

＋ 今日の内容

山田編の導入

＋ ガイダンス

授業意図

建築情報学の導入として位置付く授業である。

昨今「建築情報学」への期待は極めて高い。現代は、誰もが高度な情報技術を利用できる環境が得られる時代だ。情報技術に限らず、立場や年齢を問わず全ては発想と技能の組み合わせ次第、というチャンスの時代である。個人であってもやりたいことを実行する阻む壁は極めて低い。少し大袈裟に言えば何にでもなれる時代、と言ってもいいだろう(=「できなかった」という言葉が通じなくなった=「できなかったことはやらなかったこと」)。やりたいと思うことをできることの多様化とそれが産み出す可能性の増大は、「楽しさとやりがい」の増大でもある。これを原動力とした先駆的な取り組みも多く、日々の生活に限らず建築・都市分野でも高い注目と投資を集めている。先行事例は無数にある。自分で探し欲しい。それらを見れば建築・都市でも情報技術が高い注目を集めていること、ワクワク感を感じられるはずだ。

上記に関連する先行事例が多数存在することは、「建築情報学の理解・習得は必須(教養)」となりつつあることも意味している(他分野で情報技術の習得はとっくに必須である)。しかし「もはや当たり前だから学ぶことも簡単」と片付けられないのも事実である。その筆頭が学習開始時の戸惑いと挫折だ。可能性の広大さ・高い分野横断に比して建築情報学は内容が多岐に渡るため、学習の入り口に戸惑うことも多い。これは本授業の内容に限らず、建築情報学を用いたデザイン・提案・研究を習得するには、対象そのものに対する能力だけでなく、「コンピューターシンキング」と「操作」と「知識」を学ぶ必要があるため、何が足りないのかすら自分で分からない状態に陥ることが多いからだ。そこで、

・情報処理にて「**コンピューテーショナルシンキング+プログラミングを用いたデザイン(=Argosmic Design)**」,

・情報処理演習では「**3次元モデリング+環境シミュレーションを用いたデザイン(Engineered Design)**」,

を建築情報学の入り口としてを学ぼう。

この授業だけで何かスゴイこと成し得るまでは至らないかもしれないが、今後の自習がスムーズに進むよう、基礎的な操作を演習形式で学習しよう。なお情報分野は良くも悪くも「先行者利益」という言葉が通じる分野である。また自身の主体性による学習が8割以上を占める分野でもある。建築学科の1回生ですが、今から情報分野も頑張っていきましょう！

※ 欄外について

欄外には、本文中に記述しづらい事で大切な事が記述されています。授業中に話す場合と話さない場合があります。不要な事・些細なことではありません。目を通すように。差がつく項目とも言えます。

※ 建築情報学の学びのいい所

やれば必ずできるようになる。やったらやっただけ必ず伸びる。いわゆる神ゲーである。

何ができるようになるのか(この授業の延長に)

* コンピューショナルシンキング:

「知識は技術を基に課題が何であるかを理解し、課題を適切に解決する方法を導出する思考及び能力」のこと。論理的思考とほぼ同義だが、対象をコンピューターが理解・解決できるアルゴリズム(計算可能な手続き・論理回路)に翻訳するためには、コンピューターの処理に関する知識と特性をふまえた上で、人間の論理的思考よりさらに現象を客観的な手続きの連鎖として記述可能にする必要がある。そのため日々の生活や国語科目としての論理的思考に長けているからと言ってコンピューショナルデザインができる訳ではない。実はかなり難しく実際のプログラミング体験に基づく勉強が必要であり。コードを覚えてきたらここが最重要にある。この能力を高めることで情報技術を用いた問題解決能力だけでなく、物事の洞察の確固たる一つの方法論を獲得となり、思考が鋭くなる。授業ではこの基本と重要性を体験的に理解する。

* プログラミング:

上述のコンピューショナルシンキングに基づき実際に実行できるようにコーディングすること。この部分の経験と知識がコンピューショナルシンキングを鋭くする。授業では基本的な文法と関するを体験的に理解する。

