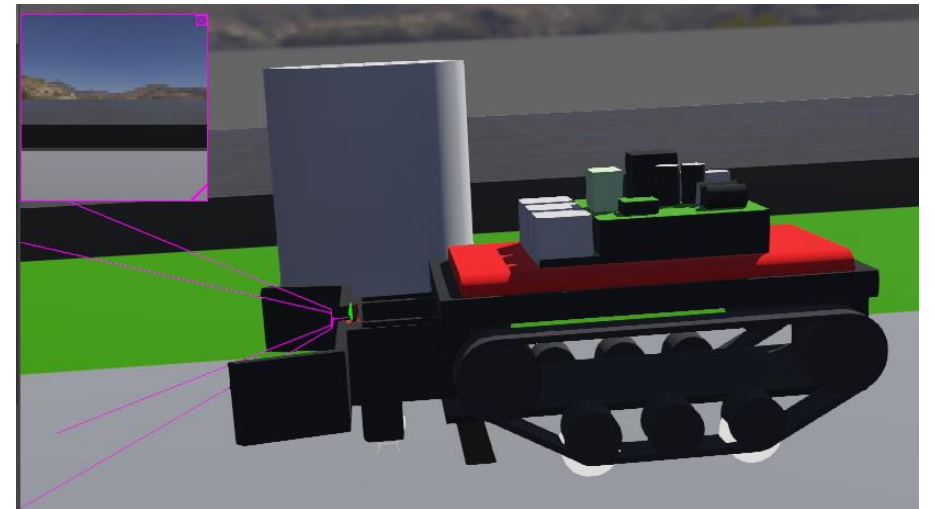
# Cameraを取り付けよう(3/3)

- 以下の画像に従ってコードを変更

```
10 #include <webots/Robot.hpp>
11 #include <webots/Motor.hpp>
12 #include <webots/Camera.hpp> //追加
13
```

