

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

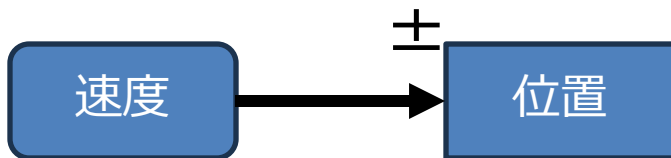
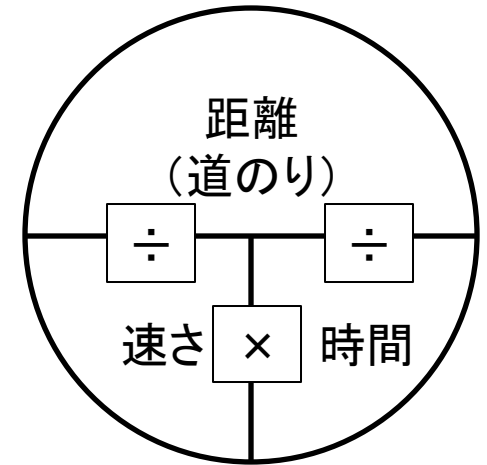
コンピュータ上で模擬するためには必要なこと

- ・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



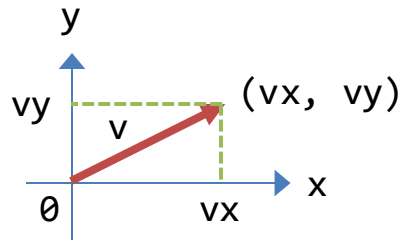
$$x' = x + v_x \times \Delta t$$

(元の x に速度 \times 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 \times 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



1秒間に動く量

t秒後の位置は？

$$x(t) = x(0) + v_x \times t$$

$$y(t) = y(0) + v_y \times t$$

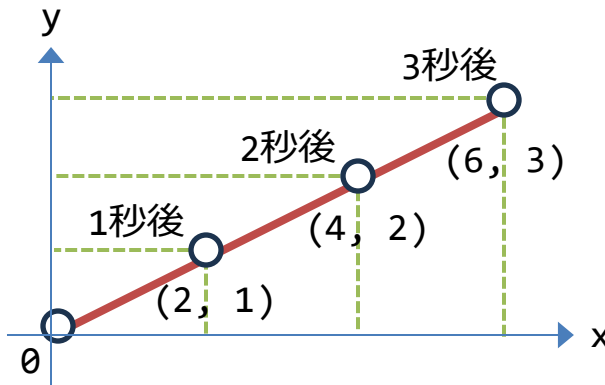
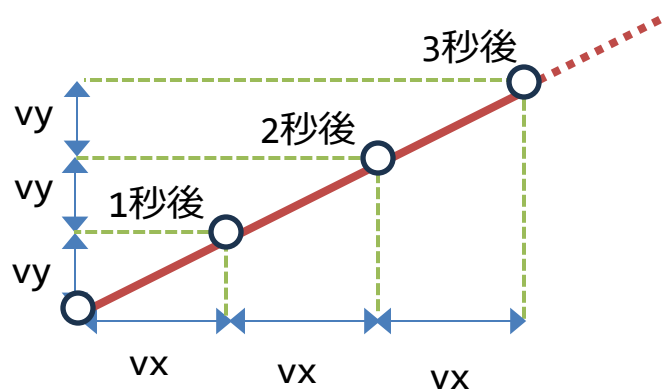
↓

