プログラミング概論

第10回　2024年11月27日
App Inventorによる
Androidアプリ開発の実践
(4) 物理シミュレーション1

今回の授業内容

- コンピュータシミュレーションとは
- コンピュータの画面上で「動き」を表現するには
- 等速直線運動
- 的当てゲームを作る

コンピュータシミュレーションとは

コンピュータシミュレーション

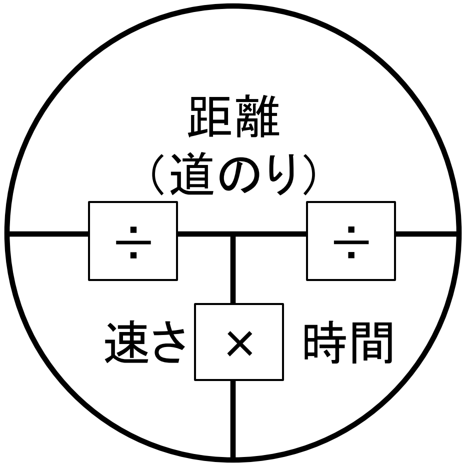
自然や社会に見られる現象をコンピュータ上で模擬的に再現すること　→　分析や予測，ゲームに有用

コンピュータ上で模擬するためには必要なこと
・現象の本質をとらえ，そのしくみを単純化
・ルールや数式により記述

第10回～第12回の授業では
物理現象のひとつである力学のうちの
「物体の運動」に焦点を当て，コンピュータ上で
模擬する方法の基礎を学ぶ

“動き”のシミュレーション

- 一回あたりの移動量＝速度
- 毎回速度×時間を位置に加えればよい

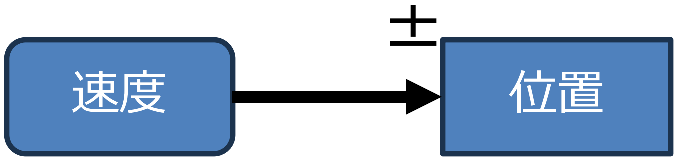


距離

（道のり）

÷ ÷

速さ × 時間



±

速度

位置

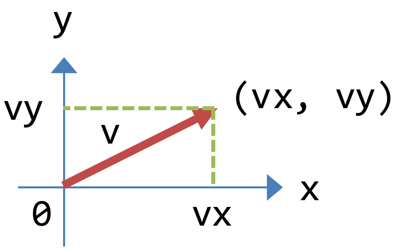
$$x' = x + v_x \times \Delta t$$

（元のxに速度x時間を足したものが次のx）

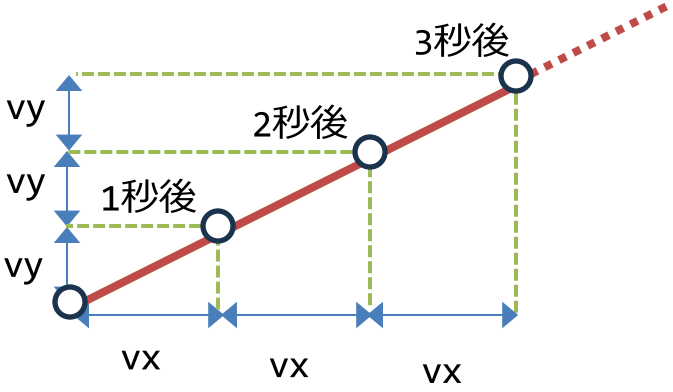
$$y' = y + v_y \times \Delta t$$

（元のyに速度x時間を足したものが次のy）

コンピュータの画面上で「動き」を表現するには


y
vy
v
(vx, vy)
0
vx
x

1秒間に動く量


3秒後
vy
2秒後
vy
1秒後
vy
vx
vx
vx

t秒後の位置は？

$$x_{(t)} = x_{(0)} + v_x \times t$$

$$y_{(t)} = y_{(0)} + v_y \times t$$

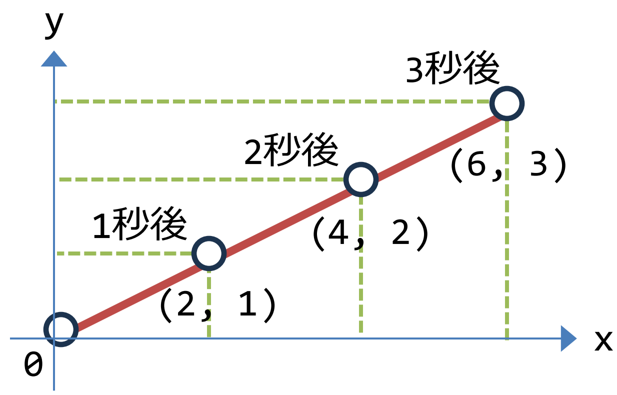
↓

速度が一定でないとダメ

↓

$$x_{(t+1)} = x_{(t)} + v_{x(t)}$$

$$y_{(t+1)} = y_{(t)} + v_{y(t)}$$


y
3秒後
2秒後
(6, 3)
1秒後
(4, 2)
(2, 1)
0
x

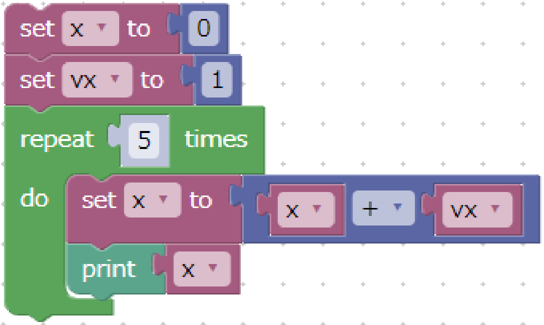
$$x_{(4)} = 0 + 2 \times 4 = 8$$

$$y_{(4)} = 0 + 1 \times 4 = 4$$

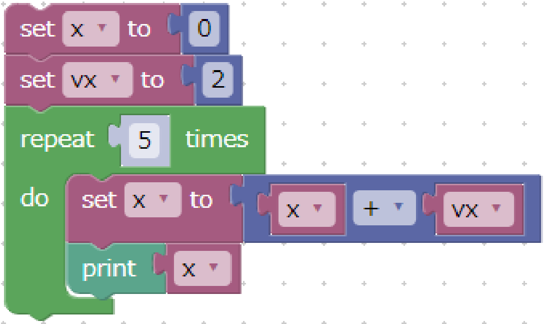
いまの時刻tから1秒後の位置を求める.
x(t)とy(t)はいまの位置,
vx(t)とvy(t)は1秒間に動く量を表している.

等速直線運動

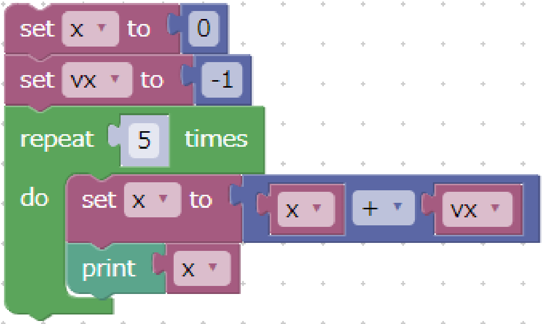
表示される値をそれぞれ考えてみよう



```
set x to 0
set vx to 1
repeat 5 times
do  set x to x + vx
    print x
```



```
set x to 0
set vx to 2
repeat 5 times
do  set x to x + vx
    print x
```



```
set x to 0
set vx to -1
repeat 5 times
do  set x to x + vx
    print x
```

, , , ,

, , , ,

, , , ,

vx（移動量）は繰り返しの間，常に一定

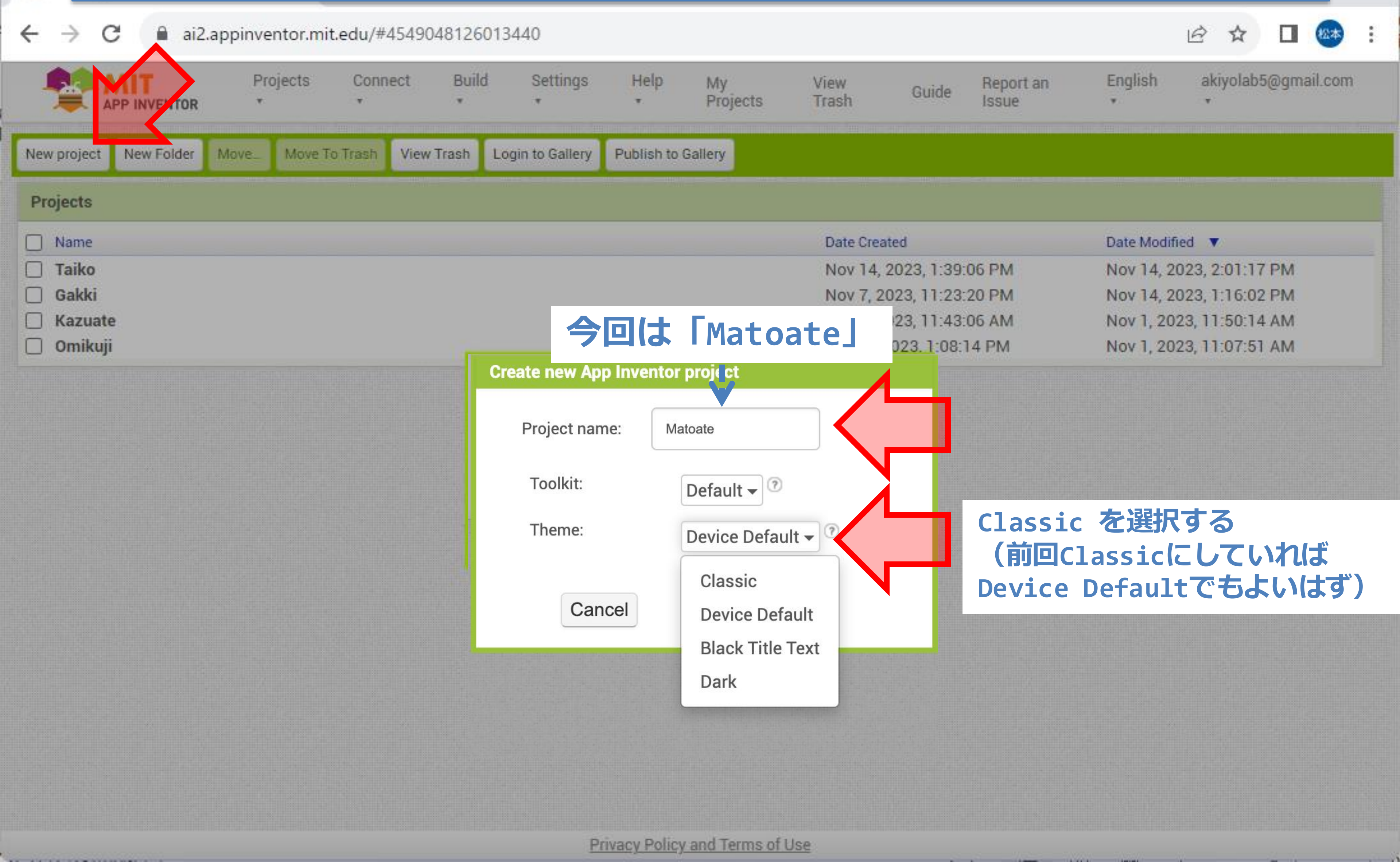
↓

等速直線運動

考えたら Blockly Code で確かめよう

的当てゲームを作る

新しいプロジェクトをつくる



ai2.appinventor.mit.edu/#4549048126013440
MIT APP INVENTOR
Projects
Connect
Build
Settings
Help
My Projects
View Trash
Guide
Report an Issue
English
akiyolab5@gmail.com
New project
New Folder
Move...
Move To Trash
View Trash
Login to Gallery
Publish to Gallery
Projects
Cancel
Privacy Policy and Terms of Use

