

# プログラミング概論

第10回 2023年11月22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

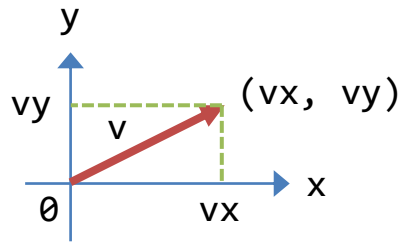
自然や社会に見られる現象をコンピュータ上で模擬的に再現すること → 分析や予測, ゲームに有用

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

# 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？

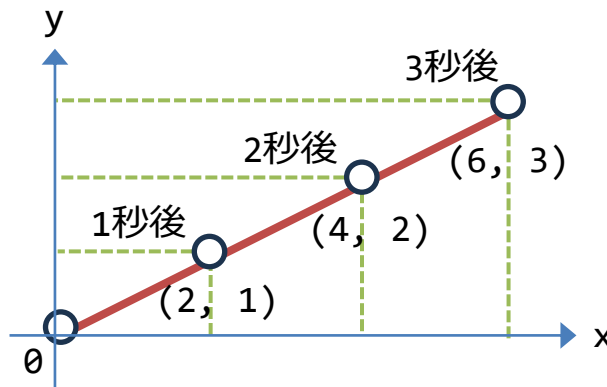
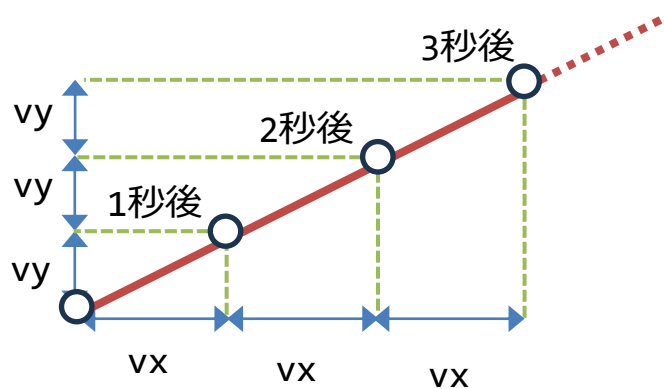
$$x(t) = x(0) + v_x \times t$$

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

↓  
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



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

Projects

<input type="checkbox"/> 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	Nov 1, 2023, 11:43:06 AM	Nov 1, 2023, 11:50:14 AM
<input type="checkbox"/> Omikuji	Oct 25, 2023, 1:08:14 PM	Nov 1, 2023, 11:07:51 AM

Create new App Inventor project

Project name: Matoate

今回は「Matoate」

Cancel OK

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 profile. Below the navigation bar is a sidebar with categories like Layout, Media, Drawing and Animation, Maps, Charts, Data Science, Sensors, Social, Storage, Connectivity, LEGO® MINDSTORMS®, and Experimental. The main workspace is divided into three sections: a mobile device simulator on the left, a central canvas with a 'Media' component and an 'Upload File ...' button, and a right-hand properties panel. The properties panel is currently set to the 'Application' component, and the 'Title' property is highlighted with a red box. The title is currently '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.

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

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

Media Upload File ...

今回は自由に変更OK ---->

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 the banner, the browser address bar shows the URL "ai2.appinventor.mit.edu/#4783042956492800". The interface is divided into several panels:

- Palette:** A sidebar on the left containing various component categories such as "User Interface", "Layout", "Media", "Drawing and Animation", "Maps", "Charts", "Data Science", "Sensors", "Social", "Storage", and "Connectivity". The "Canvas" component is highlighted in green, and a red arrow points from it to the viewer.
- Viewer:** A central panel showing a mobile phone simulation. The phone screen displays the text "的当てゲーム". A red arrow points from the "Canvas" component in the palette to a small canvas icon on the phone screen, which is also highlighted with a green box.
- Components:** A panel on the right showing the components currently on the screen, including "Screen1" and "Canvas1".
- Properties:** A panel on the far right showing the properties for the selected "Canvas1" 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 link for "Privacy Policy and Terms of Use".

# 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 on the screen; "Components" on the right showing a tree view with "Screen1" and "Canvas1"; and "Properties" on the far right. The "Properties" panel for "Canvas1" has two red boxes highlighting the "Appearance" section. The first box highlights the "BackgroundColor" property, which is set to "Light Gray". A blue arrow points from the text "Light Gray" to this property. The second box highlights the "Height" and "Width" properties. The "Height" is set to "320 pixels..." and the "Width" is set to "Fill parent...". Blue arrows point from the text "320" and "Fill" to these respective values. The "Canvas" component in the "Components" panel is also highlighted with a blue arrow pointing to the text "Light Gray".

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

Canvas1

Light Gray →

320 →

Fill →

Properties

Canvas1 (Canvas)

Appearance

BackgroundColor <sup>?</sup>

Light Gray

BackgroundImage <sup>?</sup>

None...