速度が一定でないとダメ

↓

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

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



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

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

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
```

/ / / /

/ / / /

/ / / /

v_x (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

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

Name	Date Created	Date Modified
<input type="checkbox"/> Taiko	Nov 14, 2023, 1:39:06 PM	Nov 14, 2023, 2:01:17 PM
<input type="checkbox"/> Gakki	Nov 7, 2023, 11:23:20 PM	Nov 14, 2023, 1:16:02 PM
<input type="checkbox"/> Kazuate	2023, 11:43:06 AM	Nov 1, 2023, 11:50:14 AM
<input type="checkbox"/> Omikujj	2023, 1:08:14 PM	Nov 1, 2023, 11:07:51 AM

今回は「Matoate」

Create new App Inventor project

Project name: Matoate

Toolkit: Default

Theme: Device Default

Cancel

Classic
Device Default
Black Title Text
Dark

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

Privacy Policy and Terms of Use

アプリのタイトルを変える

The screenshot shows the MIT App Inventor web interface. At the top, there is a navigation bar with the MIT App Inventor logo and various menu items like Projects, Connect, Build, Settings, Help, My Projects, View Trash, Guide, Report an Issue, English, and a user email. Below the navigation bar is a left sidebar with a list of components: Spinner, Switch, TextBox, TimePicker, and WebViewer. The main workspace is divided into three sections: a central canvas showing a mobile app preview, a top-right section with 'Rename' and 'Delete' buttons, and a bottom-right section with an 'Upload File ...' button. On the right side, there is a properties panel with various settings. The 'Title' property is highlighted with a red box and contains the text '的当てゲーム'. A blue arrow points from the text '今回は自由に変更OK' to the 'Title' property field.

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

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

Rename Delete

Media Upload File ...

今回は自由に変更OK ---->

Privacy Policy and Terms of Use

VerticalArrangementを配置する

The screenshot shows the MIT App Inventor web interface. The top navigation bar includes the MIT App Inventor logo and various menu items like Projects, Connect, Build, Settings, Help, My Projects, View Trash, Guide, Report an Issue, English, and a user email. The main workspace is divided into four panels: Palette, Viewer, All Components, and Properties.

- Palette:** The 'Layout' section is expanded, and 'VerticalArrangement' is selected. A red arrow points from this component to the viewer.
- Viewer:** A mobile phone simulator is shown with 'Screen1' on the screen. A checkbox 'Display hidden components in Viewer' is checked. The phone size is set to 'Phone size (320 x 505)'.
- All Components:** A list shows 'Screen1' as the active component.
- Properties:** The 'Appearance' section is expanded. The 'Height' property is set to '320 pixels...' and the 'Width' property is set to 'Fill parent...'. These two properties are highlighted with a red box. Blue arrows point from the text '“320” pixels' to the Height field and '“Fill parent”' to the Width field.

At the bottom of the interface, there are 'Rename' and 'Delete' buttons for the selected component.

[Privacy Policy and Terms of Use](#)

キャンバスを配置する

The screenshot displays the MIT App Inventor web interface. At the top, a blue banner contains the title "キャンバスを配置する". Below it, the browser address bar shows the URL "ai2.appinventor.mit.edu/#4783042956492800". The main interface is divided into several panels:

- Palette:** A sidebar on the left with a search bar and categories like "User Interface", "Layout", "Media", "Drawing and Animation", "Maps", "Charts", "Data Science", "Sensors", "Social", "Storage", and "Connectivity". The "Canvas" component is highlighted in green in the "Drawing and Animation" section. A red arrow points from this component to the viewer.
- Viewer:** A central area showing a mobile phone screen labeled "Screen1". A red arrow points from the "Canvas" component in the palette to a small Canvas icon on the screen. Above the viewer, there is a checkbox for "Display hidden components in Viewer" and a dropdown menu for "Phone size (320 x 505)".
- All Components:** A panel on the right showing a tree view of components: "Screen1", "VerticalArrangement1", and "Canvas1".