* 3次元モデリングによりデザインスタディ:

スキルが伴えばどんな形態でも「外部化」「視覚化」することができる。デザインにおいて頭の中にあるアイデアや形態を「外部化」「視覚化」することはとても重要である。自身での推敲、他者との相談において「外部化」が必須。またそのデザインを外部化して分析して改良する(スタディ)回数を増やすことができれば、出来栄を比例的に向上させることができる(直線ではなく指数関数的な時もあるが)。つまりこのため速度・質に関するスキルはデザイン力そのものである。授業では3次元モデリング用いたスタディの基本と有用性(欠点)を体験的に理解する。

* デジタル技術を用いたプレゼンテーション

他者(クライアント、仕事仲間など)に対してアイデアを示すためのプレゼンテーション能力を向上させることができる。図の種類で分けると「一般的なパース(内外観)、アクトメ図、断面パースなど」、用途で分けると「完成予想図のような写実的な図、コンセプトやシステムを示す立体的なダイアグラム(モデル図)」など、場合によってはアニメーションも才能に一切関係なくやれば必ず表現できるようなる。授業では3次元モデリング用いた基本と有用性(欠点)を体験的に理解する。

* 環境シミュレーション

光の当たり方、風の通り方を仮想的に視覚化して検討することができる。本授業の環境シミュレーションは簡易なので「正しい結果とまでは言えない」ことには注意が必要である。それでは環境という要因は視覚化してデザインする体験には意義がある。ただ専門性が高い。そこで授業で「アプリケーションの使い方を学び使ってシミュレーションを援用した計画プロセスのイメージ醸成する」ことを目的である。そのため「環境シミュレーションの実行及び結果の解釈に必要な環境工学に関わる専門的知識」は後期以降の授業で学ぼう。後期の授業では蓄積された環境計画の先人の知恵に関する授業・演習もある。山田の授業ではない環境工学・環境計画の専門家の先生の授業なので楽しみに!

» 技術と思考

こちら鶏と卵と同じ関係ですね。両方がお互いに刺激し合うことがやはり大切です。この授業ではまず技術からスタート。

» プレゼンテーションの重要性

細かい説明は省略しますが、とても重要です。残念ながら評価の半分はプレゼンと思った取り組みの方がよい。

» VR・AR

3次元モデルが作れば意外と簡単にできる。時間が余れば紹介します。

✦ 用語

専門性が高い授業のため専門用語が多い授業になる。体感的な理解が重要だが、言葉としても下記を覚えておこう。

Computational Design(コンピューショナルデザイン)

人とコンピュータの共創によるデザインのこと。先端的な情報技術(AIやIoT, Parametric design(パラメトリックデザイン)やシミュレーション)を使ったデザインに対して呼称されることが多いが、「Computer Aided Design」「Computer Graphics」「Building Information Modeling」も含まれる。本授業では期待感の高まりが著しいコンピューショナルデザインの基礎として「Computer Graphics」技術の技能と体験を培いましょう。

Computer Aided Design(CAD)

1960年台にサザランド博士(1938~)が開発。広義ではComputational Designと同義だが、狭義で2次元CADのことを指すことが多い。

- * マウスとキーボードを用いた製図
- * 複雑で手間のかかる製図作業をデジタル化
- * 図面の引用を可能にした。
- * 建築設計の高度化・高品質化に大きく貢献
- * 「製品化に伴う画一化」と「建築デザインの自由」の両方を生じさせた。
- * 2回生の後期の「CAD/CG演習」学習する。
- * AutoCADを学習する予定(学生利用は無料)

Computer Graphics

いわゆる「CG」のこと。広義ではコンピュータを使ったグラフィックを総称するが、3次元モデリングを使ったレンダリング画像・映像を指すことが多い。

- * マウスとキーボード+αを用いた3次元造形物(モデル・モデリング)
- * アプリケーション上で表現される光(ライト・ライティング)
- * アプリケーション上で表現される質感(マテリアルとテクスチャ)
- * レンダリング>光と質感が適用された絵のこと
- * 圧倒的な自由な表現力
- * CGは表現が画一的は誤り、発想とスキル次第
- * 2回生の後期の「CAD/CG演習」で再度学習するが、先取りがお勧め
- * 3dsMAXを学習する予定(学生利用は無料)