☐ Name	Date Created	Date Modified ▼
☐ Taiko	Nov 14, 2023, 1:39:06 PM	Nov 14, 2023, 2:01:17 PM
☐ Gakki	Nov 7, 2023, 11:23:20 PM	Nov 14, 2023, 1:16:02 PM
☐ Kazuate	23, 11:43:06 AM	Nov 1, 2023, 11:50:14 AM
☐ Omikuji	023, 1:08:14 PM	Nov 1, 2023, 11:07:51 AM

Create new App Inventor project

Project name: Matoate

Toolkit: Default

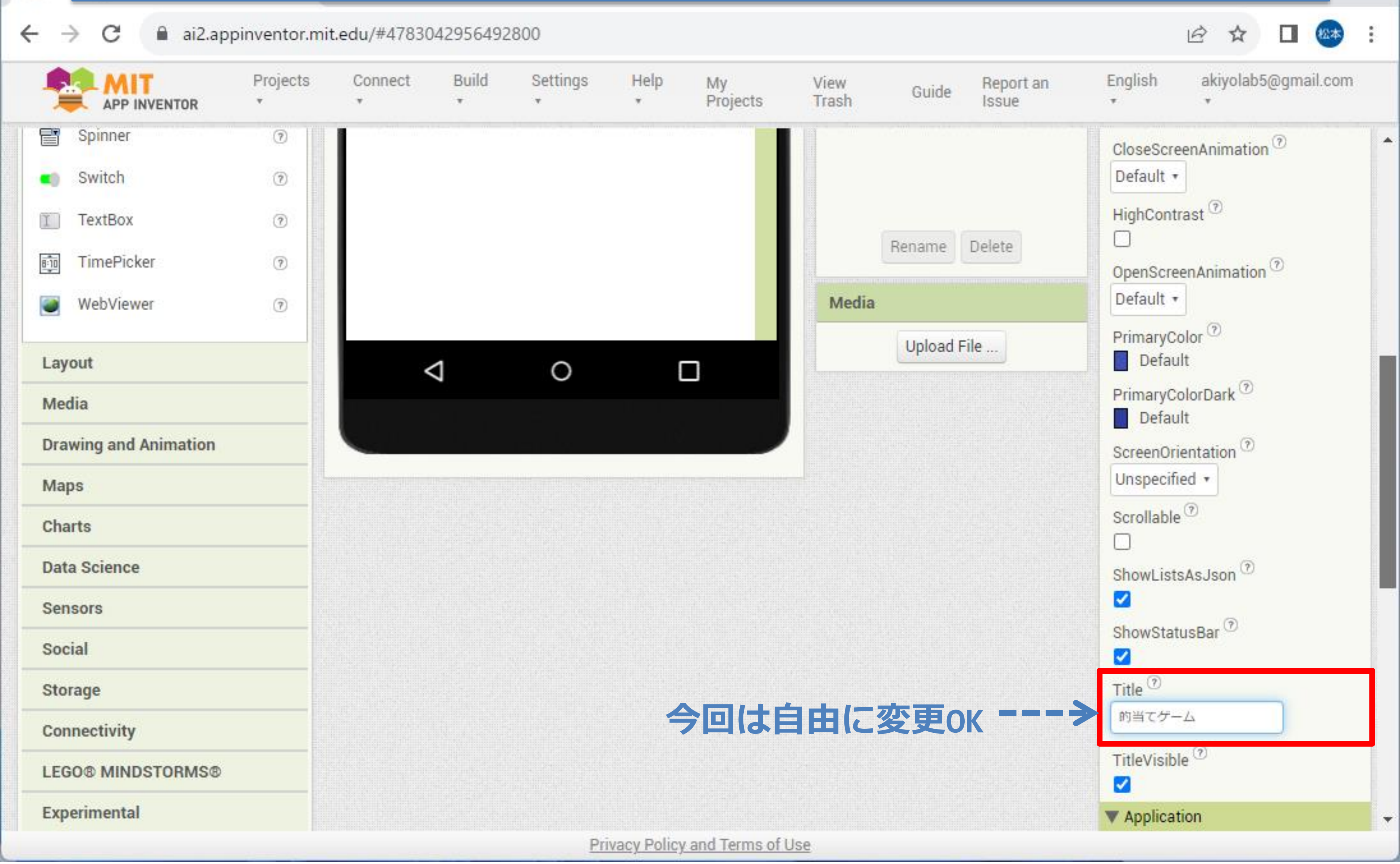
Theme: Device Default

- Classic
- Device Default
- Black Title Text
- Dark

今回は「Matoate」

Classic を選択する
（前回Classicにしていれば
Device Defaultでもよいはず）

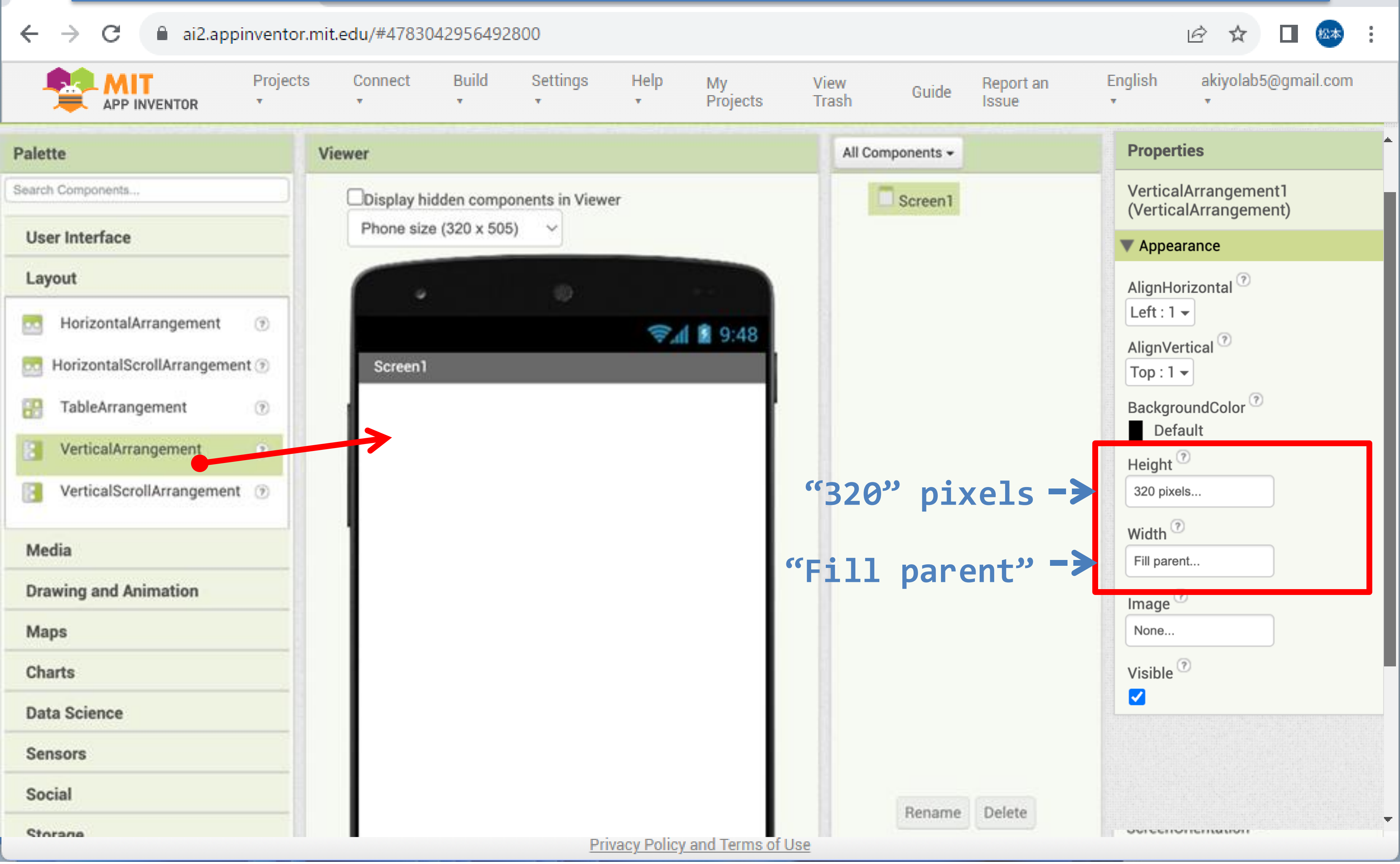
アプリのタイトルを変える


ai2.appinventor.mit.edu/#4783042956492800
MIT APP INVENTOR
Projects
Connect
Build
Settings
Help
My Projects
View Trash
Guide
Report an Issue
English
akiyolab5@gmail.com
Spinner
Switch
TextBox
TimePicker
WebViewer
Layout
Media
Drawing and Animation
Maps
Charts
Data Science
Sensors
Social
Storage
Connectivity
LEGO® MINDSTORMS®
Experimental
Rename
Delete
Media
Upload File ...
Application
Privacy Policy and Terms of Use

CloseScreenAnimation: Default
HighContrast: ☐
OpenScreenAnimation: Default
PrimaryColor: Default
PrimaryColorDark: Default
ScreenOrientation: Unspecified
Scrollable: ☐
ShowListsAsJson: ☑
ShowStatusBar: ☑
Title: 的当てゲーム
TitleVisible: ☑

今回は自由に変更OK ---→

VerticalArrangementを配置する


ai2.appinventor.mit.edu/#4783042956492800
MIT APP INVENTOR
Projects
Connect
Build
Settings
Help
My Projects
View Trash
Guide
Report an Issue
English
akiyolab5@gmail.com
Palette
Search Components...
User Interface
Layout
HorizontalArrangement
HorizontalScrollArrangement
TableArrangement
VerticalArrangement
VerticalScrollArrangement
Media
Drawing and Animation
Maps
Charts
Data Science
Sensors
Social
Storage
Viewer
Display hidden components in Viewer
Phone size (320 x 505)
9:48
Screen1
All Components
Screen1
Rename
Delete
Properties
Appearance
Privacy Policy and Terms of Use

VerticalArrangement1 (VerticalArrangement)

AlignHorizontal: Left : 1

AlignVertical: Top : 1

BackgroundColor: Default

Height: 320 pixels...