FontSize <sup>?</sup>

14.0

Height <sup>?</sup>

320 pixels...

Width <sup>?</sup>

Fill parent...

LineWidth <sup>?</sup>

2.0

PaintColor <sup>?</sup>

Default

TextAlignment <sup>?</sup>

center : 1

Visible <sup>?</sup>

Rename Delete

Privacy Policy and Terms of Use

# 球をつくる

The screenshot shows the MIT App Inventor web interface. At the top, a blue banner contains the text "球をつくる". Below it, the browser address bar shows the URL "ai2.appinventor.mit.edu/#4783042956492800". The interface includes a top navigation bar with options like "Projects", "Connect", "Build", "Settings", "Help", "My Projects", "View Trash", "Guide", "Report an Issue", "English", and "akiyolab5@gmail.com".

On the left, a sidebar lists various components under categories like "Layout", "Media", "Drawing and Animation", "Maps", "Charts", "Data Science", "Sensors", "Social", "Storage", "Connectivity", "LEGO® MINDSTORMS®", "Experimental", and "Extension". The "Drawing and Animation" section is expanded, showing "Ball", "Canvas", and "ImageSprite". A red arrow points from the "Ball" component to a small black square on the central mobile device preview screen.

The central preview screen shows a mobile phone with the title "的当てゲーム" (Target Game) and a grey background with a small blue icon in the center. A red arrow also points from the "Ball" component in the sidebar to this icon.

On the right, the "Properties" panel for the "Ball1" component is visible. It includes settings for "PaintColor" (Default), "Radius" (5), "Visible" (checked), "X" (50), "Y" (50), "Z" (1.0), "Behavior" (Enabled checked, Heading 0), "Interval" (50), and "OriginAtCenter" (checked). The "X" and "Y" fields, and the "Interval" field, are highlighted with red boxes. Blue arrows labeled "50" point to the "X" and "Y" fields. A blue arrow labeled "チェック" (Check) points to the "OriginAtCenter" checkbox.

At the bottom of the interface, there are "Rename" and "Delete" buttons, and a "Media" section with an "Upload File" button.

[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 (?)

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 blue banner contains the title "ボタンなどを配置するためのレイアウト". Below the banner, 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:

- Layout:** A sidebar on the left lists various layout types: HorizontalArrangement, HorizontalScrollArrangement, TableArrangement, VerticalArrangement, and VerticalScrollArrangement. A red arrow points from the "HorizontalArrangement" option to a red dot on the mobile app preview.
- Mobile App Preview:** A central smartphone screen displays a game titled "的当てゲーム" (Target Game). It features a black dot in the upper left and a red circle in the lower right. A red arrow points from the "HorizontalArrangement" option to a red dot on the bottom bar of the app.
- Properties Panel:** On the right, the "Appearance" panel is open, showing settings for "HorizontalArrangement1", including "AlignHorizontal" (Left: 1), "AlignVertical" (Top: 1), "BackgroundColor" (Default), "Height" (Automatic...), "Width" (Automatic...), "Image" (None...), and "Visible" (checked).

At the bottom of the interface, there are "Rename" and "Delete" buttons, and a "Media" section with an "Upload File ..." button. A "Privacy Policy and Terms of Use" link is visible at the very bottom.

# 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 interface includes a top navigation bar with options like "Projects", "Connect", "Build", "Settings", "Help", "My Projects", "View Trash", "Guide", "Report an Issue", "English", and "akiyolab5@gmail.com".

On the left, a component palette lists various widgets. The "Button" component is highlighted in green, and a red arrow points from it to a "Reset" button on the mobile app preview. The preview shows a mobile screen with the title "的当てゲーム" (Target Game) and a red circle on the right side. The "Reset" button is located at the bottom left of the screen.

On the right, the "Properties" panel for the selected "Button1" component is visible. The "Text" property is highlighted with a red box and contains the text "Reset". A blue arrow points from the "Reset" text in the preview to the "Text" property field. Other properties like "BackgroundColor", "FontBold", "FontSize", and "Image" are also visible.

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

# 当たった回数/試行回数と座標表示部分

The screenshot displays the MIT App Inventor web interface. The central area shows a mobile phone simulation titled "的当てゲーム" (Target Game). The phone screen displays a black dot at the top left, a red circle at the bottom right, and a status bar at the top with the time 9:48. Below the phone screen is a control panel with a "Reset" button and three input fields containing "0/0", "x", and "y".

On the left, the "User Interface" palette lists various components, with "Label" selected. A red dot on the "Label" component is connected by three red arrows to the "0/0", "x", and "y" input fields on the phone screen.

On the right, the "Appearance" properties panel is visible. The "Text" property is highlighted with a red box and contains the value "y". A blue dashed arrow points from the Japanese text below to this "Text" property.

The component palette on the right lists: Canvas1, Ball1, Ball2, HorizontalArrangement1, Button1, Label1, Label2, and Label3. Label3 is currently selected.

