

# コンピュータ基礎演習

## 第11回

理工学部 情報科学科 隅田 康明

[sumida@ip.kyusan-u.ac.jp](mailto:sumida@ip.kyusan-u.ac.jp)

# 対面/遠隔での質問受付体制

- 7/13から教室で授業時間に対面でも質問受付開始予定
  - 本来の時限の受講者のみ入室可
    - 入室前に手洗い、消毒、座席表への記入
    - 私語厳禁、座席は完全固定
    - 指示に従わない場合には減点や欠席措置
  - 質問は遠隔でも受け付ける
    - Zoomの画面を教室のプロジェクターに投影
    - Zoomでの個別対応は難しくなるかも知れません

詳細はK'sLifeで連絡の後に、Moodleに掲載

# 今後の予定

- **第11回～第13回：制作課題の作成**

- 3回分かけてプログラムとレポートを作成
- 第10回までのプログラムをアレンジして提出
- 独自性の高いプログラムを作りたい場合は、設計書を提出すること

- 第11回講義開始時点で、  
第10回のレポートを提出していない場合
  - 第10回のレポートは最低限のみで提出し、  
制作課題のプログラム開発に取り組むこと
    - 締め切りまでに100分以上は
  - 制作課題を後回しにすると単位が危なくなる

# レポートの締め切り (減点なしで受け付ける期間)

- 第11回：7/17（金） 18：00まで
- 第12回：7/22（水） 18：00まで
  - 締め切り以後の遅れ提出は認めない
- 制作課題：最終回授業日の18:00まで
  - **7/23（木）、月曜授業日の18：00まで**
- 追加課題：7月24日 18:00まで
- 遅れ提出の締め切り：**7/24 18:00**まで
  - Moodleでの提出も打ち切り
  - 以降は特別に認めた場合を除いて受け付けない
  - 最終的なレポート締め切りと考えて問題ない
  - **最終講義日の翌日が最終締め切り、注意すること**

# 制作課題について

- 第11回～13回の3回でプログラムとレポートを作成
- 第10回までのプログラムのアレンジ
  - 追加課題のアレンジでも可
    - ただし、追加課題をアレンジする場合は、それぞれが別の作品とすること
  - 提出済みのレポートのアレンジでも可、  
ただし、元のプログラムからの変更点が少ない場合は、元のレポートを0点とした上で評価する
- 独自性の高いプログラム
  - 事前の設計書を必ず提出すること
    - 事前に設計書を提出すれば、事前にアドバイスを返す
  - 作れないものを作ろうとすると、単位を落とす危険がある
  - 最終的には自己責任だが、出来ないものを作ろうとしている場合は止めるので、その時は指示に従うこと

# 何を作るか？

- 基本的には、これまでのアレンジで問題ない
  - ただし、最低でも変数、条件分岐、繰り返しを使った工夫をすること（全てでなくても良い）
  - 最低3回分の作業時間をかけたレポートを提出すること
    - 当たり前だが、少しアレンジして終わり、などだと点数はほとんどでない
- 既に出したレポートの続きでも問題ない
  - 第8回のゲームをさらに改良する、など
  - ただし、同じものを出してきた場合には、元のレポートは0点とする(点数も当然低くなる)

# 自分で作ったプログラムの再利用

- 自分で書いたプログラムは積極的に利用してよい
  - ゲームの障害物に利用
  - 背景として利用
  - 動く図形を増やすときに利用など
  - 過去のレポートで作ったプログラムも工夫点に入れてよい
- ただし、テキストにコピーして貼り付けても上手くは動かない
  - 横着はせずに、必要な部分を必要なだけコピーして貼り付けること！

# 第11回からのレポート

- 第11回、第12回：プログラムの途中経過を提出
  - レポートというよりは進捗報告とバックアップ
  - タイトルとプログラムが貼り付けられていれば2点
    - 実行画像はなくても良い
    - 特に減点もない、ただし、遅れ提出は認めない
      - 締め切りまでに出さなければ2点損することになる
    - 何回提出しても良い、  
作業が進むごとに提出してバックアップしておくこと
- 第13回：課題のレポート提出
  - レポート提出で最大40点の評点となる
    - プログラムの途中経過だけでは評価しない