Width: Fill parent...

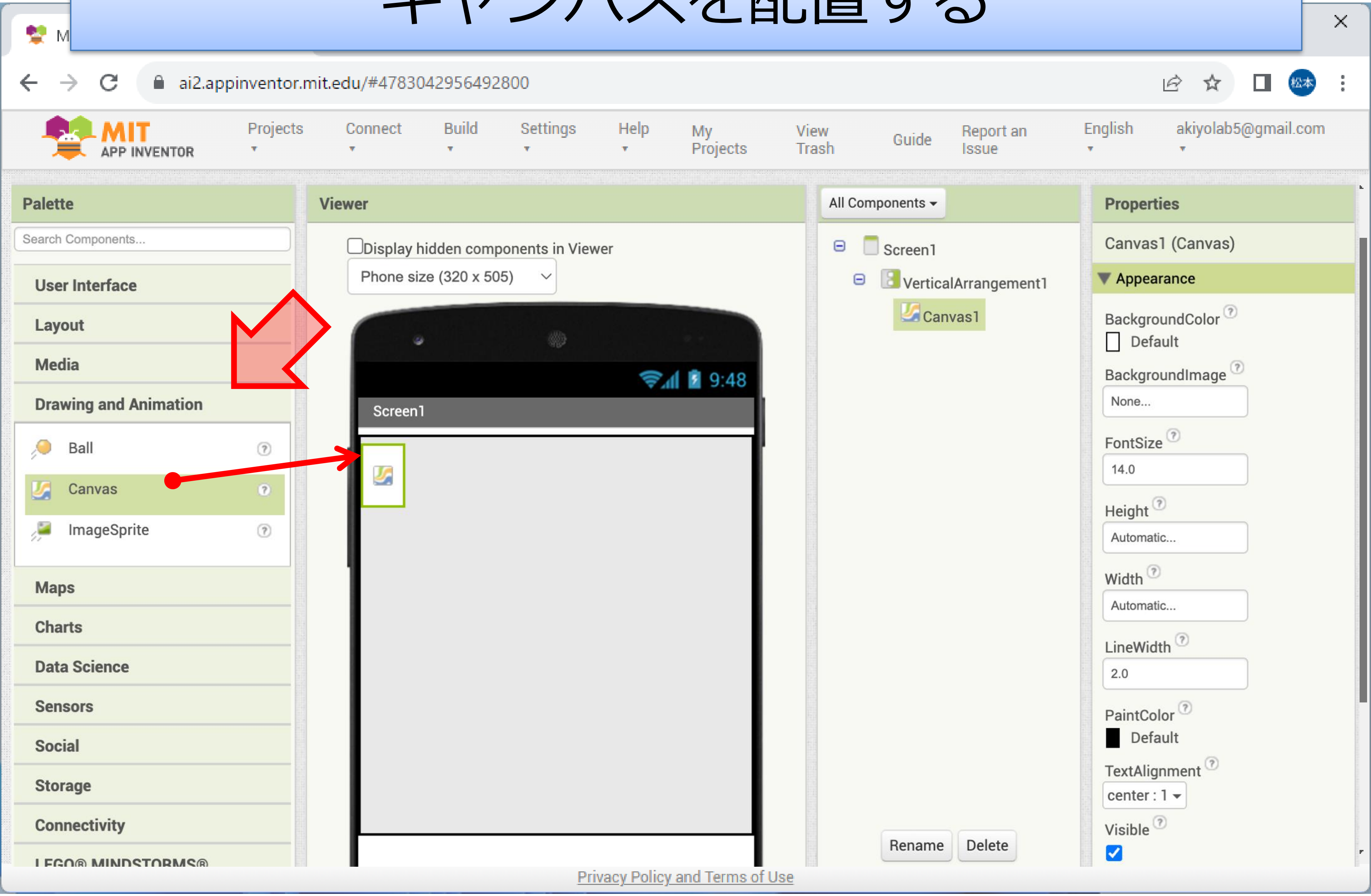
Image: None...

Visible: ☑

“320” pixels →

“Fill parent” →

キャンバスを配置する


ai2.appinventor.mit.edu/#4783042956492800
MIT APP INVENTOR
Projects
Connect
Build
Settings
Help
My Projects
View Trash
Guide
Report an Issue
English
akiyolab5@gmail.com
Palette
Search Components...
User Interface
Layout
Media
Drawing and Animation
Ball
Canvas
ImageSprite
Maps
Charts
Data Science
Sensors
Social
Storage
Connectivity
LEGO® MINDSTORMS®
Viewer
Screen1
9:48
All Components
Screen1
VerticalArrangement1
Canvas1
Rename
Delete
Properties
Privacy Policy and Terms of Use

☐ Display hidden components in Viewer

Phone size (320 x 505)

Canvas1 (Canvas)

Appearance

BackgroundColor: Default

BackgroundImage: None...

FontSize: 14.0

Height: Automatic...

Width: Automatic...

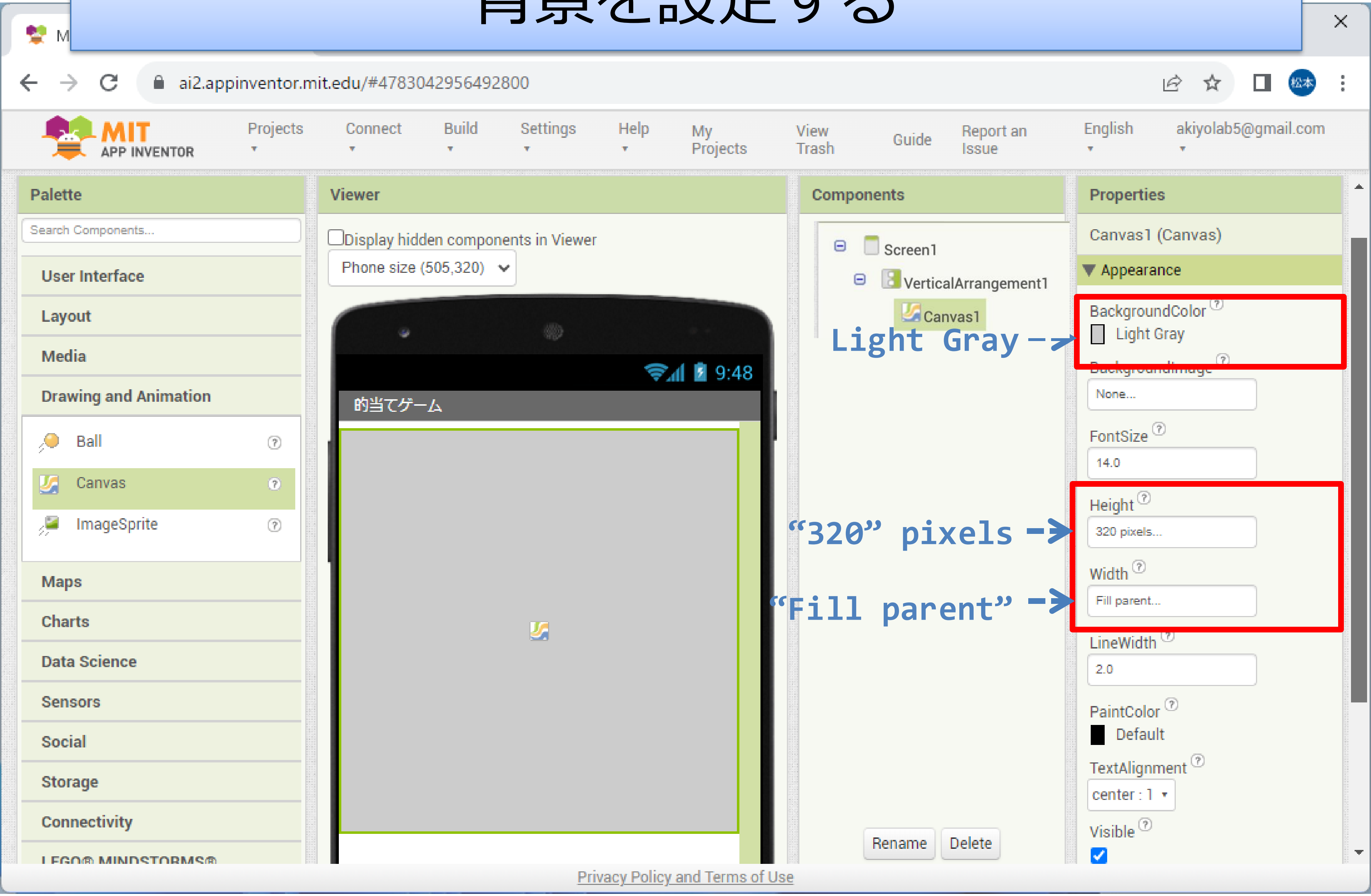
LineWidth: 2.0

PaintColor: Default

TextAlignment: center : 1

Visible: ☑

背景を設定する


ai2.appinventor.mit.edu/#4783042956492800
MIT APP INVENTOR
Projects
Connect
Build
Settings
Help
My Projects
View Trash
Guide
Report an Issue
English
akiyolab5@gmail.com
Palette
Search Components...
User Interface
Layout
Media
Drawing and Animation
Ball
Canvas
ImageSprite
Maps
Charts
Data Science
Sensors
Social
Storage
Connectivity
LEGO® MINDSTORMS®
Viewer
Display hidden components in Viewer
Phone size (505,320)
9:48
的当てゲーム
Components
Screen1
VerticalArrangement1
Canvas1
Rename
Delete
Properties
Privacy Policy and Terms of Use

Canvas1 (Canvas)

▼ Appearance

BackgroundColor: Light Gray

BackgroundImage: None...