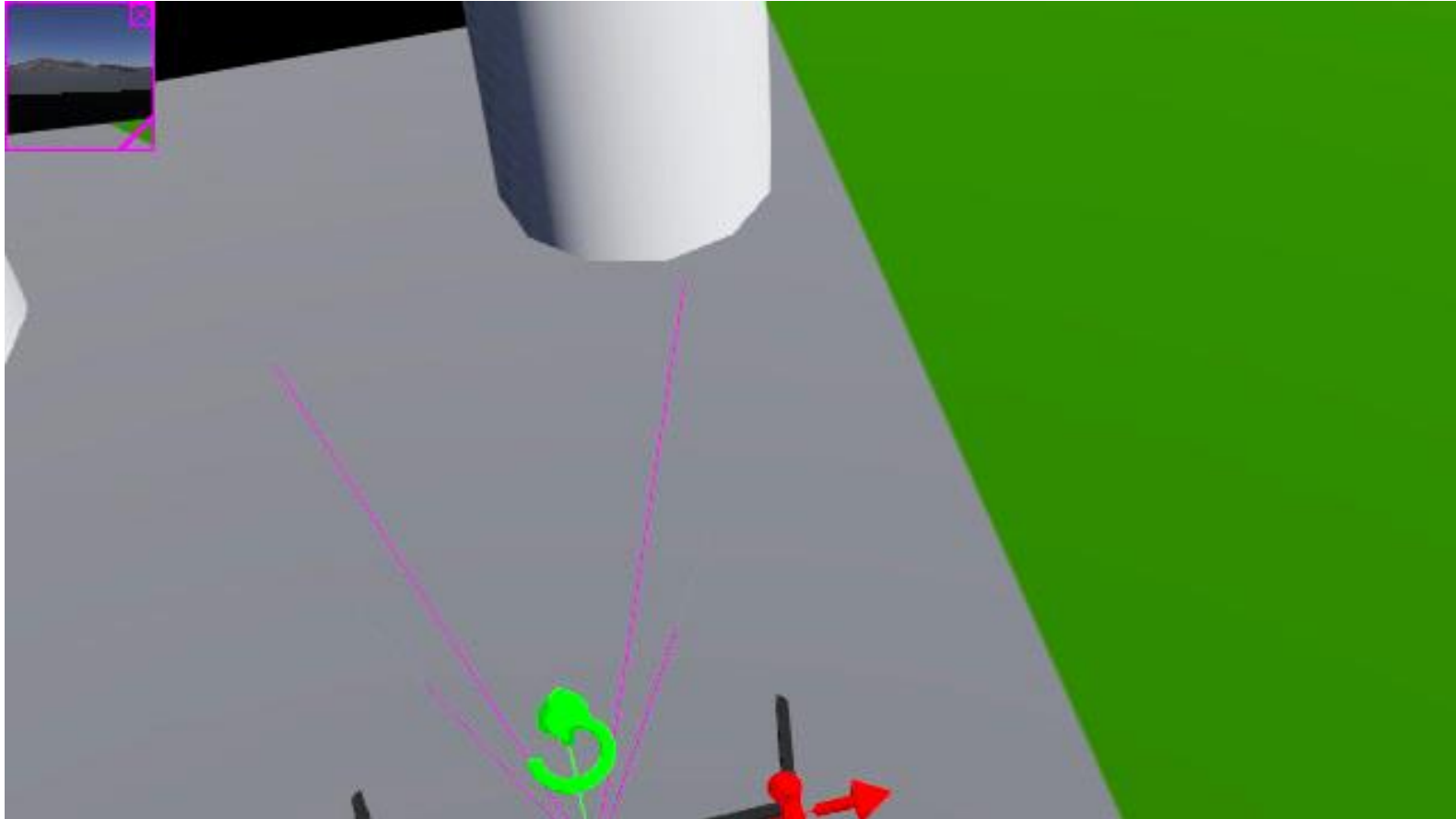
```
33 motorR->setVelocity(10);
34 // 以下を追加
35 Camera *cam = robot->getCamera("camera_swext");
36 cam->enable(20);
37 // 以上を追加
38 // set the time step of the current world
```