# まずはプログラムを作ること

---

- レポートの書き方や評価については次回説明
  - まずはプログラムがなければ話にならない
- プログラム開発の着手が遅れた場合は、  
単位を落とす可能性が高くなる
- 遅くとも、第13回の開始までにはプログラムを完成させる
  - 第13回が7/23(木)、締め切りが7/24(金)
    - 第13回になってからメールで質問などしていたら、  
まず間違いなく間に合わない

# 最終回と、制作課題レポートについて

- **最終回（第13回）：7/23(木)** 月曜授業日
  - 第12回と13回の間隔が短い
  - 授業日に作業せずに後回しにしよう、、  
などと考えていると単位を落とすので気を付けること
- 制作課題の締め切り：最終回講義日の18:00まで
  - 成績提出の締め切りもあるため、待てても7/24まで
    - 未提出者への通知を行うことを考えると7/23が限度
  - 最終回までにプログラムが出来ていれば問題はない
    - **11回・12回の作業を後回しにすると、  
本当に単位を落とします**

# 遠隔授業期間中の質問

## • 困ったら早めに質問・相談！！

- 学生側から質問されないと、  
誰が困っているのか、何が分からないのか、分かりません
- メール：やり取りに時間はかかるが一番確実
- Zoom：授業時間中限定
  - 時間は限られるが、作業中の画面を見ながら教えられるので、問題を短時間で解決出来る可能性が高い
- Line OpenChat：授業時間中限定
  - 文字だけのやり取りに限定（画像アップロードは禁止）
  - 質問内容が他の学生にも分かるので注意すること
    - プログラムの全文貼り付け等は厳禁！

# 授業についての質問メールについて

[授業名(曜日時限)]についての質問	} 件名
～先生	} 誰宛か
[授業名(曜日時限)]を受講しています、 20AA999の九産太郎です。	} 何者か
(質問内容)	} 用件
--	
20AA999 九産太郎 九州産業大学 芸術学部 ○○学科 1年	} 署名

**2日返信がなければもう一度送って下さい (なるべく見落とさない)**

# 質問しなかった場合

---

- 自己責任です
- 自力で完成させられない、質問もしない、  
どうすることも出来ません
  - 対面教室に来ていたら、  
こちらから進捗を聞いたりしますが
- 早目に質問したうえで間に合わない場合は、  
多少の遅れ提出を認める場合があります

# 第11回からの授業中の質問について

- 授業中：対面で質問かZoomで質問
  - 対面対応が始まったら、授業時間中のメール返信はまず出来なくなる
  - なるべく授業中に対面かZoomで質問すること
  - Zoomの映像と音声は教室のプロジェクターに投影
    - 質問については教室から回答
    - なるべく声を出して質問すること
  - 個別対応については、対面も含めて順番に対応
    - 個別対応中はマイクをオンにすること
      - 特別な事情がある場合を除く
- 授業時間外：メールで対応

# 追加課題について

- 順次アップロードしていく、  
基本的には第13回終了後に取り組むことを想定
- いくつかの内容を出題するので、  
どれかをアレンジして提出すれば加点  
※現時点の案で、すべての資料を用意するかは未定  
※あくまでやる気のある学生向け
  - 3Dプログラミング、クラスとオブジェクト
  - 音との連動、画像処理
- ~~これの内一つを制作課題のベースとしても良い~~
  - 遠隔授業だと理解不足が怖いので取りやめ
    - どうしても難しい内容になる
      - これに惑わされると危険なので・・・
  - アレンジしたいなら、制作課題を完成させてから
    - ある程度プログラムが出来てから、一部を取り入れるのはOK

# 各回のアレンジ例

---

- 1～3回
  - 対象外
  - マウスで絵を描くプログラムを他で応用するのは可



# 第4回：静止画

- この回単体でのアレンジでは単位は出ない
- 第4回で作った静止画を背景にして、  
そのうえで図形を動かすようにしてアレンジする
- 第7回や第10回の内容を組み合わせる
  - 背景の一部を動かす
    - 背景の上で図形を動かす、沢山の図形を動かす
- 第8回の内容を組み合わせる
  - 雛型Cを参考に、  
マウスで背景の一部をクリックしたら動き出すなど
- 背景で繰り返しを使う、図形を多く配置して動かす、  
条件分岐で色々な動きをさせる、図形をメソッド化して配置  
など