「何故CGから?」

CADとBIMには建築の専門知識や基礎製図の技能が必要なためです。CGでもそうですが、前者に比べれば無くて学びやすい。

Building Information Modeling

「BIM(ビム)」と呼称される。狭義では属性情報付きの3次元モデルを指し、広義ではそれらを用いたシミュレーションやデザイン方法そのものを指す。ここ5年位で実社会で急速に普及が進む建築情報技術。

- * CGに建築的な属性情報が加わったモデルのこと(BIMモデル)
- * 3次元と2次元が紐づいている(図面が自動生成される、逆もまた然り)
- * BIMを様々な立場の人が共有し建築行為を進める思想(プラットフォームとしてBIM)
- * 設計・設備・構造・施工・管理、全分野で導入が進んでいる
- * レンダリング>光と質感が適用された絵のこと
- * モデリングがやや制約的(特に初期)
- * 建築的な知識も必要
- * 3回生の前期の「BIM総合演習」で学習するが、先取りがお勧め
- * Revitを学習する予定(学生利用は無料)

✦ 学習するアプリケーション (/ 言語) と類似

CG

* 3ds MAX (Autodesk) : 通称マックス (情報処理演習で使用する)

ゲーム業界を主に映画でも用いられる映像業界の老舗かつメジャーアプリケーションの一つ。建築分野でも多く用いられている。アプリケーション内でレンダリングまでが可能、かつ拡張性も高い。技術があれば出来ないことはほぼ無い (例:「アバター」「エヴァンゲリオン新劇場版:破」「イノセンス」等)。反面、やや慣れづらい。またやや重い。WEB上の情報・参考書多数。

その他の有名なソフトとしては、

- * **Maya** : 3dsMAX と同様に業務用の AutoDesk 社の CG ソフト、映画はこちらが多い?
- * **Rhinoceros** : 工業製品分野を狙ったモデリングソフト、NURBS (Non-Uniform Rational B-Splines) モデリングが得意、プログラミング思考、GRASSHOPPER との連携で急速普及、海外の現代的な著名建築物で利用が多い?
- * **SketchUP** : 習得が容易で、スケッチのような操作感と表現効果を持ち、スタディや模式表現に非常に強い。かつ無料、安い。レンダリングは別
- * **Shade** : プロダクトよりのソフトだが自由な造形や細かい表現に強く、建築で使う人はアトリエ系が多い?
- * **Form-z** : 建築分野では人気が高かったソフト、落ち着いた風合いの CG が作りやすい
- * **AutoCAD, VectorWorks** : 2D-CAD の延長で制作可能ではある。
使い勝手に差がある。どれか1つを身につければ他のソフトウェアでも応用がかなりきく。

CAD

* AutoCAD (Autodesk) : (CAD/CG 演習で使用予定)

デファクトスタンダード (業界標準) と呼ばれる製品。

- * **JW-CAD** (実質無料であることと、スピーディーな操作感のため根強い人気)
- * **VectorWorks** (シェアは大きくないが、その使用感や表現力ゆえに教育界や小さな事務所でも人気が高い)
それぞれ使い勝手は異なるが2次元CADは差が小さい。

BIM

* Revit (Autodesk) : (BIM 総合演習で使用予定)

国際的にはデファクトスタンダード (事実上の業界標準) と読んで差し支えない製品。特に大手で導入が進む。国内でも最大シェアになりそうな気配。

- * **ArchiCAD** : 何故か日本では Revit 以前に普及が進んだのはこちら。制約条件が緩めで意匠 BIM (3次元モデルからの図面ジェネレータ) としては使いやすい。
- * **ARCHITREND ZERO** : 福井コンピューティングという会社が開発する国産 BIM, シェアは高くないがモデル連携や細かな部分で評判が良い?
やはりそれぞれ使い勝手は異なるがどれか一つを習得すれば乗り換えに苦労は少ない。

