

# COMP4008 Programming

## Group Coursework

### Overview

In groups of around five students, you will work together to produce a Python program for playing a new variant on the card game *Uno*.

### Group Formation

The information about groups is posted on the Moodle page (Group Project section).

### Description of the Game

*Uno* is a card game based on a deck of cards. There is a description of these cards here:

<https://www.unorules.org/how-many-cards-in-uno/>

For the purposes of this coursework, please ignore the blank cards.

There is a basic tutorial on how the game works here:

<https://www.youtube.com/watch?v=sWoSZmHsCls>

and there is a more detailed description here:

<https://www.unorules.org>

In addition to those rules, your game should include one further rule. At the beginning of each turn, before playing any card, the player must choose one card from their hand, put it on the bottom of the discard pile, and then take the top card from the draw pile and put it in their hand.

There is a rule in the game that the player should shout “Uno” once they have one card in their hand, and if they forget to do so, an opponent can shout this instead. Implementing this rule is not necessary in this variant of the game, though you could implement it as an extension.

Your program will be simulating a single round of the game, i.e. playing until one of the players is left without cards. Once the game round ends, you need to show the scores of each player (based on the cards they still have). You do not need to implement a scoring mechanism for playing multiple game rounds as playing multiple rounds of the game is not a compulsory part of the task.

Some versions of the game include *Swap Hands* and *Shuffle Hands* cards. Implementing these is not necessary in this variant of the game, though you could implement them as an extension.

### Task

The overall task in this coursework is to create a Python program to allow a human player to play a single round of this new variant of the Uno game, playing against one or more computer opponents. You must use the Pygame framework to implement this. Details of this are at <https://www.pygame.org>, and it will be introduced in lectures.

You should include the following in your solution:

- A simulation of the deck of cards and the basic gameplay, including the initial dealing of 7 cards to each player, creating the draw pile and discard pile, allowing the human player to play turns of the game (including the new rule), controlling computer players who take it in turns with the human player. The effects of the non-numbered cards (skip turn, reverse, take 2, wildcard, and wildcard+take4) should be included.
- A turn consists of the current player swapping one of their cards (the rule introduced in this variant of the game), choosing a card to play onto the discard pile (if they can), repeating that

until they cannot play then choosing the top card from the draw pile and seeing if they can play that card. If they cannot, then their turn ends there.

- You might find it useful to initially create just the 76 numbered cards, with matching just on numbers/colours, get that working, and then add the other cards later in the development of the program.
- You should use the facilities in Pygame to create a visual interface to this, so that the user can see pictures of their cards, see the card on top of the discard pile, and watch the cards being played onto the discard pile by other players.
- You should implement computer players for the game. Initially, this could consist of the computer playing a random card that matches the card on the top of the discard pile. Later on in the development of your program, you could make it more sophisticated. For example, the computer might keep track of which cards have already appeared in the game, and choose which card to play based on that information. You might also think about strategies for when best to play one of the non-numbered cards (for example, the computer player might hold back a “draw 2” card until the opponent after them in the turn sequence has a very small number of cards).
- When a player (human or computer) wins, the program should display a congratulation message and the scores of the players, based on the values of the cards they have; the higher is the total value, the worse it is for the player.
- Importantly, you must incorporate an aspect of Nottingham into the visual design of your cards. That could be the University or the city. Games that do not include this aspect will fail the assessment. For example, your cards might include images from the campus or city on the cards, as illustrated here:



- Your group should implement some extension(s) to the game. This could be a better computer player, a new kind of card, an additional rule for the game, a scoring mechanism across multiple rounds, more sophisticated visuals, or any other additional aspect of the game.

There will be time set aside for working on the coursework in the lab sessions, but you will not be able to complete the task just in that time, you will need to do the majority of work outside of that time.

## Working on the Project

This is a group project and every member has to contribute to the project. The contribution of each member must include writing substantial amount of code.