- Properties:** A panel on the far right showing the properties for the selected "Canvas1 (Canvas)" component. The "Appearance" section is expanded, showing properties like "BackgroundColor", "BackgroundImage", "FontSize", "Height", "Width", "LineWidth", "PaintColor", "TextAlignment", and "Visible".

At the bottom of the interface, there is a "Privacy Policy and Terms of Use" link. The bottom right corner of the page shows the number "12".

背景を設定する

The screenshot displays the MIT App Inventor web interface. At the top, a blue banner contains the title "背景を設定する". Below it, the browser address bar shows the URL "ai2.appinventor.mit.edu/#4783042956492800". The interface is divided into several panels: "Palette" on the left with categories like "User Interface", "Layout", "Media", "Drawing and Animation", "Maps", "Charts", "Data Science", "Sensors", "Social", "Storage", and "Connectivity"; "Viewer" in the center showing a mobile phone simulation with a "Canvas" component highlighted; "Components" on the right showing a tree view with "Screen1", "VerticalArrangement1", and "Canvas1"; and "Properties" on the far right. The "Properties" panel for "Canvas1 (Canvas)" has two red boxes highlighting the "Appearance" section (containing "BackgroundColor" set to "Light Gray") and the "Size" section (containing "Height" set to "320 pixels..." and "Width" set to "Fill parent..."). Blue arrows point from the text "Light Gray" and "320" pixels to their respective fields. The "Width" field is labeled "Fill parent". At the bottom, there are "Rename" and "Delete" buttons. The footer contains "Privacy Policy and Terms of Use".

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)

Canvas1

Components

- Screen1
- VerticalArrangement1
- Canvas1

Properties

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

Rename Delete

Privacy Policy and Terms of Use

球をつくる

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

的当てゲーム

Ball1

PaintColor[?]
Default

Radius[?]
5

Visible[?]

X[?]
50

Y[?]
50

Z[?]
1.0

▼ Behavior

Enabled[?]

Heading[?]
0

Interval[?]
50

OriginAtCenter[?]

Rename Delete

Media 50 →

アップロード →

Privacy Policy and Terms of Use

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

Ball1
Ball2 Red →
25 →
270 →
270 →

PaintColor[?]
Red
Radius[?]
25
Visible[?]

X[?]
270
Y[?]
270
Z[?]
1.0
▼ Behavior
Enabled[?]

Heading[?]
0
Interval[?]
100
OriginAtCenter[?]

Rename Delete

Upload File → チェック

Privacy Policy and Terms of Use

ボタンなどを配置するためのレイアウト

The screenshot displays the MIT App Inventor web interface. At the top, a navigation bar includes 'Projects', 'Connect', 'Build', 'Settings', 'Help', 'My Projects', 'View Trash', 'Guide', 'Report an Issue', 'English', and the user email 'akiyolab5@gmail.com'. The main workspace is divided into three sections: a left sidebar, a central canvas, and a right-hand properties panel.

The left sidebar contains several categories: 'Layout', 'Media', 'Drawing and Animation', 'Maps', 'Charts', 'Data Science', 'Sensors', 'Social', 'Storage', 'Connectivity', and 'LEGO® MINDSTORMS®'. Under the 'Layout' category, 'HorizontalArrangement' is selected and highlighted in green. A red arrow points from the 'Projects' menu to this selection. Another red arrow points from the 'HorizontalArrangement' selection to a corresponding widget on the canvas.

The central canvas shows a mobile phone interface with the title '的当てゲーム' (Target Game). It features a black dot in the upper left and a large red circle in the lower right. A grey rectangular widget is positioned at the bottom of the screen, which is the 'HorizontalArrangement' widget selected in the sidebar.

The right-hand properties panel is titled 'Appearance' and lists various settings: 'AlignHorizontal' (Left: 1), 'AlignVertical' (Top: 1), 'BackgroundColor' (Default), 'Height' (Automatic...), 'Width' (Automatic...), 'Image' (None...), and 'Visible' (checked). Below the panel, there are 'Rename' and 'Delete' buttons, and a 'Media' section with an 'Upload File ...' button.

At the bottom of the interface, there is a link for 'Privacy Policy and Terms of Use'.

リセットボタンをつくる

The screenshot shows the MIT App Inventor web interface. At the top, a blue banner contains the title "リセットボタンをつくる". Below it, the browser address bar shows the URL "ai2.appinventor.mit.edu/#4783042956492800". The main interface is divided into several sections:

- User Interface:** A list of UI components on the left. A red arrow points from the "Button" component to a "Reset" button placed on a mobile device preview in the center. The mobile preview shows a game titled "的当てゲーム" (Target Game) with a red circle on the screen.