FontSize: 14.0

Height: 320 pixels...

Width: Fill parent...

LineWidth: 2.0

PaintColor: Default

TextAlignment: center : 1

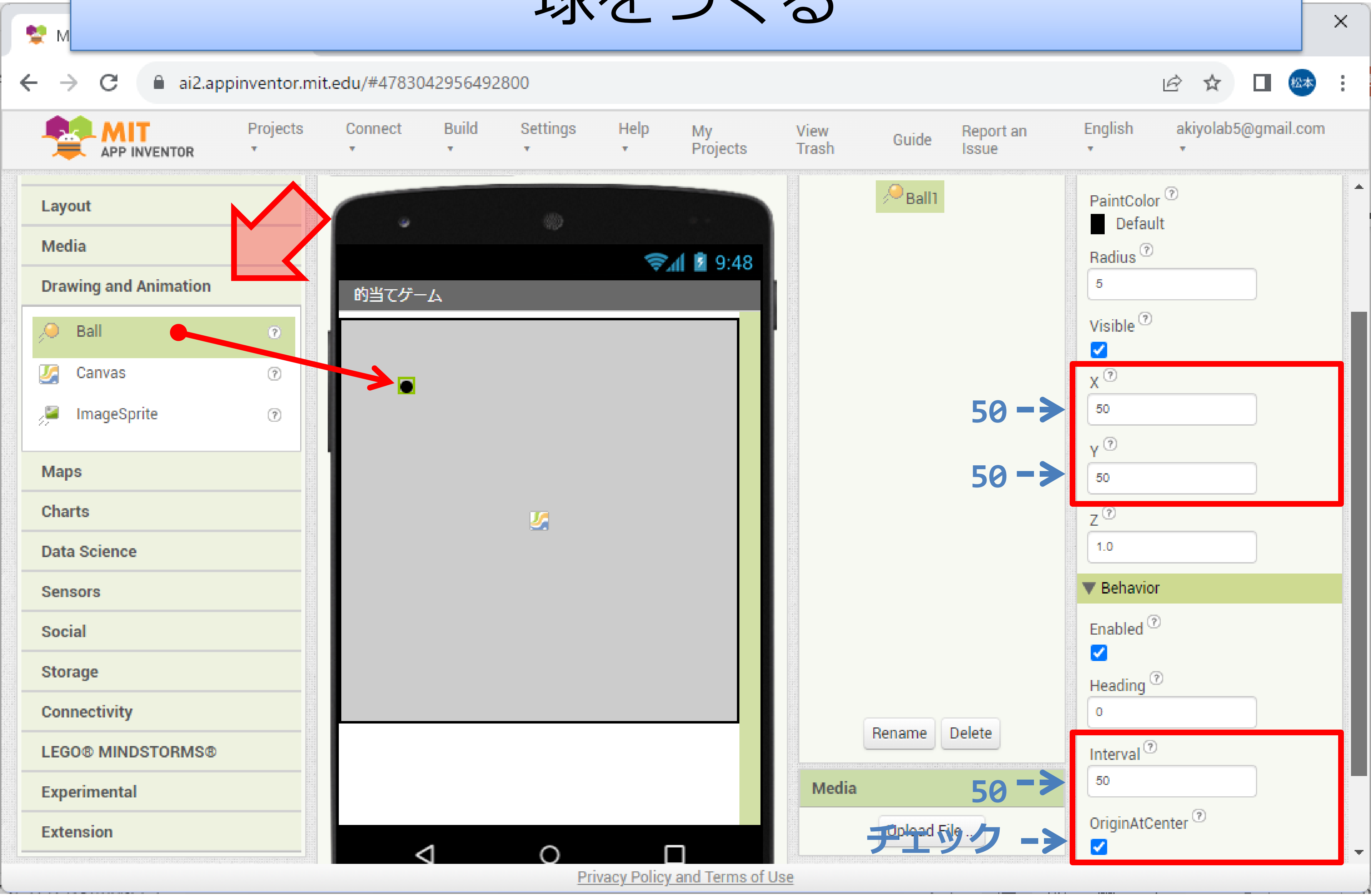
Visible: ☑

Light Gray →

“320” pixels →

“Fill parent” →

球をつくる


ai2.appinventor.mit.edu/#4783042956492800
MIT APP INVENTOR
Projects
Connect
Build
Settings
Help
My Projects
View Trash
Guide
Report an Issue
English
akiyolab5@gmail.com
Layout
Media
Drawing and Animation
Ball
Canvas
ImageSprite
Maps
Charts
Data Science
Sensors
Social
Storage
Connectivity
LEGO® MINDSTORMS®
Experimental
Extension
9:48
的当てゲーム
Ball1
Rename
Delete
Media
Upload File ...
Privacy Policy and Terms of Use

PaintColor ■ Default

Radius 5

Visible ☑

X 50

Y 50

Z 1.0

Behavior

Enabled ☑

Heading 0

Interval 50

OriginAtCenter ☑

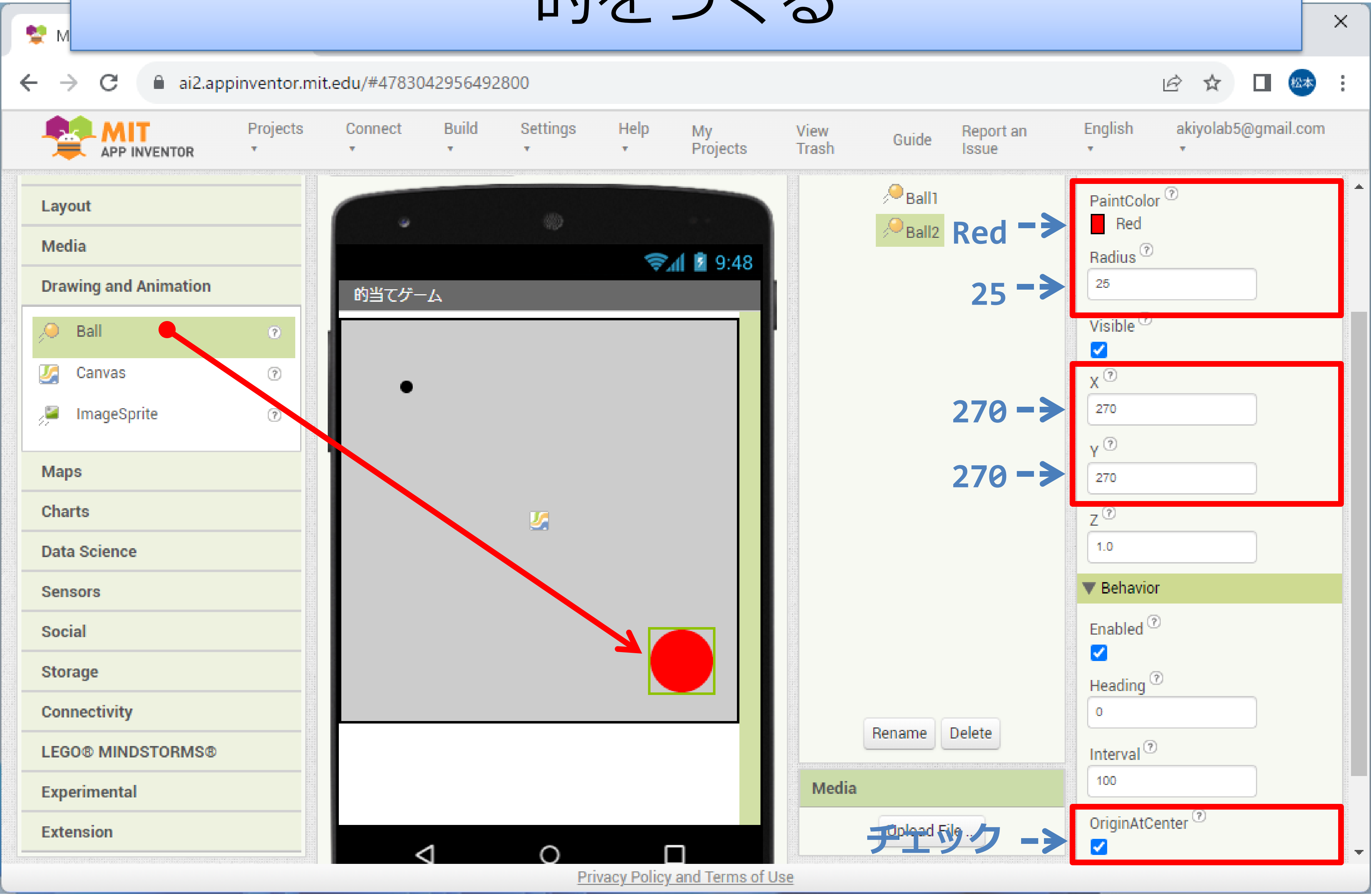
50 →（X）

50 →（Y）

50 →（Interval）

チェック →（OriginAtCenter）

的をつくる


ai2.appinventor.mit.edu/#4783042956492800
MIT APP INVENTOR
Projects
Connect
Build
Settings
Help
My Projects
View Trash
Guide
Report an Issue
English
akiyolab5@gmail.com
Layout
Media
Drawing and Animation
Ball
Canvas
ImageSprite
Maps
Charts
Data Science
Sensors
Social
Storage
Connectivity
LEGO® MINDSTORMS®
Experimental
Extension
9:48
的当てゲーム
Ball1
Ball2
Rename
Delete
Media
Upload File ...
Privacy Policy and Terms of Use

