

C언어 기반 '너구리' 게임 완성하기

항목	내용
과제명	크로스 플랫폼 터미널 게임 'Nuguri' 완성 및 확장
제출 기간	2025년 11월 20일 (목) ~ 2025년 12월 4일 (목) 23:59 (2주간)
제출 방법	아래에서 언급
배점	100점 + α
대상 학년/강좌	3학년/고급자료구조

1. 과제 목표

본 과제는 제공된 초기 단계의 게임 소스 코드(nuguri.c)를 분석하여, 이를 완성된 형태의 게임으로 발전시키는 것을 목표로 한다. 학생들은 이 과정을 통해 다음 역량을 함양해야 한다.

- 크로스 플랫폼 프로그래밍: 운영체제(Windows, Linux, macOS)에 종속적인 시스템 API(입출력 처리 등)를 이해하고, 전처리기(Pre-processor)를 활용하여 모든 환경에서 동작하는 이식성 높은 코드를 작성한다.
- 협업 도구 및 버전 관리: GitHub를 활용하여 개발 진행 상황을 체계적으로 기록하고 관리하는 습관을 기른다.
- 자료구조 및 알고리즘 응용: 2차원 배열 기반의 맵 데이터 처리, 충돌 감지 알고리즘 개선, 게임 루프 최적화를 수행한다.

2. 과제 내용 및 방법 (Contents and Method)

가. 기반 코드 (Baseline)

- 제공된 nuguri.c와 map.txt를 분석하여 게임의 기본 로직(이동, 충돌, 맵 로딩)을 이해한다.
- 현재 코드는 Linux/macOS 환경(termios.h)에 맞춰져 있으므로, 이를 기반으로 확장을 시작한다.

나. 필수 구현 기능 (Requirements)

1. 크로스 플랫폼 지원 (Cross-Platform Compatibility):

- 현재 코드는 Windows에서 컴파일 및 실행이 불가능하다.
- #ifdef _WIN32 등의 전처리 지시어를 사용하여, Windows 환경에서는 <conio.h> 등을, Linux/macOS에서는 <termios.h>를 사용하여 키보드 비동기 입력을 처리하도록 코드를 수정해야 한다.
- 화면 클리어(cls vs clear), 대기 함수(Sleep vs usleep) 등 OS 의존적인 기능들을 모두 호환되게 구현해야 한다.

2. 게임 완성도 향상:

- 생명력 시스템: 플레이어에게 3개의 생명(Heart)을 부여하고, 적과 충돌 시 생명이 감소하며 시작 지점으로 돌아가는 로직을 추가한다. (생명 0이 되면 Game Over).
- 타이틀 및 엔딩 화면: 게임 시작 전 타이틀 화면과 게임 오버/클리어 시 나타나는 엔딩 화면을 별도로 구현한다.

3. GitHub를 이용한 버전 관리 (Version Control):

- 프로젝트 시작과 동시에 GitHub Repository를 생성한다. (Private/Public 무관하나 제출 시 접근 가능해야 함)
- 기능 하나를 구현할 때마다 커밋(Commit)을 수행해야 하며, 최소 10회 이상의 유의미한 커밋 로그가 남아야 한다.
- 단순히 최종 결과물만 한 번에 업로드하는 행위는 인정하지 않는다.

다. 선택 구현 기능 (Optional - 가산점)

- 동적 맵 할당: define으로 고정된 맵 크기를 제거하고, map.txt의 크기를 읽어 malloc을 통해 동적으로 맵 메모리를 할당하도록 개선.
- 사운드 효과: 각 OS의 시스템 비프음 등을 활용한 간단한 효과음 구현.

3. 제출 파일 (Deliverables)

아래의 파일들을 하나의 ZIP 파일로 압축하여 제출한다. (파일명: 팀장학번_이름_nuguri.zip)

1. GitHub Repository 주소: README.txt 최상단에 명시.
2. 소스 코드 (.c, .h): 완성된 소스 코드 파일 일체.
3. 맵 데이터 (map.txt): 게임 실행에 필요한 맵 파일.
4. 실행 보고서 (README.md 또는 PDF):
 1. 학번 / 이름
 2. GitHub 주소 (필수)
 3. OS별 컴파일 및 실행 방법 가이드 (Windows/Mac/Linux 각각 명시)
 4. 구현 기능 리스트 및 게임 스크린샷
 5. 개발 중 발생한 OS 호환성 문제와 해결 과정 기술

4. 평가 기준 및 배점 (총 100점)

평가 항목	세부 내용	배점
가. GitHub 활용(30)	Repository 생성 및 URL 제출 여부	10
	커밋 로그의 충실성 (최소 10회 이상, 개발 과정이 타임라인으로 보여야 함)	20
나. 플랫폼 호환성(30)	Windows 환경에서 컴파일 및 정상 구동 확인	10
	Linux/macOS 환경에서 컴파일 및 정상 구동 확인	10
	전처리기(ifdef)를 통한 코드 분리가 깔끔하게 되었는가	10
다. 기능 구현(30)	생명력 시스템, 타이틀/엔딩 화면 구현 여부	20
	기존 버그 수정 및 게임 플레이의 매끄러움	10
라. 보고서(10)	실행 가이드의 명확성 및 문제 해결 과정 서술	10
마. 감점 항목	GitHub 커밋 기록이 없거나(한 번에 업로드), "Update" 같은 무의미한 메시지만 있는 경우	-20
	특정 OS에서 실행 불가 시	-15

5. 중요: 생성형 AI 사용 금지 (Strict Policy on AI)

본 과제는 학생의 고유한 알고리즘 설계 및 기존 코드 분석 능력을 평가하기 위함입니다.

ChatGPT, GitHub Copilot, Gemini 등 일체의 생성형 **AI(Generative AI)** 도구를 사용한 코드 생성, 수정, 디버깅을 엄격히 금지합니다.

제출된 **GitHub**의 커밋 로그 시계열 분석 및 불시 구두 테스트(**Oral Test**)를 통해 본인 작성 여부를 검증할 예정입니다.

AI 사용 흔적(부자연스러운 주석, 복사/붙여넣기 패턴 등)이 발견되거나 소명이 불가능할 경우, 해당 과제는 0점 처리되며 이번 학기 성적은 F가 부여됩니다.