- 保存・ビルドして実行！



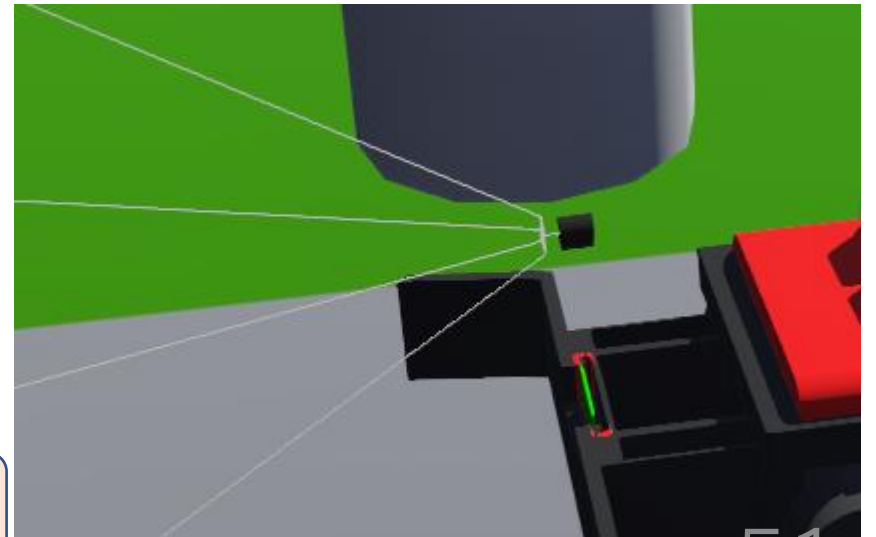
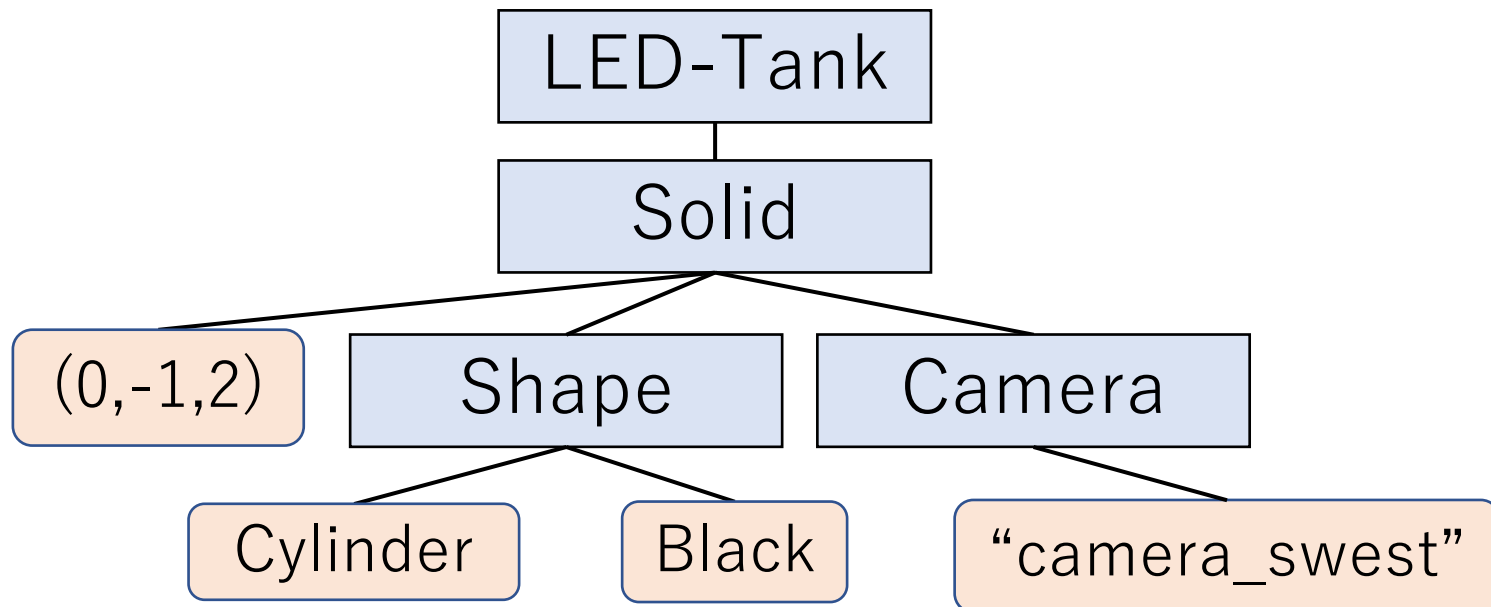
# プチハンズオン2:Cameraを取り付けよう

- このようにカメラ画像が表示されます



# Cameraを取り付けよう:補足

- Cameraなどデバイス自体は何も描画されません
- 図形をつけたい場合, 以下のようにSolidを作成し, Shapeと一緒にCameraをぶら下げます



# 発表目次

1. はじめに
2. LED-Campってどんなイベント？
3. ロボットシミュレータwebotsについて
4. LED-Camp 9 教材開発裏話
5. Webotsプチハンズオン
- 6. Camp9開催報告・振り返り**

