

# アプリと学校教育

東海大学 大西建輔

# 自己紹介

氏名：大西 建輔(おおにし けんすけ)

所属：東海大学理学部情報数理学科  
… 教員養成も.

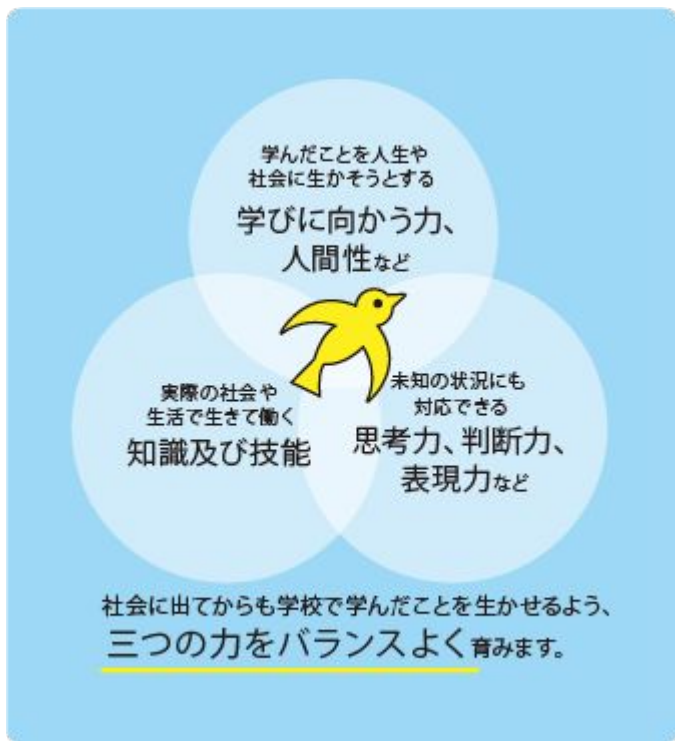
専門：データ構造, データ工学, 情報/数学教育

Androidとの出会い：Google Developer Day 2009 Japan で GDD Phoneと.

# 今日の内容

- 今の小中学校, 高等学校で起こっていること.
  - 指導要領の改訂と実施(2020-)
  - GIGAスクール構想の実現
- アプリ開発者にできること.
- 研究室で開発したアプリ

# 指導要領の改訂と実施



- 答申と改訂(2016年度)

生きる力 学びの、その先へ

- 学びに向かう力, 人間性.
- 知識及び技能.
- 思考力, 判断力, 表現力.

# 指導要領の改訂と実施

- 全面実施

- 2020年度(小学校), 2021年度(中学校)
- 2022年度(高等学校)

アクティブ  
ラーニング

主体的・対話的で深い学び… どのように学ぶか？

- 子どもたちの頭の中が**アクティブ**に働いているか？
- 見通しをもって作業したり, 粘り強く考える.
- 振り返る内容を充実させ, 次に生かせる気付き
- 周りの人達と共に考え, 学び, 新しい発見や豊かな発想を生む.
- 知識がつながり, ”わかった”, ”おもしろい”と思える

# 小学校の一例（焼津市立豊田小学校） 2021/6



# 新指導要領での情報に関する授業内容

小学校(2020) … “プログラミング的思考”の育成

[小学校を中心としたプログラミング教育ポータル](#)

- 正多角形の描画(5年生, 算数)  
… Scratch を用いたプログラミング.
- 電気の利用(6年生, 理科)  
… micro:bit などを用いた制御.
- 総合的な学習の時間, 情報に関する探究的な学習  
… 低学年でも Viscuit などを利用.

