



# 3장

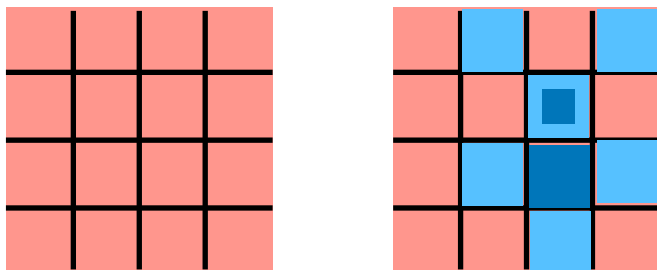


# 컴퓨터 시뮬레이션이란?

- 컴퓨터 시뮬레이션?
  - 어떤 시스템(체계)의 수학적 모델을 만들고, 모델을 표현하는 프로그램을 작성한 다음, 모델을 시간에 걸쳐 진화시키는 방법으로 시스템을 연구할 때 컴퓨터 시뮬레이션이 사용된다.
- 컴퓨터시뮬레이션
  - 1. 모델만들기
  - 2. 모델을 운영하기
    - 어떻게? (스스로) 진화시키는 방법으로 시스템을 운영하기.
- 이 책에서는 컴퓨터시뮬레이션의 예로, “콘웨이의 생명게임”을 든다.

# 콘웨이 생명게임 이란?

- 영국사람, 존 코웨이가 만듦.
- 세포 자동자, 셀룰러 오토마타의 한 예시 (세포들이 자동적으로 생성되는데서 착안을 얻었다는 뜻)  
[https://ko.wikipedia.org/wiki/%EC%84%B8%ED%8F%AC\\_%EC%9E%90%EB%8F%99%EC%9E%90](https://ko.wikipedia.org/wiki/%EC%84%B8%ED%8F%AC_%EC%9E%90%EB%8F%99%EC%9E%90)
- 어떤 그리드 상에 존재하는 색깔있는 셀(세포)의 집합으로서 각 셀은 인접하는 셀의 상태를 정의하는 규칙에 따라서 시간의 흐름에 따라 진화한다.



# 게임의 규칙

- 1단계)  $N \times N$ 셀로 이뤄진 그리드를 생성한다.
- 2단계) 콘웨이의 생명게임 규칙을 적용해 시간의 흐름에 따른 시스템의 진화를 시뮬레이션 한다.
- (각 셀은 인접하는 셀의 상태를 정의하는 규칙에 따라서 시간의 흐름에 따라 진화한다.)
- 3단계) 단위 시간이 지날 때마다 시간의 상태를 보여준다.
- 4단계) 최종 결과를 파일로 저장한다.
- 0단계) 시스템의 초기 조건은 임의적 분포 혹은 사전 정의된 패턴으로 설정한다.
- 구성요소
  - 1차원 또는 2차원공간에서 정의된 속성. —> 공간!
  - 시뮬레이션의 각 단계에서 이 속성을 변경하는 수학적 규칙 —> 변경 규칙 (인접하는 셀이 변한다고 했는데 어떻게 변하게 할꺼야??)
  - 시스템의 진화에 따른 상태 변화를 표시하거나 포착하는 수단 —> 변화 상태여부 (어떻게 알아 낼꺼야?)

# 여기서 다루는 내용

- 4개의 규칙으로 간단한 패턴으로 그리드가 어떻게 변화하는지를 보여준다.
- 패턴의 종류 : 글라인더, 블링커패턴
- 그래서.. 이 프로젝트에서 다루는 내용은,
  - matplotlib의 imshow를 사용해 2차원 데이터 그리드 나타내기
  - matplotlib으로 애니메이션 만들기
  - numpy 배열 사용
  - % 연산자를 사용해 경계 조건 설정하기
  - 값을 임의로 분포하기

# 콘웨이 생명 게임 규칙

- 어떤 셀이 ON상태이고 ON상태인 인접 셀이 2개 미만이면 그 셀은 OFF로 바뀐다.
- 어떤 셀이 ON상태이고 ON상태인 인접 셀이 2개 ~ 3개 이면 그 셀은 ON상태를 유지한다.
- 어떤 셀이 ON상태이고 ON상태인 인접 셀이 3개 이상이면 그 셀은 OFF로 바뀐다.
- 어떤 셀이 OFF상태이고 ON상태인 인접셀이 3개이면 그 셀은 ON으로 바뀐다.

# 인접셀을 정의

- 경계 조건(Boundary Conditions)의 개념을 도입
- 경계 조건은 그리드의 가장자리, 즉 경계에 위치한 셀에 적용되는 규칙
- 인구 과소, 인구과다

- 이 규칙들은 어떤 유기 생명체 그룹이 시간의 흐름에 따라서 취할 수 있는 기본적인 전략을 반영한 것이다.
- 인구과소(ON 상태인 인접셀이 2개 미만)와 인구과다(ON상태인 인접셀이 3개 초과)의 경우 해당 셀을 죽이고 (OFF상태로 만듦), 인구가 균형을 유지할 때는 셀의 ON상태를 유지하거나(2번 규칙) OFF인 셀을 ON으로 만든다. (4번 규칙)
- 하지만 그리드의 가장자리에 위치한 셀은 어떻게 해야 할까? 이 셀들의 인접셀은 무엇으로 해야 할까? 이 질문에 답하기 위해서는 경계조건의 개념을 도입해야 한다.
- 이번 프로젝트에서는 환상형 경계조건을 사용해야 한다.