# 第5回：図形を動かす

- この内容まで理解出来ていれば何とか単位は取れる
  - 本当にギリギリ単位が取れるかも…といったライン
    - 他のレポート、小テストが6割以上が最低条件
  - **ただし、図形を多く動かすことが前提**
  - 逆に言うと、第5回の内容が理解出来ていない場合は、単位が危ない(変数を宣言して動く図形を増やせば何とか単位を取れるかも)
- 動く図形を増やす、動きの種類も増やす
  - 変数を宣言して、その変数で図形を動かす
    - なるべく同じ変数を使いまわさない
      - 意図して使いまわすなら工夫点になるが、図形が沢山あっても変数が少ないのは駄目

# 第6回：ジェネラティブアート

- 条件分岐、繰り返しを使わないと高評価は難しい
- テキトウに図形の種類や色を変えただけ、等は工夫点にはならないので注意すること
  - ある意味単位を落としやすい回なので注意
  - 最低でもif文を使うことが条件
- 条件分岐でパターンを変えるのを主に考えると良い
  - 時間経過や、角度、座標によって色や形を変える
- 繰り返して沢山の図形を描くなども
- 座標変換を使いこなした工夫等も考える
  - 回転体を増やす、回転体を移動させるなど
    - 移動が出来れば、  
条件分岐で動きを増やせば工夫点としやすい

# 第7回：条件分岐

- 条件分岐を使って、色々な動きをさせる
  - 図形の数(変数の数)を増やす
    - 変数をたくさん宣言して使えば、それだけでも点数になる(使っていない変数はカウントしない)
  - 条件分岐の種類を増やす
    - 同じ動きをする図形を複数おいても、点数は高くない(工夫点としてカウントはするが、点数は低い)
- 第10回と組み合わせる
  - 配列を使ってたくさんの図形に色々な動きをさせる
- 条件分岐が難しい場合は、兎に角たくさんの図形を動かすこと(図形の種類、動きの種類を増やす)
  - 条件分岐も使わない場合には点数は低くなることは覚悟しておくこと(図形の数も少ない場合には単位も出なくなる)

# 第8回：ゲーム

- ゲームをアレンジしていけば良い
- 動きの違う敵を増やす、触れたら得点アップ、得点ダウンする敵を作る
  - まずは条件分岐で違う動きをする敵を増やしていく
  - 当たり判定の使い方に注意すること
    - 自分のサイズ、敵のサイズを間違えないように！
- 敵に当たったらスタート位置に戻すようにする
- 時間制限性にしてみる、残機性にしてみるなど
  
- 難易度を上げる、は工夫点になりにくいので注意
  - タイミングを工夫等もカウントするが、座標の工夫は高い評価にはならない
    - 数が多ければ、累計で高評価には出来る
  - 開発者にとって簡単でも、ユーザーにとっては難しい
    - ゲーム作りでありがちな失敗例ですね

# 第9回：繰り返して模様を描く

- 円形模様を増やす
  - 同じものでなければ、一つ一つを工夫点としてカウント
  - 当然だが、色を変えただけ等は別
- 円形模様の配置方法を工夫
- 円形模様内で条件分岐を使う
  - for文の中でif文を使うことを意識する
- 円形模様を動かす
  - 円形模様を条件分岐などで色々な動きをさせる
  - 繰り返し回数が多いと、処理しきれないので注意すること
- 特に他の回を混ぜなくても応用しやすい

# 第10回：配列

---

- 図形の種類を増やす（配列を増やす）
- 条件分岐で動きを変える
  - 跳ね返った時に色を変える、形を変える、速さを変える
  - for文の中で、if文を上手く使っていくと良い

# 注意：バックアップについて

- **バックアップはしっかりとること！！！！**
- 数時間分の作業が消えた・・・等ならないように
- ある程度の時間作業したら、  
パワーポイント等にプログラムを貼り付けてバックアップしておく
  - スライドを上書きするのではなく、新たに作成すること
  - Moodleのオンラインコメントにも貼り付けておくにより安全
    - スライドは複数提出しておき、オンラインコメントには最新のプログラムを貼っておくと良い
- メモ帳アプリ等に貼り付けておく、等、2重3重でバックアップ
  - 自分宛てにメールを送っておくのも非常に有効
    - レポートと同じ要領で全て選択、コピー、貼り付け