PaintColor: 🟥 Red

Radius: 25

Visible: ☑

X: 270

Y: 270

Z: 1.0

Behavior

Enabled: ☑

Heading: 0

Interval: 100

OriginAtCenter: ☑

Red →

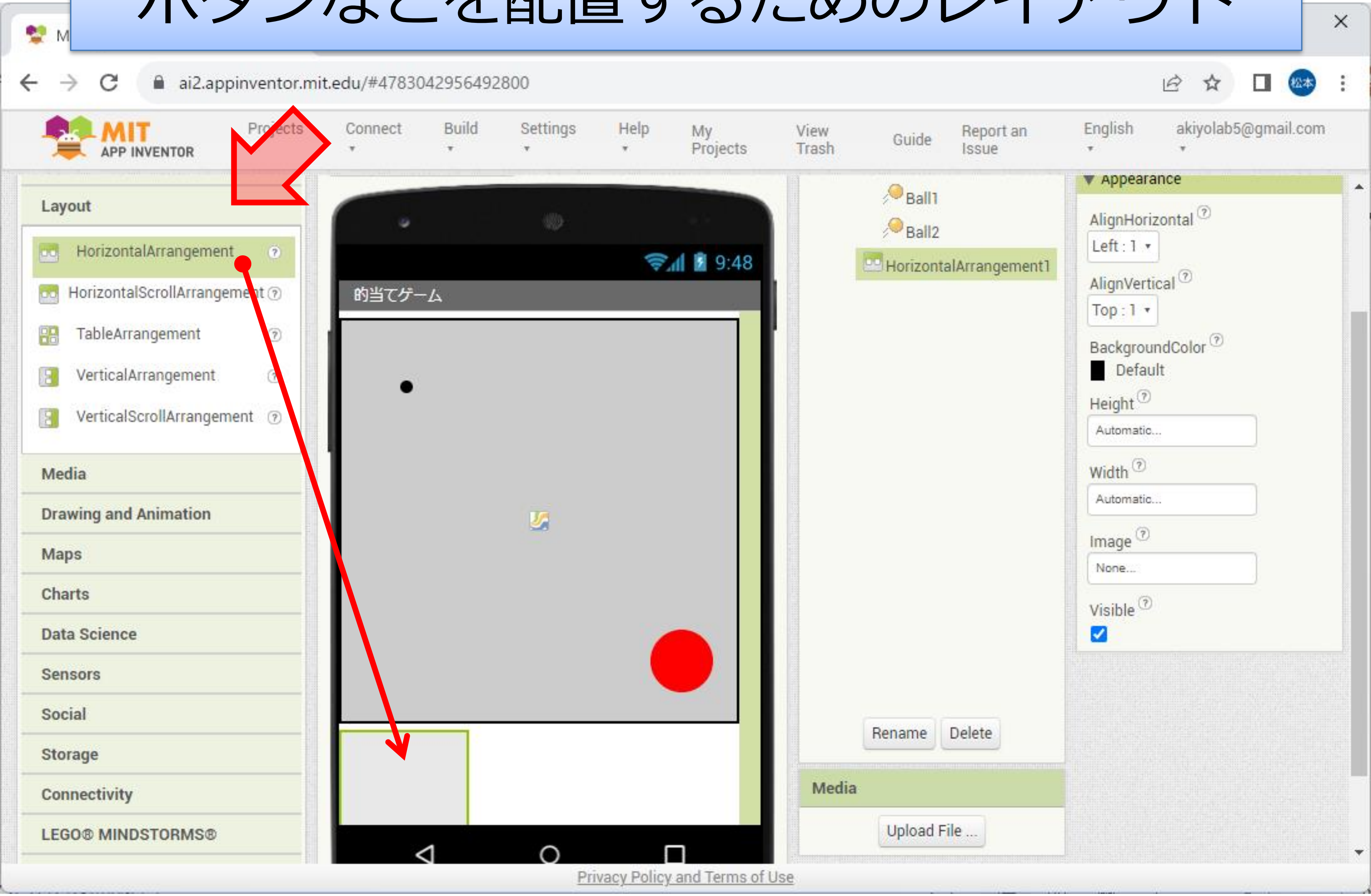
25 →

270 →

270 →

チェック →

ボタンなどを配置するためのレイアウト


ai2.appinventor.mit.edu/#4783042956492800
MIT APP INVENTOR
Projects
Connect
Build
Settings
Help
My Projects
View Trash
Guide
Report an Issue
English
akiyolab5@gmail.com
Layout
HorizontalArrangement
HorizontalScrollArrangement
TableArrangement
VerticalArrangement
VerticalScrollArrangement
Media
Drawing and Animation
Maps
Charts
Data Science
Sensors
Social
Storage
Connectivity
LEGO® MINDSTORMS®
9:48
的当てゲーム
Ball1
Ball2
HorizontalArrangement1
Rename
Delete
Media
Upload File ...
Appearance
Privacy Policy and Terms of Use

AlignHorizontal: Left : 1

AlignVertical: Top : 1

BackgroundColor: Default

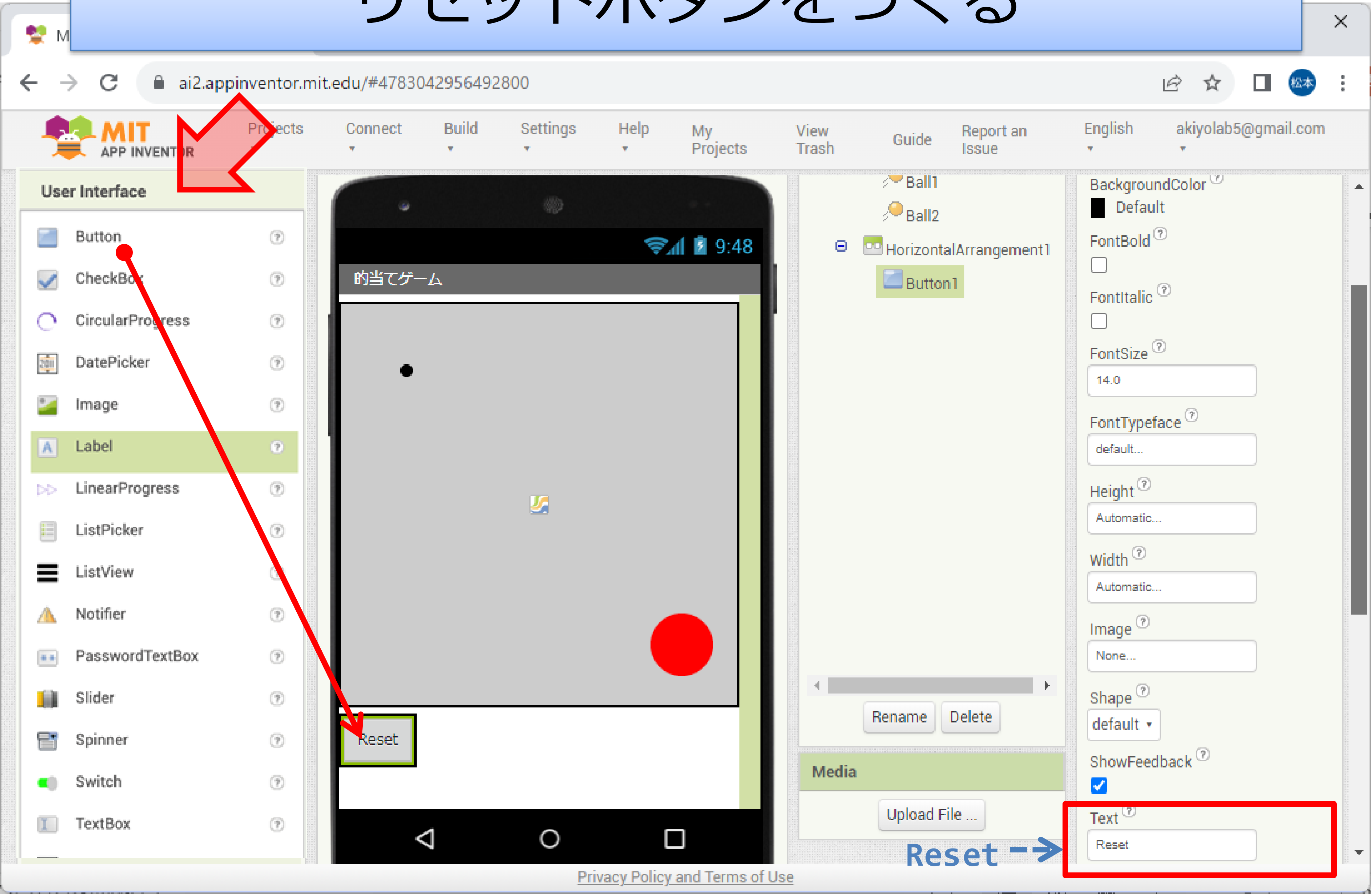
Height: Automatic...

Width: Automatic...

Image: None...

Visible: ☑

リセットボタンをつくる



ai2.appinventor.mit.edu/#4783042956492800
MIT APP INVENTOR
Projects
Connect
Build
Settings
Help
My Projects
View Trash
Guide
Report an Issue
English
akiyolab5@gmail.com
User Interface
Button
CheckBox
CircularProgress
DatePicker
Image
Label
LinearProgress
ListPicker
ListView
Notifier
PasswordTextBox
Slider
Spinner
Switch
TextBox
9:48
的当てゲーム
Reset
Ball1
Ball2
HorizontalArrangement1
Button1
Rename
Delete
Media
Upload File ...
Privacy Policy and Terms of Use

BackgroundColor: Default

FontBold: ☐

FontItalic: ☐

FontSize: 14.0

FontTypeface: default...

Height: Automatic...

Width: Automatic...

Image: None...