# 中学校の一例(奈良市立春日中学校) 2020/11





# 新指導要領での情報に関する授業内容

中学校(2021) … 技術・家庭の技術分野にあり。

- 情報の技術（他に3項目）
  - 生活や社会を支える情報の技術
  - ネットワークを利用した **双方向性のあるコンテンツ**  
**のプログラミング**による問題の解決
  - **計測・制御のプログラミング**による問題の解決
  - 社会の発展と情報の技術

時間として10  
時間程度

中学校技術・家庭科(技術分野)内容「D 情報の技術」

# 新指導要領での情報に関する授業内容

高等学校(2022) … 情報I(必修), 情報II(選択)

- (1) 情報社会の問題解決 研修用教材
- (2) コミュニケーションと情報デザイン 研修用教材
- (3) コンピュータとプログラミング 研修用教材
- (4) 情報通信ネットワークとデータの活用 研修用教材

プログラミング  
も必修

# 新指導要領での情報に関する授業内容

## 高等学校(2022) … 情報II(選択) 研修用教材

2023年から

- (1) 情報社会の進展と情報技術
  - (2) コミュニケーションとコンテンツ
  - (3) 情報とデータサイエンス
  - (4) 情報と情報技術を活用した問題発見・解決の探求
- 31% ((=開講予定 34+検討中 26)/191)で  
開講予定

# 大学入試での情報科目

新カリキュラム(2022年)として実施 = 2025年入試

令和7年度試験の問題作成の方向性, 試作問題 (2022/11公開)

- 数学I, A, II, B, III → 数学I, A, II, B, III, **C**
- 情報I (新設, 必履修)
- 歴史総合(新設, 必履修) 日本史/世界史の近現代史

入試に情報を出す大学も

- 共通テストの情報を利用 (選択科目)
- 独自の情報入試も (日大文理, 大妻女子, 南山, 京産)

# GIGAスクール構想

- 2019年(令和元年) 2318億円の予算措置
- 2020年 2292億円の予算措置
  - 生徒一人一台端末の整備 (小中)
  - 構内通信ネットワークの整備 (小中高)

主体的・対話的で深い学びが目標

2019年度末(2020年初め) Covid-19の拡大.

⇒ 急遽端末の導入が進む(2024年までに順次導入)

# GIGAスクール構想の問題点

自治体間の格差 … 自治体により活用状況に差.

ICTが分かる人が学校/教育委員会に居るか？

- 学校内：一部のICTが使える教員に負担  
端末管理, 教員向けの講習, … **何でも屋**
- 教育委員会：推進か禁止か？  
○○が使える？/使えない？ … メール, アプリ…  
端末の持ち帰り/学校のみ利用.

# 問題点の解決に向けて(1)

自治体間の格差 … 自治体により活用状況に差.

- ICTが分かる人が学校/教育委員会に居るか？
  - 学校：一部のICTが使える教員に負担
    - ⇒ GIGAスクール運営支援センター整備
    - ⇒ ネットワークアセスメント実施促進
  - 教育委員会: 推進か禁止か？
    - ⇒ 文部科学省は**推進** StuDX Style の開設

# 問題点の解決に向けて(2)

民間業者

## GIGAスクール運営支援センター整備

- 運営支援センターへの委託を都道府県単位で。
  - ヘルプデスクの開設(家庭, 学校からの対応)
  - ネットワークトラブル対応
  - 各**地域での**支援人材の育成
    - 教師, 職員, **支援人材**向けICT教育.



# 問題点の解決に向けて(3)

## ネットワークアセスメント実施促進

- 現状のネットワークを分析・診断し, 現状把握と改善などの提言をおこなう
  - 通信量やセッション数の測定
  - ネットワークの構成や機器の設定の調査
  - 通信速度, 通信遅延の調査
  - 無線の電波干渉, カバーエリアの調査