# プログラム開発時の注意

- まずは、一通り動く状態にすることを旨指すこと
  - 細部にこだわるのは後からでいい
- 一気にプログラムを書かないこと
  - こまめに実行して（何なら1行ごとに）、エラーなく動くのかを確かめながら作っていくこと
- 過去のレポートからコピペするときはバックアップを取ってから
  - 括弧の対応が合わない、同じ変数があるなどでエラーが起きやすい
- 何していいか分からないなら、図形を増やす、動きの種類を増やす
  - 沢山変数を宣言して、色んな動きをさせていけば、単位は何とかなる

# エラーが取れないとき

- エラー取りに時間がかかるなら、  
授業時間中ならとにかく質問すること
  - 目安として5分以上悩んで駄目なら質問
- 授業時間外
  - メールにプログラム全文を貼って質問
    - 画像ではなく文字を貼り付ける（レポートと同じ）
  - 10分悩んでもエラーが取れないなら、  
取り合えずメールを送っておく
    - 解決したら「解決しました」と送り直せばいい
    - どうにもならないなら返事を待ってから作業を再開
      - 他の授業のレポートなどをやっておく
    - 1日待って返事がなければ催促する
      - 2日目出しても駄目ならOpenChatにも書き込む

# 単位を落とす例

- 不正（まず間違いなくバレます）
  - インターネット上のプログラムや、Processingの参考書のプログラムをコピー
    - ただし、事前に相談があればアレンジを認めることもある
  - 友人と(殆ど)同じプログラムを提出
    - 全員0点にします
- 手抜き
  - 雛型やサンプルと同じプログラムを提出
    - 当然、0点です
  - レポートと同じプログラムを提出
    - その回のレポートに多くの時間をかけたのであれば、それなりの点数は取れるが、元のレポートが手抜きなら単位を落とす
  - そもそも、プログラムが短い
    - 最低でも200分程度の努力が見えない場合は0点
    - 短いプログラムでも、すごい工夫をしてきた場合は除く
      - まず無いケース。もしあれば追加で説明を求めます
- 不正が疑われるプログラムを提出してきた場合は、電話で試問を行います(自分のプログラムを説明できない場合は不可とする)
  - 意図して不正を行わない限り、こんなことにはなりません

# 色々書きましたが

- しっかり作りこんだプログラムなら高評価になる
  - 凝ったことをするなら、条件分岐や繰り返し、メソッドなどは自然に使うことになる
- 授業時間分プログラム開発をすれば、そうそう不可にはならない
  - ここまでの内容を全く理解していない場合は除く
  - 最低でも200分以上、プログラムを書く時間に充てること
- 点数を意識するなら：条件分岐、繰り返し、メソッド
  - 分かりやすい工夫点になりやすい
  - 変数を増やす、その変数を使った条件分岐を増やす
  - 繰り返しやメソッドの中で条件分岐を使う、など
- **作っていて楽しいものを選んで開発すること**
  - どうせなら作っていて楽しいものを作ろう
    - 好き好んでつまらない作業をする必要はない

# レポートチェックリスト（第11回）

- ミニテストを受験した(レポートの前と後どちらでも)
  - 今回から問題数は減らします
- 100分以上プログラム開発を行った**
  - プログラムを書いている時間が100分
    - 何をするか悩んでいる時間や、エラーに悪戦苦闘している時間は含まない
- PowerPointでレポートを作成した
  - タイトル、プログラムの最低2枚
    - 実行画像もあっても良い
    - プログラムは必須、画像で貼り付けたものは対象外
    - 途中経過のレポート作りに時間をかけないこと！
- Moodleでレポートを提出した
  - Moodleに質問を書いていない

# 可となる最低ラインについて

- 小テスト6点以上（10点取りましょう）
  - 何回か解けば必ず満点に出来る、満点を取れないのはただサボっているだけ
- 1～12回のレポート6割以上
  - 1回～3回で11点：出していれば11点取れている
  - 11回、12回で4点：出せば4点取れる
    - ここまでで21点
  - 残り7回で15点取れていればギリギリ単位習得ライン
    - 3点×5回で単位が取れる
- 第13回：24点(6割)
  - ギリギリ手抜きと判定されない程度のプログラムとレポート
- **ただし、ギリギリを狙うと大抵落ちる**
  - **少なくとも5点程度は余裕を持つておくこと**