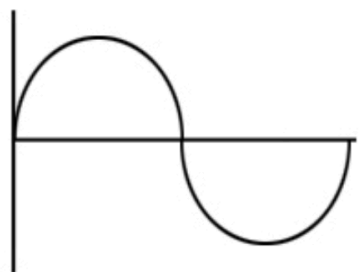


# 4장

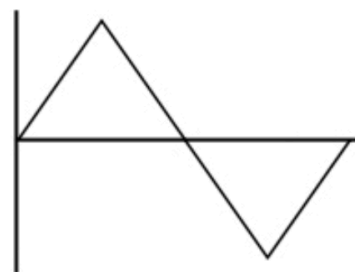


# 소리와 음악파일

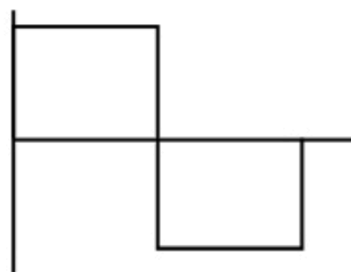
- 소리의 3요소 : 음량(Loundness), 음고(pitch), 음색(Timbre)
  - 음량: 소리의 크기
    - 단위: 데시벨 [dB] , 조금더 엄밀히 하면, 데시벨 SPL [dB SPL] (\*SPL: Sound Pressure Level)
  - 음고: 소리의 높이
    - 단위: 주피수 [Hz]
  - 음색: 소리의 색깔
    - 파형의 모양



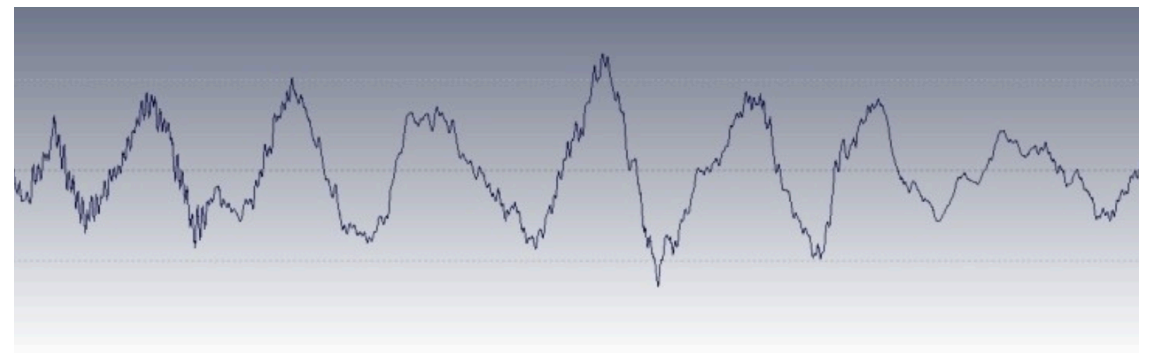
(1) 사인파(Sine Wave)



(2) 삼각파(Triangle Wave)



(3) 사각파(Square Wave)



<그림. 사자가 으르렁대는 소리 파형>

# 카플러스 스트롱 알고리즘

- 배음(Overtone)? : 피아노에서친 “도”와 기타에서친 “도” 소리가 왜 다르지? -> 배음 때문이다. (=컴퓨터에서의 “도”와 기타의 “도”가 다른 이유도 역시 배음!!!)
- 모든 소리는 2, 3, 4차 등의 정수배로 배음 발생.
- 컴퓨터로 현악기에서 만들어진 소리를 시뮬레이션 하기 위해서는 기본 주파수와 배음 (overtone) 을 모두 생성할 수 있어야 한다.  
—> 카플러스 스트롱알고리즘을 이용

# 여기서 다루는 내용

- 카플러스 스트롱 알고리즘 : 어떤 음계(일련의 서로 관련된 음들)의 기타 음을 5개 생성한다.
- 그리고 이 음들을 생성하는 데 사용된 알고리즘을 시각화하고 WAV파일로 저장한다.
  - 파이썬 deque 클래스를 사용해 원형 버퍼 구현하기
  - numpy배열과 ufuncs의 사용
  - pygame을 사용해 WAV 파일 재생하기
  - matplotlib을 사용해 그래프 그리기
  - 5음 음계 연주하기
  - + WAV 파일의 포맷 및 5음 음계 내에서 음을 생성하는 방법

# 필요한 지식

- matplotlib, numpy
- pygame
- argparse
- 자료구조 (스택, 큐, deque)
- 문법 ( ~ 클래스까지)