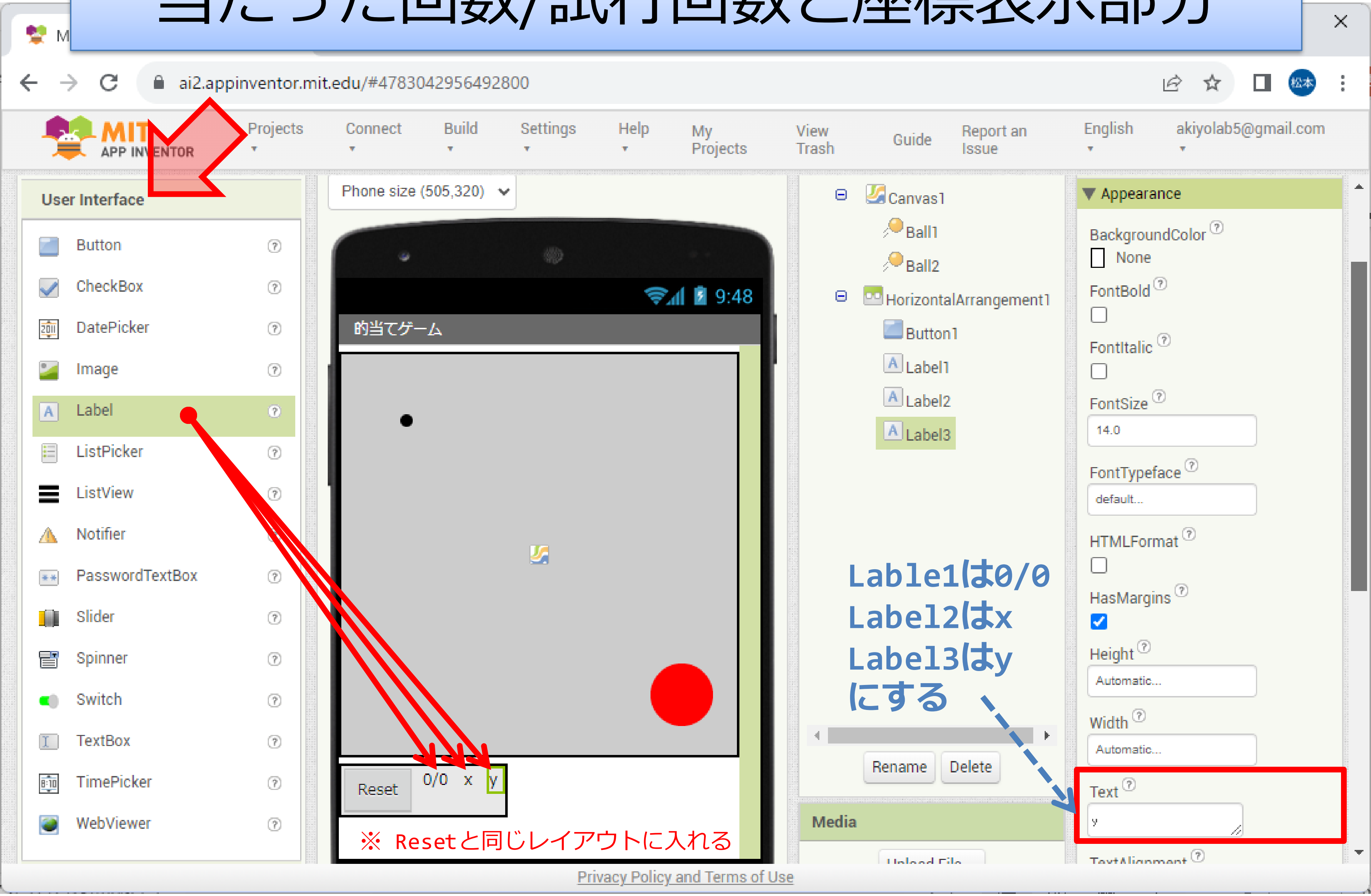
Shape: default

ShowFeedback: ☑

Text: Reset

Reset →

当たった回数/試行回数と座標表示部分


ai2.appinventor.mit.edu/#4783042956492800
MIT APP INVENTOR
Projects
Connect
Build
Settings
Help
My Projects
View Trash
Guide
Report an Issue
English
akiyolab5@gmail.com
User Interface
Button
CheckBox
DatePicker
Image
Label
ListPicker
ListView
Notifier
PasswordTextBox
Slider
Spinner
Switch
TextBox
TimePicker
WebViewer
Phone size (505,320)
9:48
的当てゲーム
Reset
0/0
x
y
Canvas1
Ball1
Ball2
HorizontalArrangement1
Button1
Label1
Label2
Label3
Rename
Delete
Media
Upload File
Appearance
Privacy Policy and Terms of Use

BackgroundColor: None

FontBold: ☐

FontItalic: ☐

FontSize: 14.0

FontTypeface: default...

HTMLFormat: ☐

HasMargins: ☑

Height: Automatic...

Width: Automatic...