» まだまだあります

- Cinema4D
- Lumion
- Dynamo
- Fusion360
- Softimage ICE

» アプリ名で検索しよう

参考書だけでなく WEB 上に大量に情報がある。比較的デファクトスタンダードに近いアプリケーションを取り扱う理由の一つは自習の容易さです。動画やサイトを見て、技術的な勉強だけでなく、モチベーションも上げましょう。

プログラミング言語

* python : 近年のデータサイエンスに対する注目の高まりと同期するように急速に世界的に普及が進んでいるプログラミング言語。特徴には、オープンソース(無料)、シンプルな構文、可読性の高さ、豊富なライブラリ(コミュニティの広大さ)、が挙げられ、参考書はWEBも多い。機械学習のフレームワークである Google の TensorFlow や Facebook の PyTorch を筆頭に、主要なフレームワークでは多くがサポートされている。普及の広がりから 20 年度から日本の国家資格である基本情報技術者試験で出題されるプログラミング言語に Python が含まれるようになった。学習の容易さの反面には、C 言語などと比べて実行処理が遅かったり、メモリ消費量が多いという点が挙げられる(スクリプト(インタプリタ)言語とコンパイラ言語)。それでも実務でも非常に多く用いられている(instagram, Instagram, Evernote, YouTube, Pinterest, Dropbox, 少なくとも一部に採用していると言われている)。ただ当然 pyhton が全てであり万能で最高!、と言う訳ではない。他のプログラミング言語についても簡単に調べてみよう。

学習するアプリケーション(言語)の選択と連携

学習するアプリケーション(言語)の選択は悩み所です。勉強を兼ねてそれぞれの特徴を調べるのは良いことです。ただ一朝一夕にはいかず、将来は分からないのが事実です。そのため「学習先を延々と迷うよりも取りあえず初めてしまう」ことが大切です。あるアプリケーション(言語)に長けている・学んだという経験は、類似の学習の非常に非常にとてつもなく強力に有効です。また一つのアプリケーションに長けていて、使おうと思えば類似アプリケーションも何となく使える能力と行動特性(習慣)がある、という能力が重要。事実、大手建設会社に対するアンケート結果で若手に期待することの上位に「アプリケーション・言語を連携して情報技術を使いこなす素養」が挙げられています。

» メモをとる
一度学んでも忘れるのは当然、
2 回目に学ぶ(思い出す)時
の速度を上げることが重要。
メモをとる習慣は早めにつけ
ましょう。時にはフリも重要。

✦ 色々

取り組むペースについて

情報処理演習では「Weekly 課題」を設定している。少し面倒に感じるかもしれないが、少なくともこのリズムよりは遅れないようにリズム良く取り組もう。情報処理は随時提出する課題は設定していない。ただ動画を**見て簡単かつ少しでもいいので自分で考えた何かを作ろう**。このサイクルの繰り返しが上達の秘訣だ。

評価方法

課題の評価は、工夫と作り込みの度合いも重視する。デジタルツールを使いこなしたコンピュータショナルデザインを習得する科目であるため、提案の中身に対する評価と同等に工夫や作り込みも重要である。「センス」「要領の良さ」ではなく、**努力を十分に評価したい、ということ**。ただ「内容・デザインの洗練さ」と「工夫・作り込み」は比例するよう。

