

プログラミング概論

第15回 2024年1月17日

レポート作成

App InventorとJavaScriptの比較

今回の授業内容

- JavaScriptとAppInventorでピアノをつかって作業時間を比較
- JavaScriptとAppInventorで太鼓をつかって入力時間を比較
- JavaScriptとAppInventorでの的当てゲームをつかって入力・作業時間を計測
- 両者の違いについてレポートを書く

まだできていない人は必ず終わらせてから次へ進む

JAVASCRIPTとAPPINVENTORで 的当てゲームをつくって入力・作業時 間を計測

入力時間・作業全体にかかる時間を計測しそれぞれ比較を行う

1. ピアノアプリのJavaScriptファイルをNotePad++で開き、的当てゲームとの共通個所を除いて削除 + 赤文字部分を追加・修正
2. ストップウォッチスタート
3. 的当てゲームの枠内を入力する
4. 入力が終わったらかかった時間を記録し、動作確認を行う
5. うまく動くようになるまで修正を繰り返し、動いたら最終的な作業時間を記録する
6. 続いてAppInventerで改めて的当てゲームを作成し、作業全体にかかる時間とブロック配置にかかる時間をそれぞれ計測する

App Inventor

ブロックの配置にかかった時間 : 分 秒

プロジェクトを新規作成してから正しく動くことを
確認するまでにかかった時間 : 分 秒

JavaScript

点線内の入力時間 : 分 秒

全体の作業時間
(動くなるようになるまでの時間) : 分 秒

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" on the left, "Viewer" in the center, "Components" on the right, and "Properties" on the far right. The "Palette" panel lists various components under 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 the "Canvas" component in the palette to its icon in the "Viewer" panel, which is also highlighted with a green box. Another red arrow points from the "Canvas" component in the palette to the "Canvas1" component in the "Components" panel. The "Viewer" panel shows a mobile phone screen with the text "的当てゲーム" and a status bar at the top displaying signal strength, Wi-Fi, battery, and the time "9:48". The "Properties" panel on the right shows the settings for "Canvas1 (Canvas)", including "Appearance" properties like "BackgroundColor", "BackgroundImage", "FontSize", "Height", "Width", "LineWidth", "PaintColor", "TextAlignment", and "Visible".

MIT APP INVENTOR

Projects Connect Build Settings Help My Projects View Trash Guide Report an Issue English akiyolab5@gmail.com

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)

的当てゲーム

9:48

Canvas1

Canvas1

Properties

Canvas1 (Canvas)

Appearance

BackgroundColor

Default

BackgroundImage

None...

FontSize

14.0

Height

Automatic...

Width

Automatic...

LineWidth

2.0

PaintColor

Default

TextAlignment

center : 1

Visible

Rename Delete

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. At the bottom of the interface, there are "Rename" and "Delete" buttons for the selected component, and a footer link for "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

Canvas1

Screen1

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

9:48

Ball1

PaintColor
Default

Radius
5

Visible

X
50

Y
50

Z
1.0

▼ Behavior

Enabled

Heading
0

Interval
50

OriginAtCenter

Rename Delete

Media 50 →

Upload File →

[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 →

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. 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 profile. The main workspace is divided into three sections: a left-hand palette, a central canvas, and a right-hand properties palette.

In the left-hand palette, the 'Sensors' category is expanded, and the 'Clock' component is highlighted with a red dot. A red arrow points from this dot to the 'Clock' component on the central canvas, which is represented by a red circle. Another red arrow points from the 'Clock' component in the palette to the 'TimerInterval' property in the properties palette.

The properties palette on the right shows the 'TimerInterval' property set to the value '50'. A blue arrow points from the number '50' to the input field. The input field is highlighted with a red box. Below the input field are 'Rename' and 'Delete' buttons. The 'Media' section below shows an 'Upload File ...' button.

At the bottom of the interface, there is a 'Non-visible components' section with a small icon and a 'Privacy Policy and Terms of Use' link.

JavaScriptに合わせたプログラム

initialize global vx to 0

initialize global vy to 0

initialize global a to 0

initialize global dt to 1

```
when Screen1.Initialize
do
  set Ball1.X to 50
  set Ball1.Y to 50
  set Ball2.X to 270
  set Ball2.Y to 270
  set Clock1.TimerInterval to 50
```

initialize global dv to 0

initialize global dx to 0

initialize global dy to 0

```
when Clock1.Timer
do
  call Canvas1.Clear
  set global dx to get global vx × get global dt
  set Ball1.X to Ball1.X + get global dx
  set global dv to get global a × get global dt
  set global vy to get global vy + get global dv
  set global dy to get global vy × get global dt
  set Ball1.Y to Ball1.Y + get global dy
  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
```

```
when Canvas1.TouchUp
  x y
do
  set global vx to 0.2 × Ball1.X - get x
  set global vy to 0.2 × Ball1.Y - get y
  set global a to 0.5
```

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

**両者の違いについてレポートを
書く**

第4回課題レポートについて

- タイトル
 - AppInventorとJavaScriptの比較
- 内容
 - 開発効率などの視点から両者の違いについて調査を行い、結果について考察する
- 提出方法
 - 前回と同じ
- 提出期限
 - 1月30日（火） 23:59
- 文字数
 - 1200字以上

第4回課題レポートについて

• 注意点

- 決められた構成で書く
- 「目的」は2つの開発方法のメリット・デメリットを比較することであって、**単位の取得ではない**
- 「方法」については第3者が再現できるよう詳細に実験方法を書くこと（授業で指示されたとおりにやっただけだとしても）
- ○分○秒といった計測結果は「結果」の章に入れること（当たり前のようにだが、なぜか例年「方法」の章に書く学生が何名もいる）
- ウェブページや書籍などを参考にした場合は、必ず出典元について明示すること（講義資料を除く）
- 「表」または「図」を1つ以上入れること

第4回課題レポートについて

- レポートの構成

1. 背景および目的

2. 方法

- 2.1 開発全体にかかる時間の調査

- 2.2 入力にかかる時間の調査

3. 結果

4. 考察

5. 結論

<参考文献>

章見出しも節見出しも
変更しないこと

実験方法について
ここにそれぞれ
説明を書く

計測結果について
はここに書く

お知らせ

本日の授業が最終回ですので
授業評価アンケート
に回答してください。