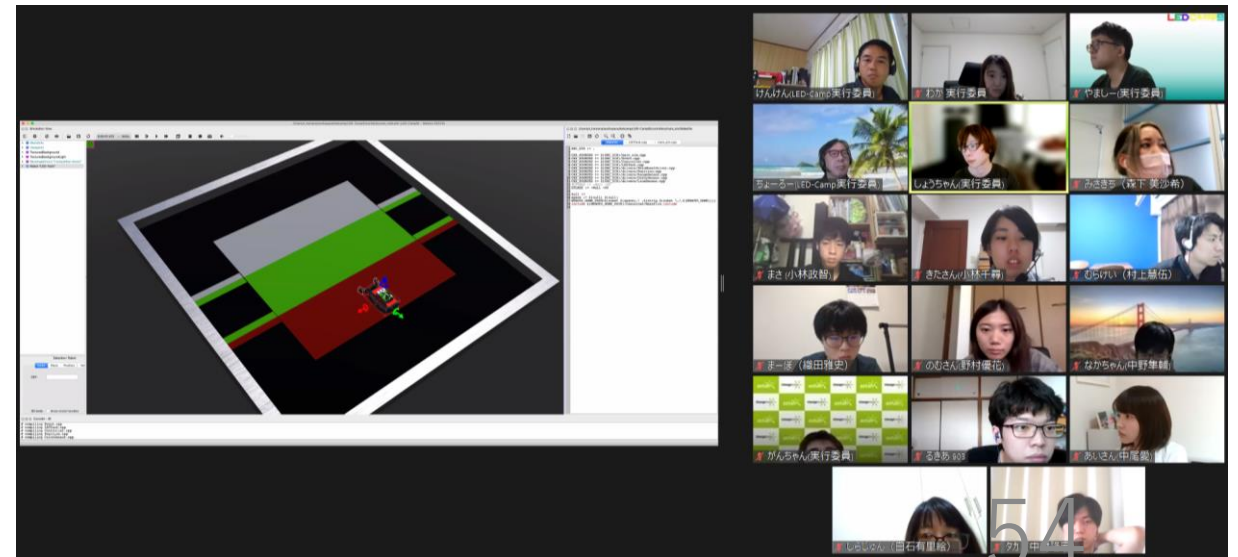
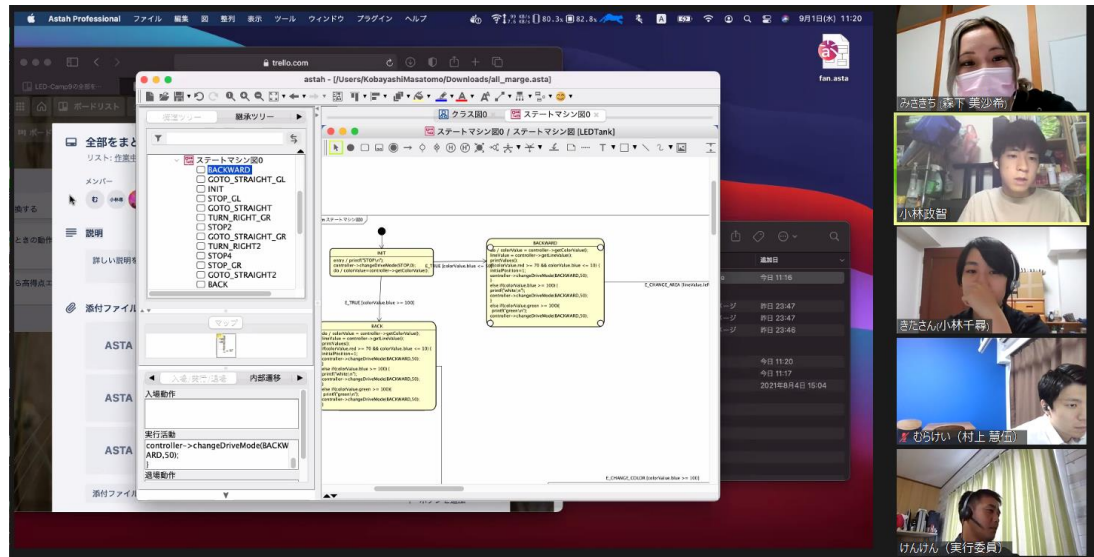