Blue Japanese text is overlaid on the right side of the interface:

Label1は0/0  
Label2はx  
Label3はy  
にする



# アプリ起動時の初期化とリセット処理

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

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

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

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

# 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

```
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
```

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

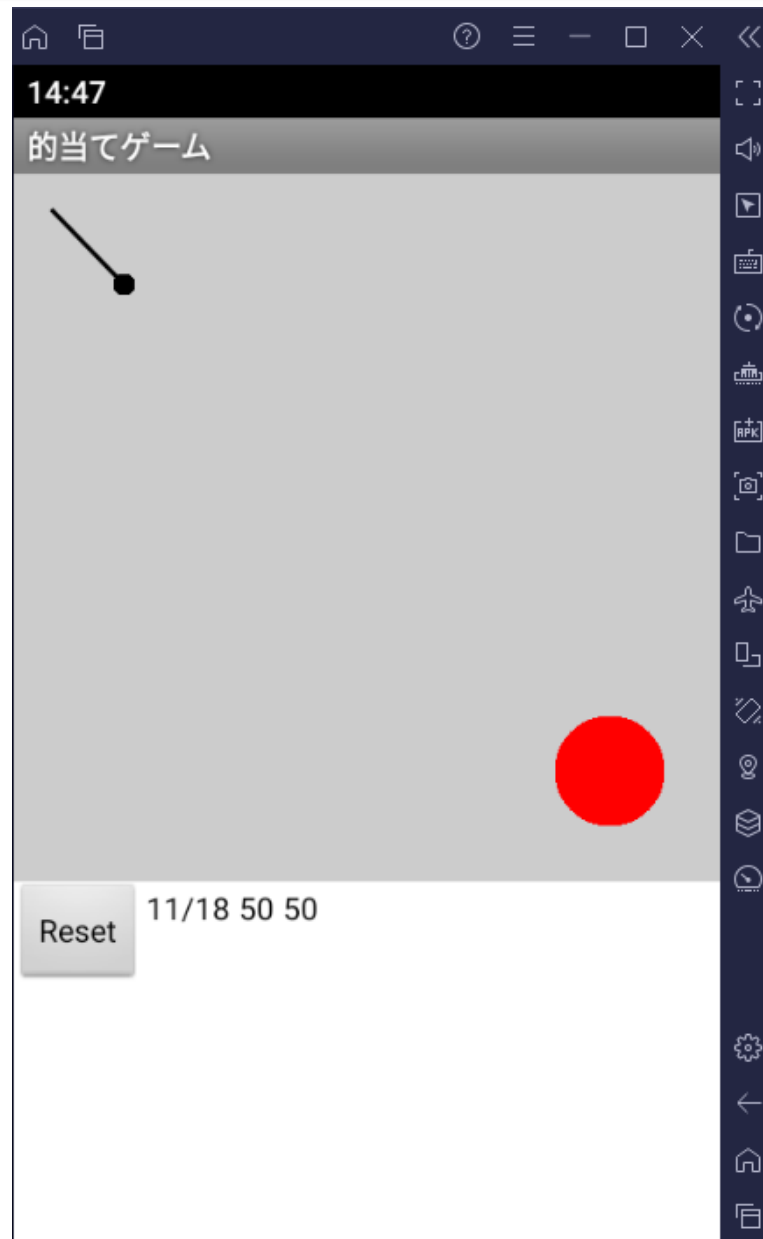
do

- 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  $\text{get global total} + 1$
- set Label1 . Text to  $\text{join}(\text{get global score}, \text{" / "}, \text{get global total})$

} 追加

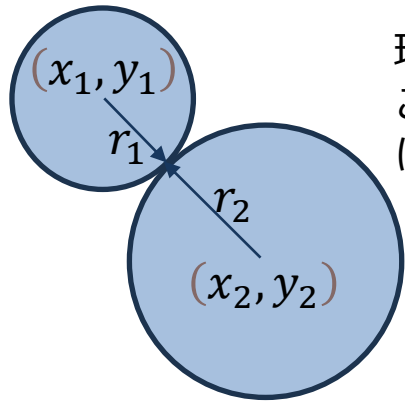
ここまですでに再び動作確認すること

# 実行画面



# 衝突したら球が跳ね返るようにする

The image shows a Scratch script for a ball's movement and collision handling. The script starts with a 'when Clock1 .Timer' event, followed by a 'do' loop. Inside the loop, the ball's X and Y coordinates are updated based on global velocity variables (vx and vy). An 'if' block checks for a collision using a 'square root' block. The 'if' block's condition is currently empty, and the 'then' block contains three 'set' blocks: 'set global vx to', 'set global vy to', and 'set global score to get global score + 1'. A red bracket on the right side of the 'if' block is labeled '追加' (Add), indicating that the collision logic is being added to the existing movement code.



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

# 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