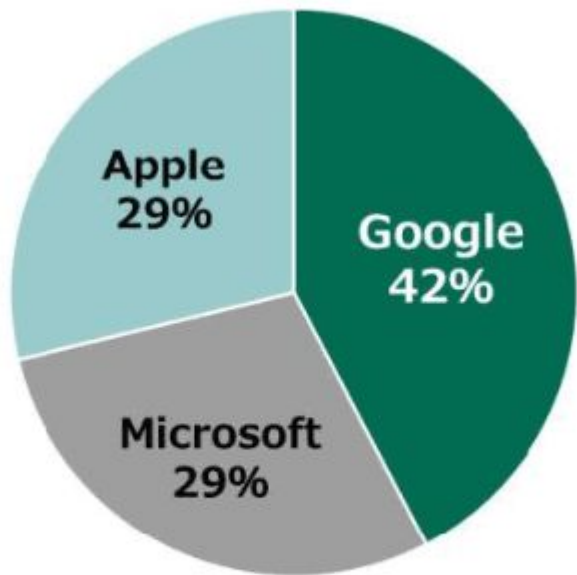
# 問題点の解決に向け(4)

## StuDX Style (スタディーエックス スタイル)

… 文科省の開設したGIGA端末利用のお手本

- GIGAに慣れる--導入にあたって, 使ってみよう  
最初の指導を少人数で
- 教師と子供がつながる 取組状況を個別に把握
- 子供同士がつながる 振り返り活動のDX
- 家庭と学校がつながる 欠席連絡をデジタル化
- 職員同士でつながる 会議のペーパーレス化

# GIGAスクールでの端末



- 約902万台のうち、ChromeOSが42%  
= Chromebookが380万台
- メーカー別順位
  1. Apple(iOS)
  2. Lenovo(Chrome,Win)
  3. NEC(Chrome)

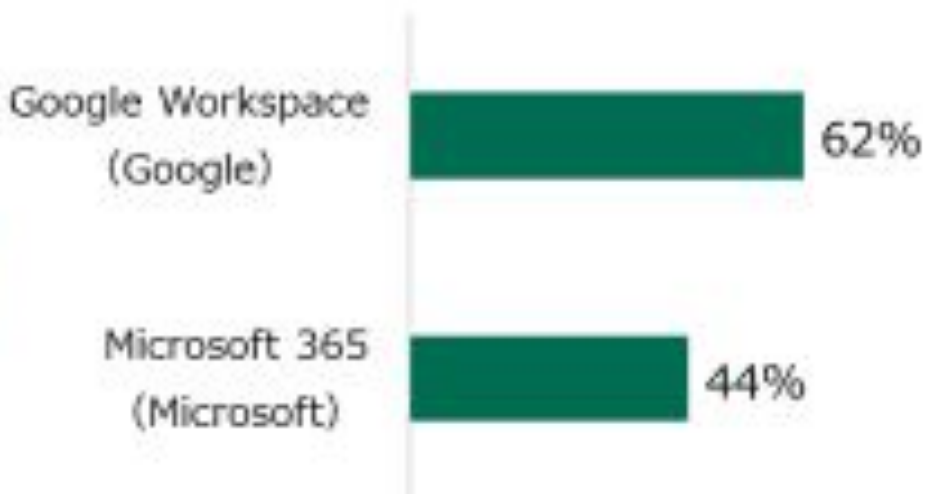
※端末台数について回答した1,696団体を対象に集計

© 2023 MM Research Institute, Ltd.

2021-2023 MM総研による調査  
対象：1741教育委員会

# GIGAスクールでの汎用クラウドツール

汎用クラウドツールのシェア (n=1,571) ※複数回答



© 2023 MM Research Institute, Ltd.

1571自治体のうち、**971自治体(62%)**で Google Workspace が利用可能

学習eポータル, 校務支援は別途あり.

2021-2023 MM総研による調査  
対象：1741教育委員会

# アプリ開発者にできること

GIGA端末で動作する教材アプリ作りませんか？

- Androidアプリは, ChromeOSで動作可能
  - ただし, UI部分など手をいれる必要あり.
  - Android アプリを Chrome OS 向けに最適化する
- Webアプリで作りましょう.
  - 使う機能にもよりますが, ChromeOS, iPadOS, Windowsでも動作可能.