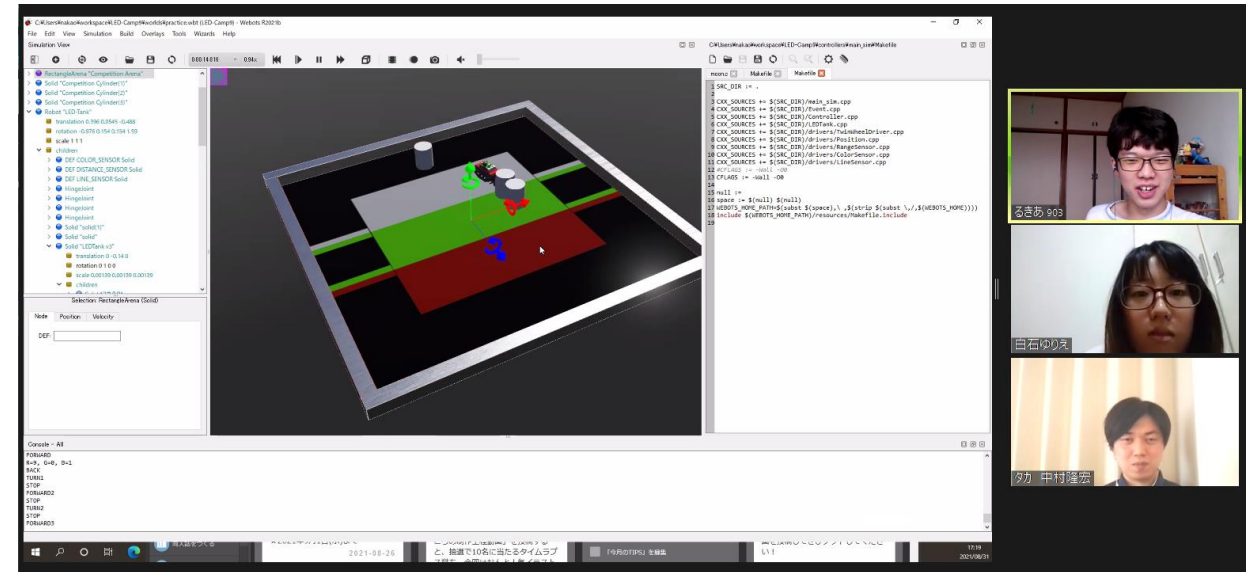
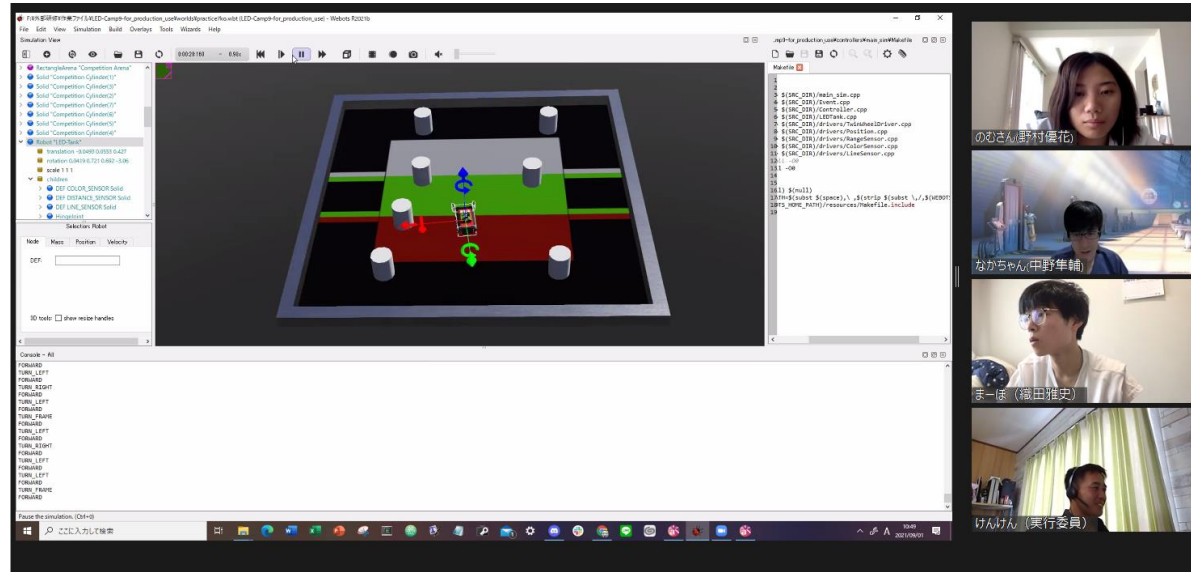