Teams have to make use of Gitlab to collaborate. The main reason for using Gitlab is to assist the development process and learn a technology used by the majority of software developers. Also, commits to Gitlab can later be used as evidence of individual contributions.

Although not compulsory, it is a good idea to select a team lead. Team lead will usually be responsible for organising team meetings, allocating tasks, leading discussions, etc. Team leads will usually be expected to do a little less coding than the other team members.

## Marking Criteria

The following criteria will be used in assessing the coursework:

Criterion	Weight
The overall functionality of the program in implementing the gameplay	40%
The quality of the computer players	15%
The quality of the visual/user-interface aspects	15%
Coding quality	10%
Quality of the demonstration	20%

Group grades will be given according to the following guidelines:

80 and above	All aspects of the task have been completed to a high quality. The submitted work also contains high quality extensions to the project, and/or goes beyond regular expectations in quality of design and implementation.
70-79	All aspects of the task have been completed to a high quality, possibly including some attempt at extensions to the core task.
60-69	Almost all aspects of the task have been completed to a good standard, perhaps with a few minor errors or misunderstandings.
50-59	Most aspects of the task have been completed to a competent standard, perhaps with a few errors or misunderstandings.
40-49	Some attempt has been made at the task, but important aspects of the functionality are missing and/or there are significant errors or misunderstandings.
20-39	A small attempt has been made at a few aspects of the task, but the majority is missing, wrongly designed/implemented, or erroneous.
0-20	Minimal or no attempt at the task, or no submission.
0	A zero mark will be given for any submission that makes no attempt to include an aspect of Nottingham in the design, regardless of the quality of the remaining work. A zero mark will be given for any submission that makes no attempt to include the new rule about swapping one card at the beginning of each turn. A zero mark will be given for any submission if the group did not adequately make use of Gitlab during the project; for example, developing the game without the use of Gitlab and then uploading the final code to Gitlab is inadequate.

A grade will be given to the group, which will then be adjusted based on assessment of each students' contribution to give a final mark.

## Submission and Demonstration

You should submit your code by giving a link to your code repository on Moodle. Details of this will be explained nearer the time.

The submission deadline is Monday 12<sup>th</sup> December at 11am. The late submissions will receive a 5% penalty for each day of the delay, up to 11am on Wednesday 14<sup>th</sup> December. Any submissions after that will receive a zero mark, except where students have extenuating circumstances or a support plan.

Around half-way through the coursework, and at the end of the coursework, you will be asked to fill out a form. This will ask you to briefly explain and quantify your contributions to the project, and ask you to assess the contributions made by other group members. This will support the assignment of an individual mark for each person in the group, although other mechanisms to evaluate individual contributions will also be used. You will also have the opportunity to identify if group members are not participating at all in the group. If you did not participate in the project, then you will receive a mark of zero.

Please contact the module convenors, [Colin.Johnson@nottingham.ac.uk](mailto:Colin.Johnson@nottingham.ac.uk) and [Daniel.Karapetyan@nottingham.ac.uk](mailto:Daniel.Karapetyan@nottingham.ac.uk) if there are any concerns about bullying, abuse, discrimination or harassment within the group.

In the lab on Wednesday 14<sup>th</sup> December, you will be required to demonstrate your program to one of the teaching assistants or lecturers. Each group will have 7 minutes to demonstrate their program, explain how it works, and discuss how their group went about the process of designing and implementing their solution, with further 8 minutes allocated for questions.

## Academic Integrity

This is a group assessment that should consist of the unaided work by the group. **You are not allowed to adapt an existing Uno implementation.** If your solution makes use of someone else's code (e.g. our code from the classes, examples taken from forums, open-source projects, etc.), please make it clear which aspects of the code are your group's work through the use of comments. The University has [detailed advice about academic integrity](#), and submissions that demonstrate a lack of that integrity will be treated under appropriate disciplinary procedures.