授業の約束事

- * 課題は課題要件を満たし必ず全課題を提出することが採点の開始条件。積み上げ型です。
- * 提出締め切り厳守。遅れ提出は、特別かつ認められる理由がある場合以外は受け付けない。
- * 課題点が低ければ全てを提出しても採点が低いことは十分にあり得る。
- * コンピュータには**トラブルも多いが、全て「自己責任」**なので注意すること。
- * 教え合いは禁止しない。相談したり教え合いながら取り組むことをむしろ推奨。誰かがすごい!と思えるデザイン、早い!と思える技術を見せていたら教えて貰おう。授業について友達と沢山話そう。
- * 教室に着いたらすぐにログイン、アプリ立ち上げを行うこと。
- * 欠席・遅刻は厳禁。積み上げ型の内容についてこれなくなる等、皆さんの不利益が大きい。事務的には5回を超える欠席にてF評価となる。課題評価においても欠席・遅刻した回に出題中の課題点を減点する。遅刻2回で欠席に相当する。
- * PCルームなので**授業中は飲食禁止**。
- * 学習ソースは授業での説明を前提としている。学習ソースだけでは学習が困難な時もある。授業を聞いて補完すること。なお印刷物は配布しない。
- * **ノートパソコンの持ち込み歓迎・推奨**

» スケジューリング能力

成果物の完成度はスケジューリング能力に依存します。仕事の能力=スケジューリング能力と言ってもいいでしょう。

» 必要時間の見込み

作業時間の見込みもトラブル見込みで考えることが重要です。自分で思う時間の1.5倍以上は最低必要です。

» PCの裏切り

PCは大事なときほど裏切ります。自分の作業プロセスの記録は兼ねて、_0407_a 感じでどんどん別名保存を繰り返しましょう。

» 「センス」?

センスの良さは確かにある。但し、たまたまかもしれない、人によっては感じ方が変わるかもしれない、圧倒的なものではないかもしれない。この授業では、講習以外の内容を自主的に使っていたり、作り込んでいる提出物も高く評価したい。

✦ 自習の方が好きになろう

- * 情報技術の学習には以下の特徴がある。
- * 天井が無い
- * 可能性がとてつもなく大きい
- * 若くても最先端になれる，先生より専門家になれる
- * 意思を持って何か新しいことを学ぶということは根本的に楽しい

これは情報技術を学ぶ上で「教わる」ことに限界があることの裏返しでもある。情報技術は「自ら学ぶ」ことが大原則です。範囲が広く日進月歩である。ただ前提となる基礎理解は存在する。この授業で自習が出来るようになろう。建築情報分野は特に授業時間外を使った学習が不可欠であり，日々進歩し広範にわたる建築情報を使いこなすには「楽しみ自ら学ぶ」姿勢が必須。

アプリケーションの無料利用

- * Autodesk 関連は実務利用は非常に高価だがほぼ全てに無料学生版がある
- * 必ず Autodesk 学生アカウントを作成すること，学内施設で学習する場合にもアカウントが必要
- * Anaconda(python) はもともと無料。

参考文献等

書籍・WEB，共に多数の情報がある。「体系的に学ぶ書籍，具体的で特定の内容を知るにはWEB」という使い分けを推奨する。書籍は「説明が丁寧なチュートリアル形式」「網羅的な機能」を併せて用意するとよい。刺激も重要，作品集や展示会，WEB 記事，日常的に評価の高いデザイン・技術にふれる環境に身を置こう。

自習できる教室

- * 学内の自習利用できる情報処理教室にはインストールされている（Revit はない）
- * WW1 階 PC ルームの方が高性能。但し規則遵守！（別途周知する）
- * デザインルームの PC も譲り合って使いましょう。

楽しんで取り組んだもん勝ち，かも

「楽しむ能力」が重要視されていますね。本当に楽しくなるかもしれませんし，効率も良いことが研究的に示唆されています。「先読みして深く考え過ぎずに」とりあえずしばらく楽しみながら取り組みましょう。優秀な人は不思議と何事に対しても楽しそうです。

» 自分で学べる能力

非常に重要な能力です。今後の学習をふまえ，自分で方法を調べて制作を進められるような習慣を身に付けることが大切です。

» 自習の注意

情報処理教室はの AutoDesk のバージョンは 2017，PC ルームは 2020 です。同じバージョンが良ければ学生無料版をダウンロード時に指定を。

» 授業スタッフの所在

山田は基本的に学内に居ます。TA さんは授業中に紹介します。

» 手描き・模型の学習は必要？

細かい説明は省略しますが，必要です（今の所）。