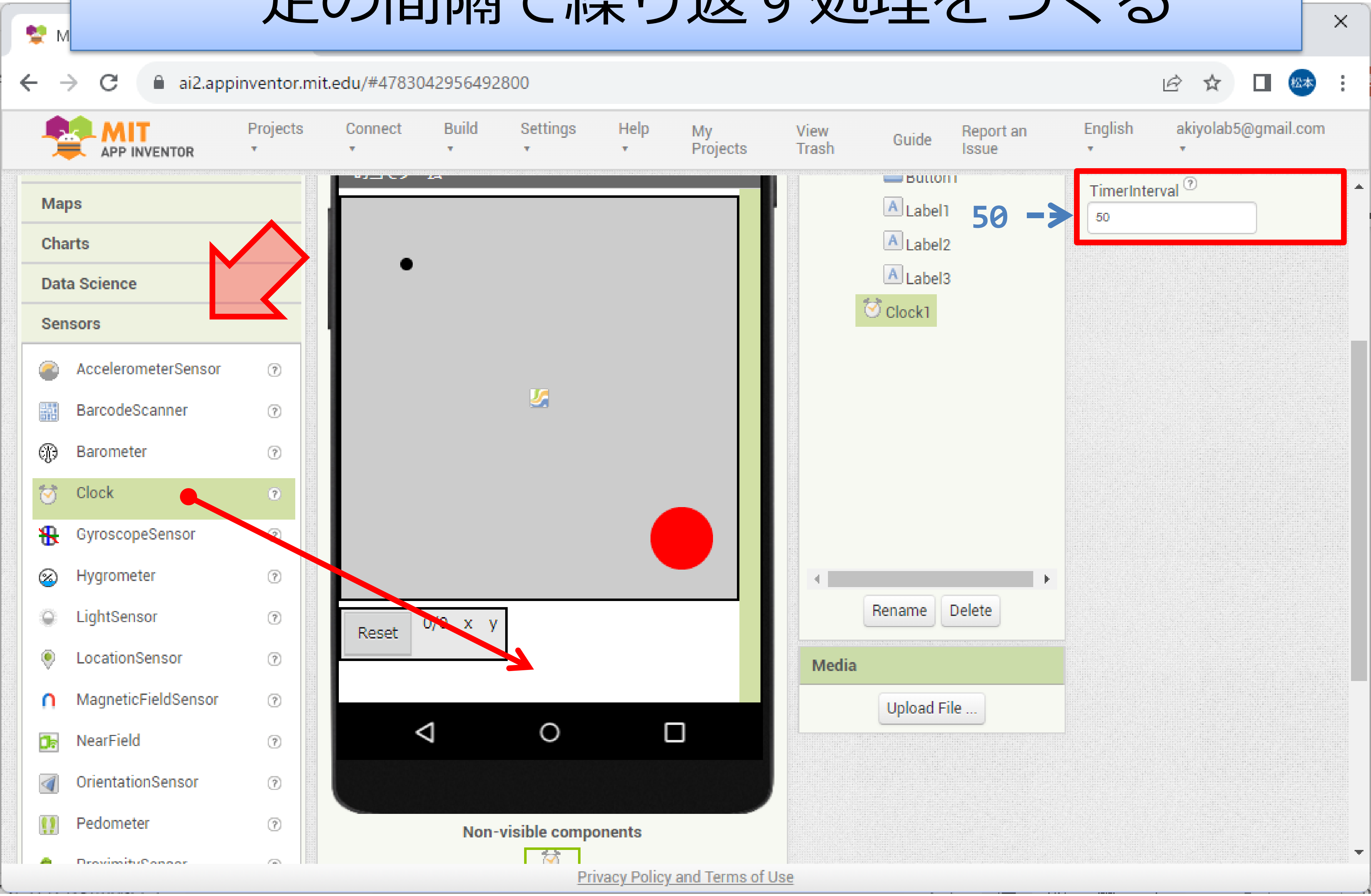
Text: y

TextAlignment

Lable1は0/0
Label2はx
Label3はy
にする

※ Resetと同じレイアウトに入れる

一定の間隔で繰り返す処理をつくる

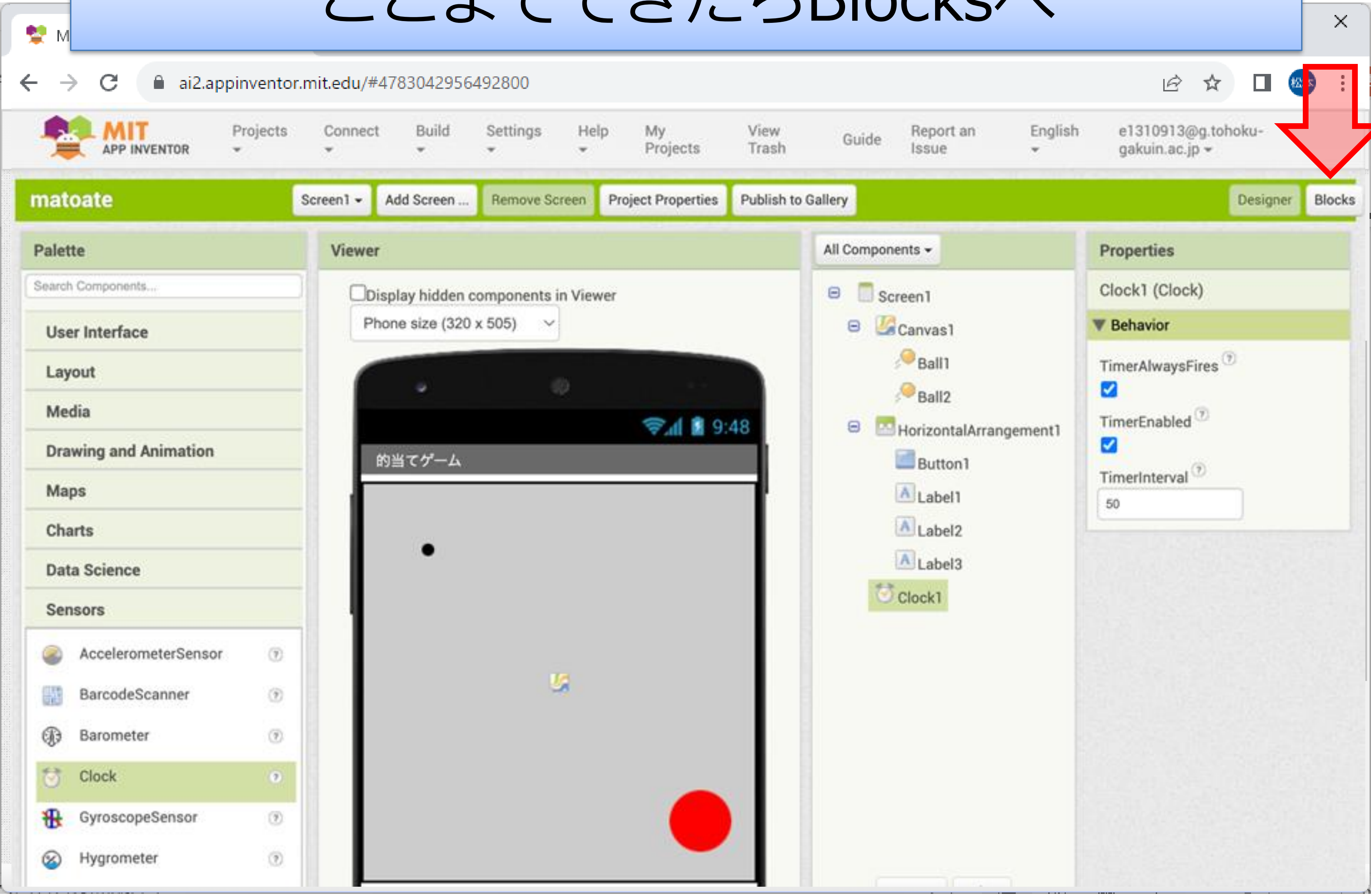


ai2.appinventor.mit.edu/#4783042956492800
MIT APP INVENTOR
Projects
Connect
Build
Settings
Help
My Projects
View Trash
Guide
Report an Issue
English
akiyolab5@gmail.com
Maps
Charts
Data Science
Sensors
AccelerometerSensor
BarcodeScanner
Barometer
Clock
GyroscopeSensor
Hygrometer
LightSensor
LocationSensor
MagneticFieldSensor
NearField
OrientationSensor
Pedometer
ProximitySensor
Reset
0/0
x
y
Non-visible components
Button1
Label1
Label2
Label3
Clock1
Rename
Delete
Media
Upload File ...
Privacy Policy and Terms of Use

50 →

TimerInterval: 50

ここまでできたらBlocksへ



ai2.appinventor.mit.edu/#4783042956492800
MIT APP INVENTOR
Projects
Connect
Build
Settings
Help
My Projects
View Trash
Guide
Report an Issue
English
e1310913@g.tohoku-gakuin.ac.jp
matoate
Screen1
Add Screen ...
Remove Screen
Project Properties
Publish to Gallery
Designer
Blocks
Palette
Search Components...
User Interface
Layout
Media
Drawing and Animation
Maps
Charts
Data Science
Sensors
AccelerometerSensor
BarcodeScanner
Barometer
Clock
GyroscopeSensor
Hygrometer
Viewer
Display hidden components in Viewer
Phone size (320 x 505)
9:48
的当てゲーム
All Components
Screen1
Canvas1
Ball1
Ball2
HorizontalArrangement1
Button1
Label1
Label2
Label3
Clock1
Properties
Clock1 (Clock)
Behavior

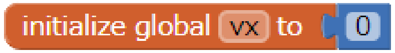
TimerAlwaysFires ☑

TimerEnabled ☑

TimerInterval 50

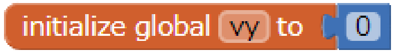
アプリ起動時の初期化とリセット処理

(Built-in)
Variables



```
initialize global vx to 0
```

Clock1.Timerが1回実行されるごとのX軸方向の移動量



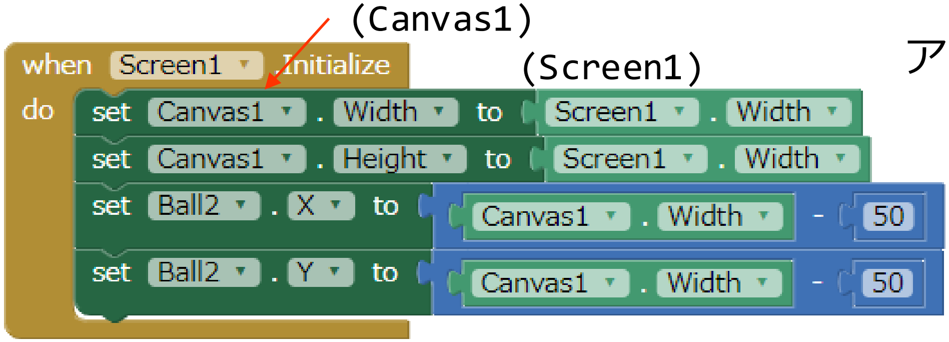
```
initialize global vy to 0
```

Clock1.Timerが1回実行されるごとのY軸方向の移動量

(Screen1)

(Canvas1)

(Screen1)

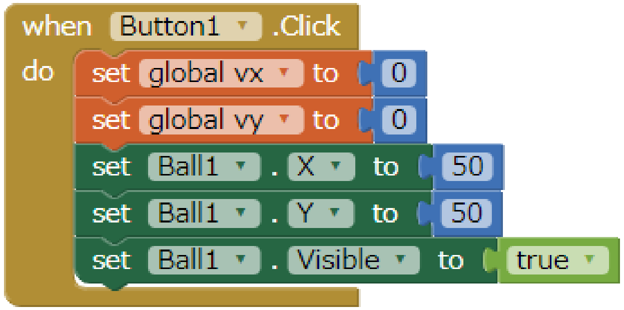


```
when Screen1 .Initialize
do  set Canvas1 . Width to Screen1 . Width
    set Canvas1 . Height to Screen1 . Width
    set Ball2 . X to Canvas1 . Width - 50
    set Ball2 . Y to Canvas1 . Width - 50
```

アプリが起動したら（初期化）
キャンバスの幅を画面の幅に
キャンバスの高さも（正方形）

的のx座標は右下から50ずらす
的のy座標も同様に

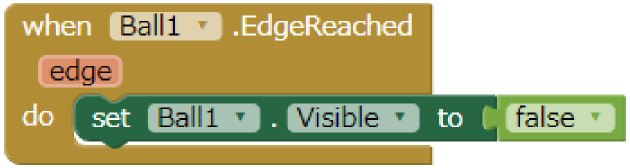
(Button1)



```
when Button1 .Click
do  set global vx to 0
    set global vy to 0
    set Ball1 . X to 50
    set Ball1 . Y to 50
    set Ball1 . Visible to true
```

リセットボタンが押されたら
球は動かないようにする
球のx座標・y座標は初期位置に戻す
球は見えるようにする

(Ball1)

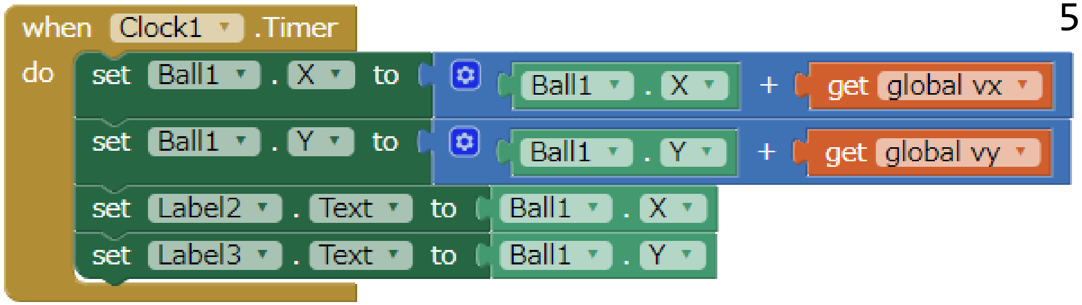


```
when Ball1 .EdgeReached
  edge
do  set Ball1 . Visible to false
```

球が画面端に当たったときには
見えなくする

※ Canvas1とかついているBlockはそれぞれのcomponentの場所にある

球を飛ばす処理



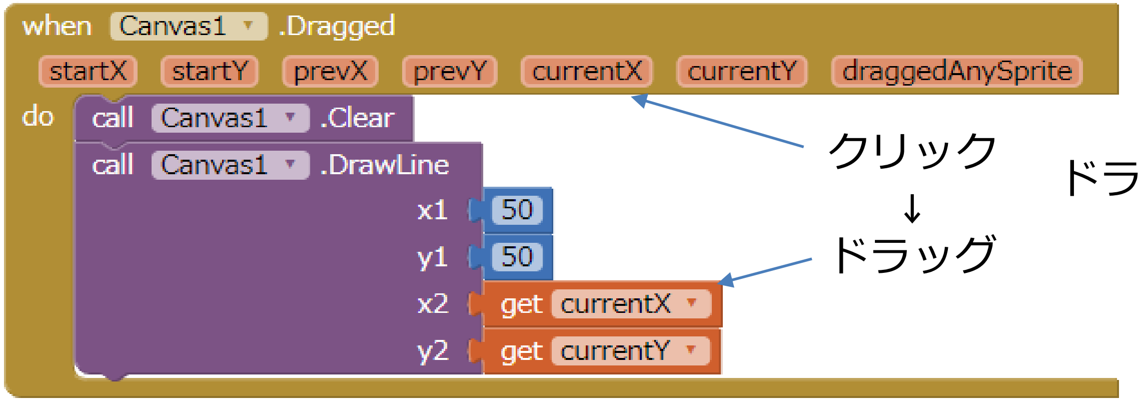
```
when Clock1 .Timer
do  set Ball1 . X to  Ball1 . X + get global vx
    set Ball1 . Y to  Ball1 . Y + get global vy
    set Label2 . Text to  Ball1 . X
    set Label3 . Text to  Ball1 . Y
```

50ミリ秒に1回行われる処理
球を移動する
球の座標を画面に表示する

$$x' = x + v_x \times \Delta t$$

$$y' = y + v_y \times \Delta t$$

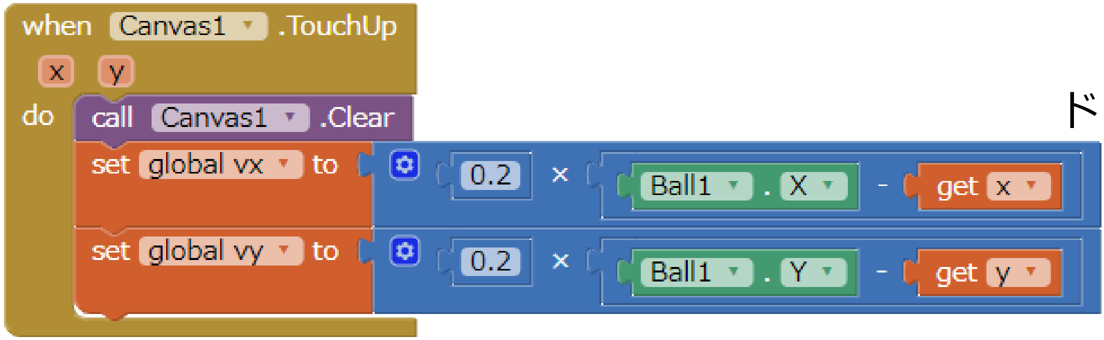
(Δt=1 と考える)



```
when Canvas1 .Dragged
 startX  startY  prevX  prevY  currentX  currentY  draggedAnySprite
do  call Canvas1 .Clear
    call Canvas1 .DrawLine
                       x1  50
                       y1  50
                       x2  get currentX
                       y2  get currentY
```