- Properties Panel:** On the right, the properties for the selected "Button1" are shown. The "Text" property is highlighted with a red box and contains the text "Reset". A blue arrow points from the text "Reset" in the properties panel to the "Reset" button on the mobile preview.
- Media Panel:** Below the properties panel, there is a "Media" section with an "Upload File ..." button.

At the bottom of the interface, there is a link for "Privacy Policy and Terms of Use".

当たった回数/試行回数と座標表示部分

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
- WebView

Phone size (505,320)

Canvas1

- Ball1
- Ball2
- HorizontalArrangement1
- Button1
- Label1
- Label2
- Label3**

Appearance

BackgroundColor

FontBold

FontItalic

FontSize

FontTypeface

HTMLFormat

HasMargins

Height

Width

Text

0/0 x y

Reset

Label1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

Privacy Policy and Terms of Use

一定の間隔で繰り返す処理をつくる

The screenshot displays the MIT App Inventor web interface. At the top, a blue banner contains the title "一定の間隔で繰り返す処理をつくる". Below it, the browser address bar shows the URL "ai2.appinventor.mit.edu/#4783042956492800". The interface includes a navigation menu with options like "Projects", "Connect", "Build", "Settings", "Help", "My Projects", "View Trash", "Guide", "Report an Issue", "English", and "akiyolab5@gmail.com".

The main workspace is divided into three sections:

- Left Panel (Sensors):** A list of sensors including AccelerometerSensor, BarcodeScanner, Barometer, Clock, GyroscopeSensor, Hygrometer, LightSensor, LocationSensor, MagneticFieldSensor, NearField, OrientationSensor, Pedometer, and ProximitySensor. The "Clock" sensor is highlighted with a red dot, and a red arrow points from this dot to a "Clock1" component on the mobile device canvas.
- Mobile Device Canvas:** A central area showing a mobile phone interface with a grey screen, a red circle, and a "Reset" button. A red arrow points from the "Clock" sensor in the left panel to the "Clock1" component on the canvas.
- Right Panel (Properties):** A list of components including "Button 1" (containing Label1, Label2, Label3) and "Clock1". The "TimerInterval" property of "Clock1" is highlighted with a red box and set to the value "50". A blue arrow points from the "50" value in the component list to the "TimerInterval" input field.

At the bottom of the interface, there is a "Non-visible components" section and a link for "Privacy Policy and Terms of Use".

ここまでできたらBlocksへ

The screenshot displays the MIT App Inventor web interface. At the top, a blue banner contains the text "ここまでできたらBlocksへ". Below this is the browser address bar showing the URL "ai2.appinventor.mit.edu/#4783042956492800". The interface includes a navigation menu with options like "Projects", "Connect", "Build", "Settings", "Help", "My Projects", "View Trash", "Guide", "Report an Issue", "English", and a user profile "e1310913@g.tohoku-gakuin.ac.jp". A red arrow points to the user profile icon in the top right corner.

The main workspace is divided into several panels:

- Palette:** A sidebar on the left with a search bar and categories: "User Interface", "Layout", "Media", "Drawing and Animation", "Maps", "Charts", "Data Science", and "Sensors". The "Clock" component is highlighted in the "Sensors" category.
- Viewer:** A central area showing a mobile phone simulation. It has a checkbox for "Display hidden components in Viewer" and a dropdown for "Phone size (320 x 505)". The phone screen displays the text "的当てゲーム" and a red circle at the bottom right.
- All Components:** A panel on the right showing a tree view of the app's components: "Screen1" (containing "Canvas1" with "Ball1" and "Ball2", "HorizontalArrangement1" with "Button1", "Label1", "Label2", and "Label3", and "Clock1").
- Properties:** A panel on the far right showing the properties for the selected "Clock1 (Clock)" component. Under the "Behavior" section, "TimerAlwaysFires" and "TimerEnabled" are checked, and "TimerInterval" is set to "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) when Screen1 Initialize (Screen1) アプリが起動したら (初期化)
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
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

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

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

($\Delta t=1$ 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れたとき

直線を消す

球の移動量を設定する

(手を離れ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))
```