# LED-Camp9 開催実績

- 参加者**11名**（社会人5名，学生6名）
- 8/30(月) ~ 9/1(水)の三日間
  - 1日目：座学
  - 2日目：**開発**
  - 3日目：**開発・競技会**



# オンライン開発の様子



# 開催にあたっての心配事（再掲）

- 教育目標は達成できるか？
- 「参加者目線」の4つの教材要件は満たせるか？  
動作の軽さ，環境構築，見た目，Sim性
- そもそも，参加・開発が楽しいか？

# 教育目標は達成できたか？

- どのチームもアジャイル・スクラム・MDDを実践
- どのチームも競技会で動くモデル&高い得点
- オンラインでの開発・コミュニケーションについてのいい感想

ただし技術的に深掘するような質問・姿勢は少なめ？

# 要件が満たせたか？

- 動作の軽さ  
→「重くて動かねえ！」はなし， ある程度動いてた？
- 環境構築（の容易さ）  
→Webotsの環境構築で躓いた参加者は確認できず
- 見た目（の良さ）  
→動く様子を見て喜んでもらったようだった
- シミュレーション性（Sim性）  
→ time-baseのノイズ挿入で実行のたびに異なる結果

 これら要件の達成が円滑かつ意欲的な実習につながっていたのでは？

# 要件が満たせたか？

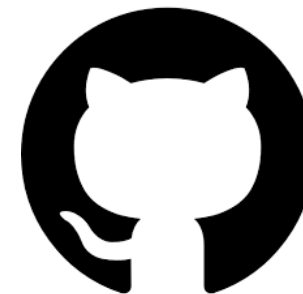
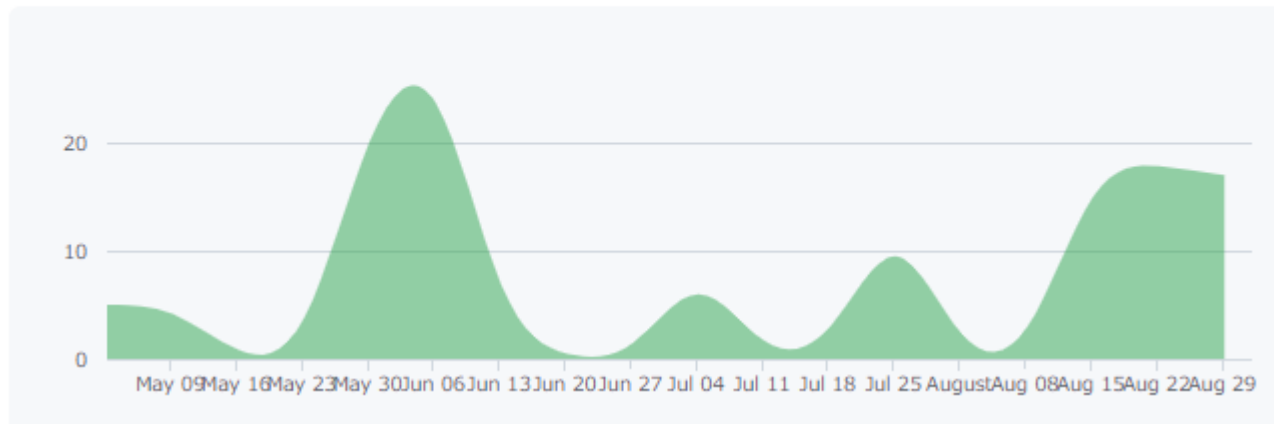
- 開発コスト

→比較的スムーズに開発できたという意味では○  
(5月中旬本格開発開始, 7月中旬には大まかに完成)

May 2, 2021 – Sep 1, 2021

Contributions: Commits ▼

Contributions to for\_production\_use, excluding merge commits and bot accounts



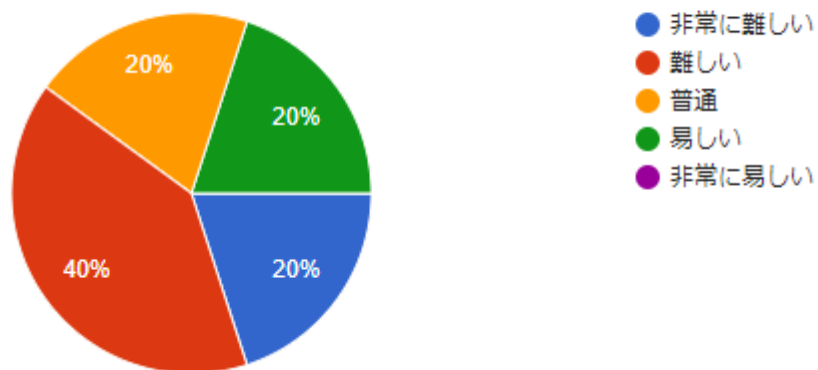
[LED-Camp/LED-Camp9](https://github.com/LED-Camp/LED-Camp9)  
(github.com)