クリック
↓
ドラッグ

ドラッグしたとき
球の初期位置から現在地まで直線を描く

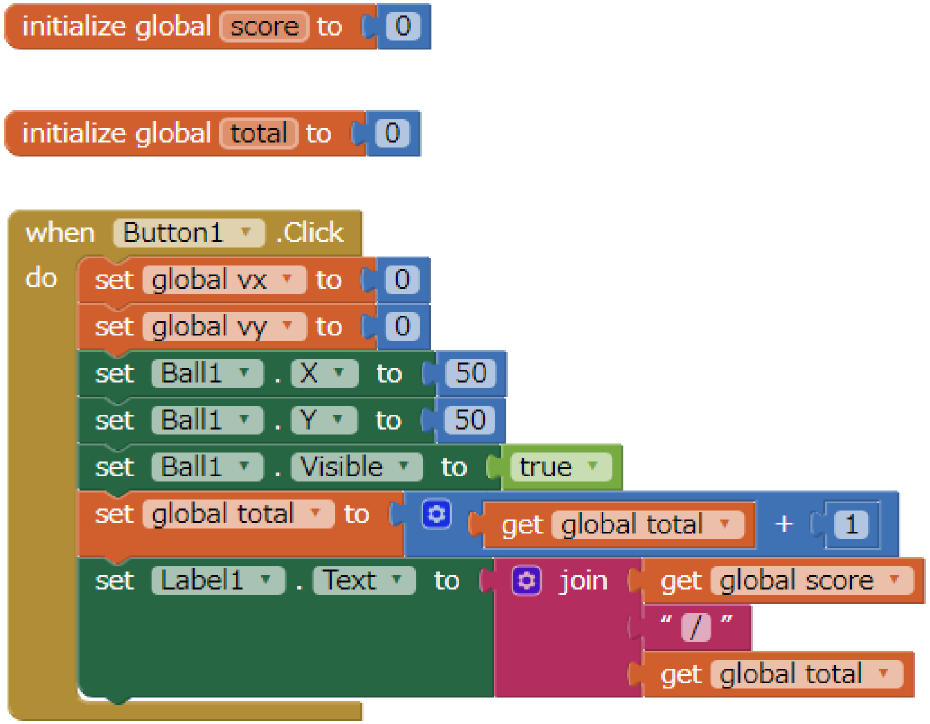


```
when Canvas1 .TouchUp
 x  y
do  call Canvas1 .Clear
    set global vx to  0.2 × ( Ball1 . X - get x )
    set global vy to  0.2 × ( Ball1 . Y - get y )
```

ドラッグした手を離したとき
直線を消す
球の移動量を設定する
（手を離した位置が球の位置より
離れている方が大きな値に）

ここまでできたら動作確認すること

当たった回数と試行回数の表示処理

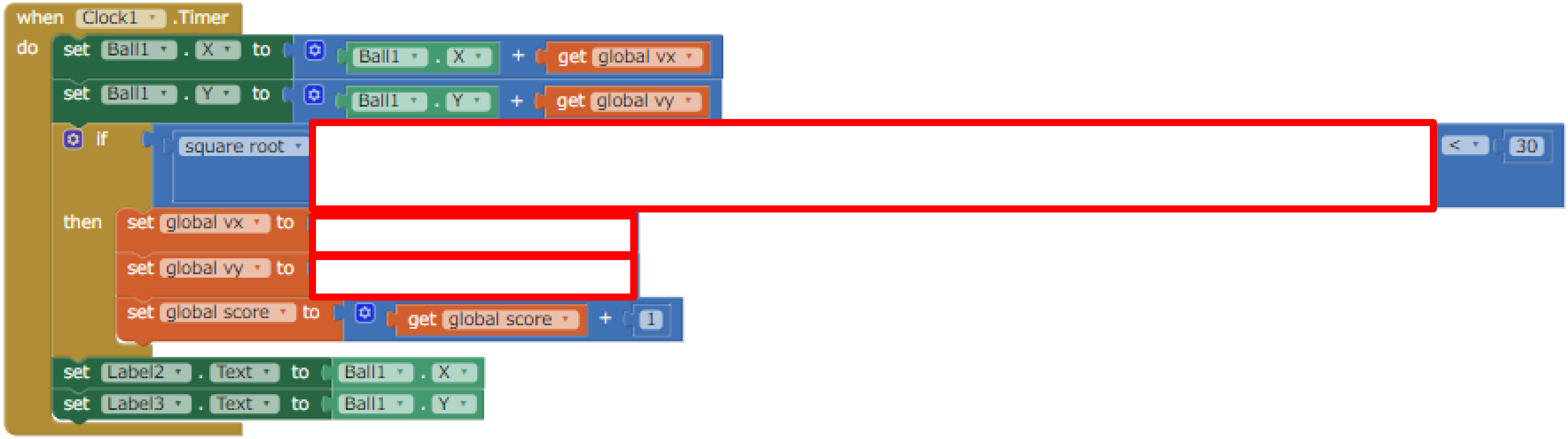


```
initialize global score to 0

initialize global total to 0

when Button1 .Click
do  set global vx to 0
    set global vy to 0
    set Ball1 . X to 50
    set Ball1 . Y to 50
    set Ball1 . Visible to true
    set global total to  get global total + 1
    set Label1 . Text to  join  get global score
                                " / "
                                get global total
```

（元からあるブロックに）
追加

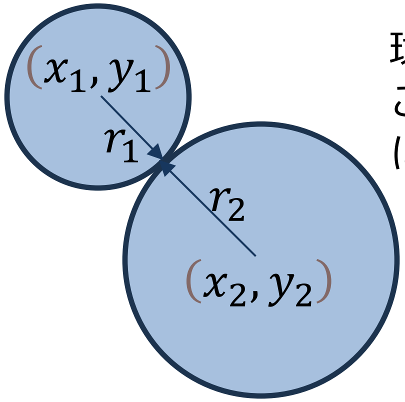
ここまでで再び動作確認すること

衝突したら球が跳ね返るようにする



```
when Clock1 .Timer
do  set Ball1 . X to  Ball1 . X + get global vx
    set Ball1 . Y to  Ball1 . Y + get global vy
    if  square root                  < 30
    then set global vx to
         set global vy to
         set global score to  get global score + 1
    set Label2 . Text to  Ball1 . X
    set Label3 . Text to  Ball1 . Y
```

追加

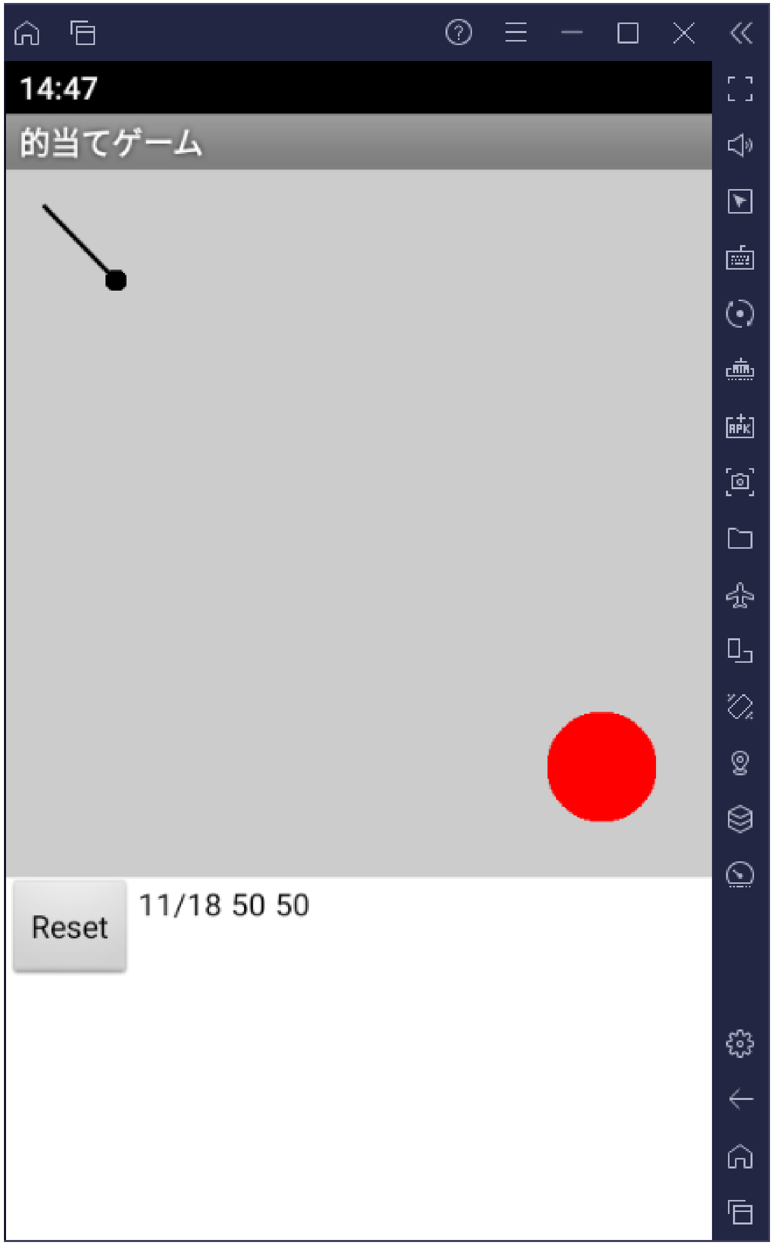


球と的の半径をr_1とr_2とし，中心の座標をそれぞれ(x_1, y_1)と(x_2, y_2)とする。このとき(x_1, y_1)と(x_2, y_2)の距離が$r_1 + r_2$より小さければ重なっていることになり「衝突している状態」と言える。

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} < r_1 + r_2$$

衝突処理を追加したら動作確認

実行画面



14:47

的当てゲーム

11/18 50 50

工夫してみよう

- Resetするたびに的のx座標が変わるようにしてみよう
- Resetするたびに的の大きさが変わるようにしてみよう
- 的が常に移動するようにしてみよう