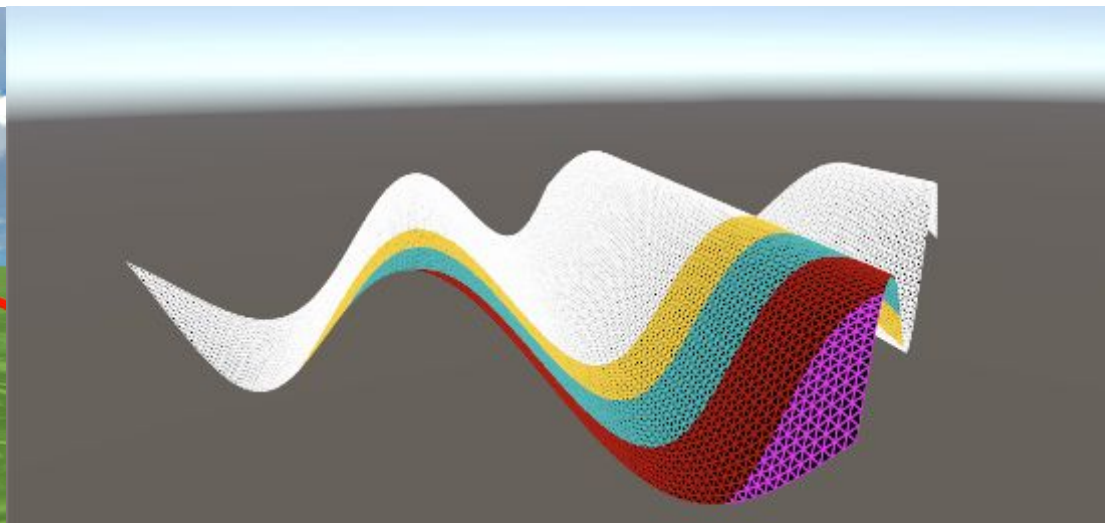
# 研究室で開発したアプリ

- [Waquema](#) : ソースコードから演習問題生成  
(出力先は, Google Form)
- [つちのこ 2.0](#) : 2025年からの共通テストで使われる(予定の)DNCL2の処理系, Web上で動作.
- アプリで学ぶデータ構造 : Monacaベースでアシアル(株)が開発. 基本アイデアは大西.

# 研究室で開発したアプリ

配信停止

- 斜方投射VR (ICT-KE2020)
- 音声で曲面描画VRアプリ (ICT-KE 2022)
- VR/ARでのオイラーの多面体定理の理解 (開発中)