# アンケート結果

## 2日目（回答10件）

「チーム開発実習」の開発教材 LED-Tankシミュレータ利用の難易度はいかがでしたか？

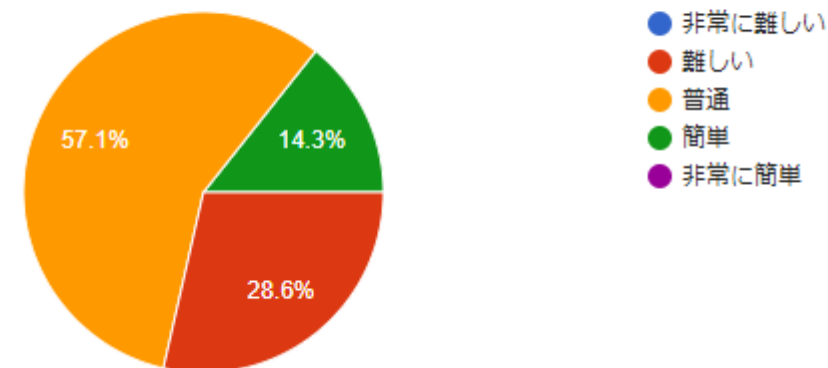
10 件の回答



## 3日目（回答7件）

「チーム開発実習」の開発教材 LED-Tankの難易度はいかがでしたか？

7 件の回答





# アンケート結果

チーム開発実習及び競技大会は楽しめましたか？

7件の回答



- 楽しめた
- どちらともいえない
- 楽しめなかった

# 参加者に聞いたシミュレータの長/短所

- 長所

- いろいろ試すのがやりやすかった
- 個体差がないことによるやさしさ・公平感  
（「なんかわからんけど動かん」がなかった）
- 「自分のコードが動いて楽しい」という感じ  
（見た目・Sim性が寄与？）

- 短所

- センサの位置等が確認しにくい（筐体が手に取れない）
- やっぱり実機に触ってみたかった

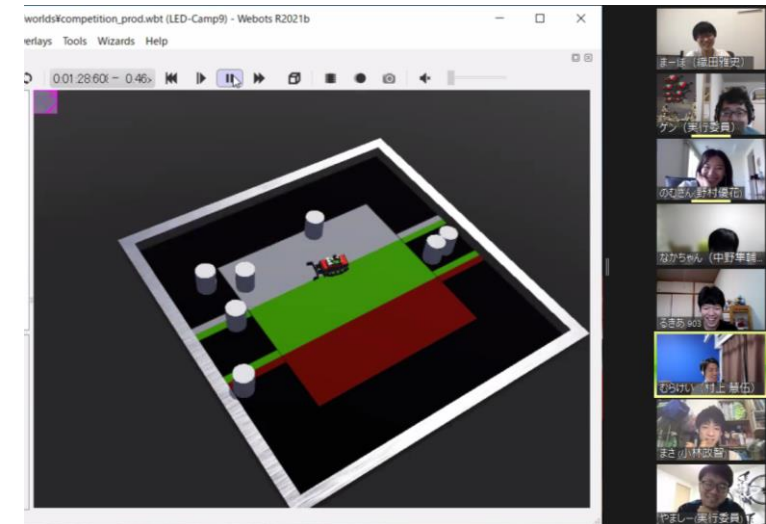
# オンライン研修についての気づき

- 参加者が時間外開発しがち  
（チーム会議で「昨日の夜試したんですけど…」がちらほら）
- 個人作業の様子が外から見えない（→フォローしにくい）
- ブレイクアウトルームの行き来が大変（→フォローしにくい）
- 他チームとの交流・情報交換が少ない  
（とはいえ、最終日の交流の時間は盛り上がっていた様子）

# 発表まとめ



- ロボットシミュレータWebotsを使用してオンライン研修してみた
- Webotsはオープンソースのロボットシミュレータ  
比較的軽量かつすぐに慣れるインタフェース
- オンラインでも参加者には楽しんでもらえる  
(ただし主催側のサポートは少ししんどい)
- 検討した4要件はかなり重要では？
- **実行委員大募集中！！！！**