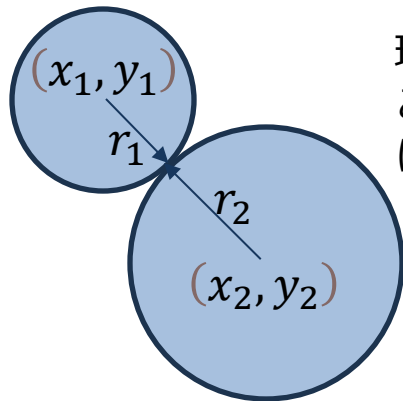
(元からあるブロックに)
追加

ここまですでに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 of (Ball1.X - Ball2.X)^2 + (Ball1.Y - Ball2.Y)^2 < 30
  then
    set global vx to get global vx * -1
    set global vy to get global vy * -1
    set global score to get global score + 1
  set Label2.Text to Ball1.X
  set Label3.Text to Ball1.Y
```

追加

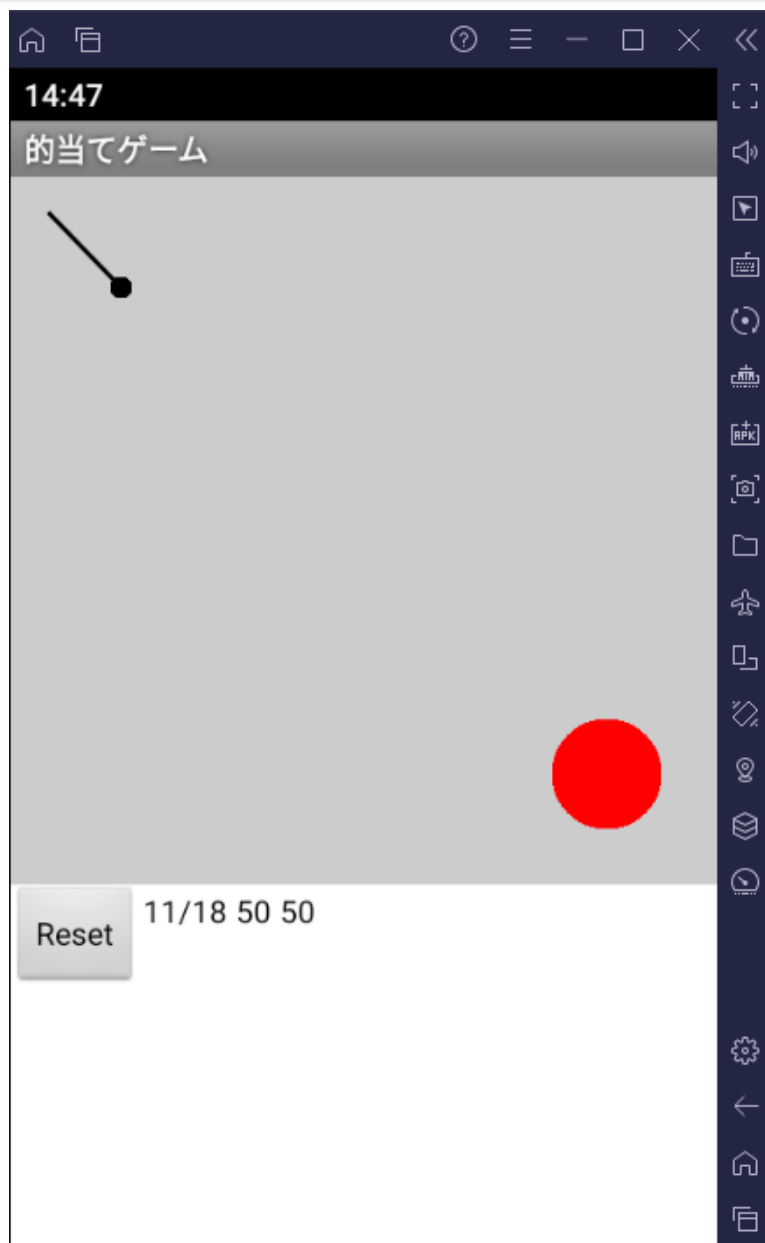


球と的の半径を r_1 と r_2 とし、中心の座標をそれぞれ (x_1, y_1) と (x_2, y_2) とする。このとき (x_1, y_1) と (x_2, y_2) の距離が $r_1 + r_2$ より小さければ重なっていることになり「衝突している状態」と言える。

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

衝突処理を追加したら動作確認

実行画面



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