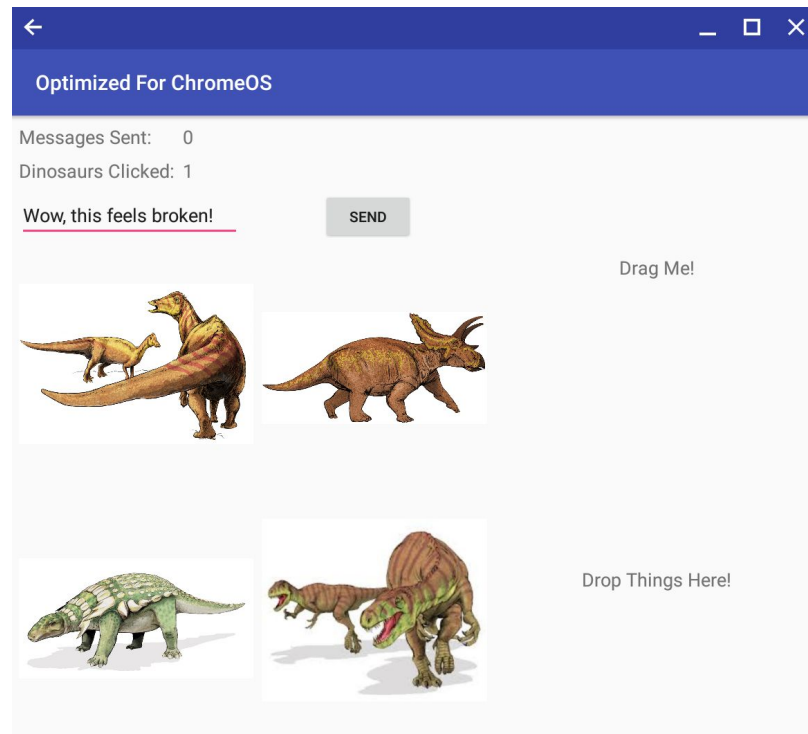
# まとめ

- 学校での教育にコンピュータが導入
  - 42%が Chromebook
  - 円滑な運営には, ITに詳しい人が必須.
- GIGA端末向け教材アプリが欲しい.
  - Androidアプリの転用は? .. それなりに大変
  - Webアプリのほうが対応機器が多い.
- (個人)アプリ公開の継続コスト増



# AndroidアプリをChromeOS向けに

- キーボード入力の操作
- マウス処理
- アーキテクチャコンポーネントの使用



CodeLab [Android アプリを Chrome OS 向けに最適化する](#)

# エンターキー

- MainActivity.kt の onCreate内

```
// Enter key listener
edit_message.setOnKeyListener(
    View.OnKeyListener { v, keyCode, keyEvent ->
        if (keyEvent.action == KeyEvent.ACTION_UP
            && keyCode == KeyEvent.KEYCODE_ENTER) {
            button_send.performClick()
            return@OnKeyListener true
        }
        false
    })
```

矢印キーでの移動 ⇒ ビューのフォーカスで実現

- MainActivity.kt の onCreate内

```
image_dino_1.setFocusable(true)
```

```
...
```

```
image_dino_4.setFocusable(true)
```

```
val highlightValue = TypedValue()
```

```
theme.resolveAttribute(R.attr.selectableItemBackground, highlightValue, true)
```

```
image_dino_1.setBackgroundResource(highlightValue.resourceId)
```

```
...
```

```
[res/values/styles.xml]
```

```
<item name="colorControlHighlight">@color/colorAccent</item>
```

# Ctrl+ ベースのショートカット(1)

(Undo: Ctrl+z, Redo: Ctrl+Shift+z)

```
[MainActivity.kt] // Undo 部分のみ.  
private var undoStack = ArrayDeque<Int>()  
private val UNDO_MESSAGE_SENT = 1  
  
//In ImageOnClickListener  
undoStack.push(UNDO_DINO_CLICKED)  
redoStack.clear()
```

# Ctrl+ ベースのショートカット(2)

(Undo: Ctrl+z, Redo: Ctrl+Shift+z)

```
override fun dispatchKeyShortcutEvent(event: KeyEvent): Boolean {
    // Ctrl-z == Undo
    if (event.keyCode == KeyEvent.KEYCODE_Z &&
        event.hasModifiers(KeyEvent.META_CTRL_ON)) {
        val lastAction = undoStack.poll()
        if (null != lastAction) {
            redoStack.push(lastAction)
            when (lastAction) {
                UNDO_MESSAGE_SENT -> {
                    messagesSent--
                    text_messages_sent.text = (Integer.toString(messagesSent))
                }
            }
        }
    }
}
```

# 右クリック (1)

[MainAcitivity.kt]

```
registerForContextMenu(image_dino_1) // コンテキストメニューがあること
```

...

```
override fun onCreateContextMenu(menu: ContextMenu, v: View, menuInfo:
```

```
ContextMenu.ContextMenuInfo?) {
```

```
    super.onCreateContextMenu(menu, v, menuInfo)
```

```
    val inflater = menuInflater
```

```
    inflater.inflate(R.menu.context_menu, menu)
```

```
}
```

[res/menu/context\_menu.xml]

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">
```

```
    <item android:id="@+id/menu_item_share_dino"
```

```
        android:icon="@android:drawable/ic_menu_share"
```

## 右クリック (2)

アイテムが選択されたときのアクション.