ここからは・・・？

Webotsやオンライン研修について語りあいましょう！

- デモ・ハンズオンの延長戦
- LED-Campの定めた教材要件についてのご意見
- LED-Camp9開催してみてどうだったの？聞きたい！
- オンライン研修をやってみようと思うけど・・・
- オンライン研修やってみたらこうだった！

以下，補足資料

- 教材について定めた要件を達成できたか？
- 事後アンケートから，参加者はどう感じていたか？

# メモ

- 結構重たさに苦しんでる様子はなさそう
- 議論するときは、一つの画面を見ながらあーでもないこーでもないしている→課題はある程度難しい・考える余地がある方がいい？
- 個人作業の時は作業の内容が見えないことが多い→フォローがしにくい
- 実行してみてわかる不具合がやっぱり多い・・・
- フォローが結構大変（実行委員7,8人，参加者11人3チーム）
- やっぱりサビ残してたみたい
- 他チームの動きが見えないので良くも悪くも成果物にばらつきが
- どうしても個人開発したものを組み合わせる形になる



# 参加者からの声

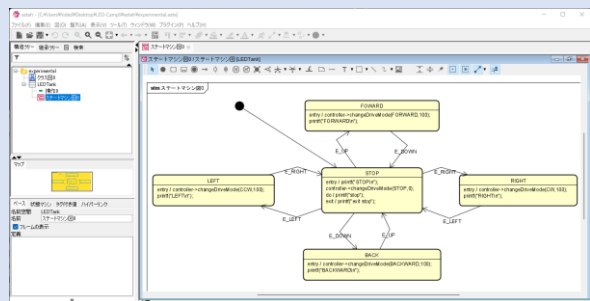
- センサの位置など筐体の特徴がわかりにくい  
→センサー位置は物理的に見えるようにした方がいい
- 実機の個体差がなかった, 「なんかわからんけど動かん」がなかった
- 見知らぬ人同士に意味を感じた
- シミュレータは重さを感じなかった
- 何回も試せてよかった
- とはいえ実機触って見たかった
- ステートマシン図から直接生成して動いたのが面白かった
- シミュレータが結構簡単に動いてよかった

# 参加者の想定を基にした教材の要件

- **教育目標を達成できる教材であること**
  - アジャイル，スクラム，**MDD**を実践できる
  - **オンラインでコミュニケーションをとりながらチーム開発できる**
- 組込み初学者の参加を想定としたものであること
  - **Windows環境でできること**
    - 開発環境やその構築が複雑でないこと
    - ある程度のランダム要素が含まれること
- ある程度快適に演習ができること
- 見た目がいいこと

# Camp7での環境（既存資産）

参加者PC



UML図（astah\*上）

コード生成

C++

Tankコントローラ

既存の資産：

- コントローラ用テンプレート
- RasPi向けTankドライバ
- Astah\*に関する知識
- Tankの筐体
- 競技会のルール

Tank上RasPi



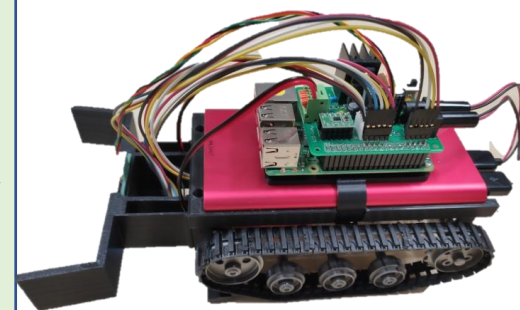
Tankドライバ

C++

コンパイル・  
リンク

.a

実行ファイル



実行！

# 既存資産を基にした教材の要件

- **astah\*のUML図から生成するコードが使えること**
  - 別プロセスで立ち上げて通信？
  - リンクさせて同一の実行ファイルに？
  - 動的リンクライブラリ（DLL）を利用？
- **Windows上に構築できるビルドツールを使用すること**
  - CYGWIN? MSYS? WSL?
- **Camp7の競技ルールを再現できること**

# 発表目次

1. はじめに
2. LED-Campってどんなイベント？
3. LED-Camp 9 の議論
4. LED-Camp 9 での教材開発議論
5. ロボットシミュレータwebotsについて
6. Camp9開催報告・振り返り