```
[MainActivity.kt]
```

```
override fun onContextItemSelected(item: MenuItem): Boolean {  
    if (R.id.menu_item_share_dino == item.itemId) {  
        Snackbar.make(findViewById(android.R.id.content),  
            getString(R.string.menu_shared_message),  
            Snackbar.LENGTH_SHORT).show()  
        return true  
    } else {  
        return super.onContextItemSelected(item)  
    }  
}
```

# ツールチップ：カーソルを合わせたときに表示される文字列

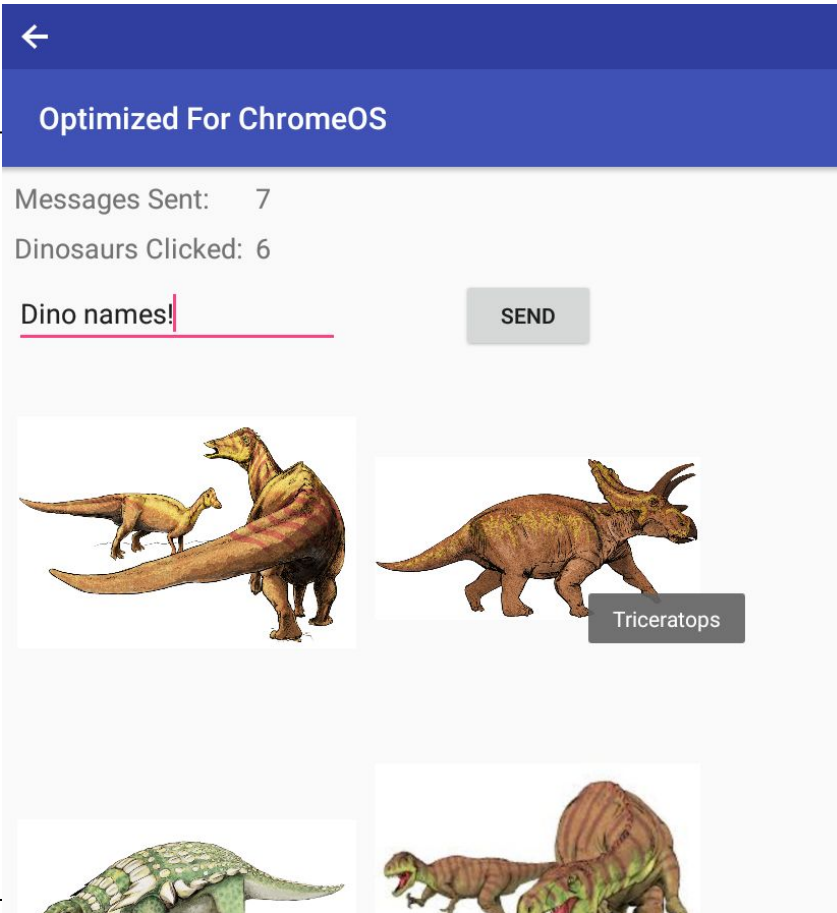
```
[MainActivity.kt]
```

```
// Add dino tooltips
```

```
TooltipCompat.setTooltipText(image_dino_1,  
getString(R.string.name_dino_hadrosaur))
```

```
TooltipCompat.setTooltipText(image_dino_2,  
getString(R.string.name_dino_triceratops))
```

```
...
```





# マウスオーバー効果：

## カーソルを合わせるとボタンの色を緑色に

```
[MainActivity.kt]
```

```
button_send.setOnHoverListener(View.OnHoverListener { v, event ->
    val action = event.actionMasked
    when (action) {
        ACTION_HOVER_ENTER -> {
            val buttonColorStateList = ColorStateList(
                arrayOf(intArrayOf()), intArrayOf(Color.rgb(127, 0, 255, 0))
            )
            button_send.setBackgroundTintList(buttonColorStateList)
            return@OnHoverListener true
        }
    }
})
```