CSED451: Computer Graphics (Fall 2025)

Assignment 1: 2D Drawing

Submission due: Wednesday, September 24, 2025, 11:59PM

1. Introduction

In this course, we aim to develop a simple game to build a comprehensive understanding of computer graphics. Specifically, this semester we will develop so-called "Bullet Hell shooters." Bullet Hell shooters are a subgenre of shoot-'em-up games, where enemies typically emit massive numbers of bullets. By developing Bullet Hell shooters, you can learn how to optimize a large batch of entities with rendering and collision as well as basic graphics pipeline handling using OpenGL.

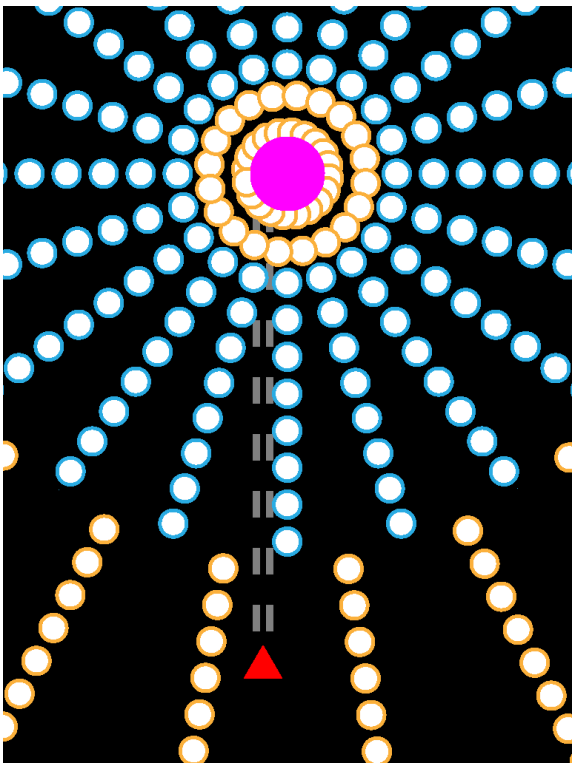

Figure 1. Bullet Hell shooter implementation example.

In Assignment 1, you will develop a simple Bullet Hell shooter using OpenGL 2D drawing APIs. You can check out some motivating references here: Touhou 20th Fossilized Wonders, Ikaruga, Mushihimesama, Dodonpachi Saidaioujou, and so on.

2. Requirements

Minimum Goals

- Player
  - The player can move in four directions: up, down, left, and right.
  - The player cannot escape an in-game bounding box.
  - The player can shoot attacks when the corresponding keyboard button is pressed.
  - The player starts with a fixed number of lives.
  - The player loses a life and disappears when its hitbox collides with Bullets.
  - If the player has no lives left, the game ends.
  - Otherwise, the player reappears after death in a short period of time.
- Attack
  - An attack deals damage to the enemy on hit.
- Enemy
  - There is only one Enemy in this assignment.
  - The enemy loses health when hit by attacks and is destroyed once its health reaches zero.
  - The enemy emits Bullets.
- Bullet
  - A bullet disappears once it leaves the screen.

Additionally, different entities must be visually distinguishable in both shape and color.

Additional Goals

To receive full credit, you must establish at least two additional goals beyond the minimum requirements above. These goals should be sufficiently challenging. Examples include:

- Complex shape rendering
- Primitive rendering optimization
- Global camera animation

Note that credit will be given **only for graphics-related features**. Little or no credit will be given for mere game features (e.g., pause menus, gamepad support, multi-stage boss fights, etc.). You may choose additional goals from the examples above, but remember that more generous credit will be given for original goals you propose yourself.

Non-goals

- You are not required to implement a hierarchical structure for entities in this assignment.
- You are not required to render fancy sprites in this assignment. It is enough to render entities as simple polygons.

3. Submission

Each team must submit one report along with the program's source code.

Report

Your report can be any document file format. However, please do **not** use `*.hwp`. Your report should be brief and concise, and include the following:

- Basic information:
  - The name of the team.
  - Each team member's name, department, student ID, and HEMOS ID.
- Technical details:
  - The development environment used (e.g., IDEs, framework versions, etc.).
  - Any additional technical background necessary to understand the source code.
- Implementation details:
  - An outline of your program's features.
  - Detailed description of your additional requirements.
  - The rationale behind your program's design.
  - How you implemented your design.
  - Any additional background necessary to understand the design and implementation.
  - Do **not** include your program's source code in the report.
- End-user guide:
  - How to run and operate your program.
  - Screenshots of each feature you described.
- Discussions/Conclusions:
  - The obstacles you encountered during development and how you resolved them.
  - Idea on how you can improve your program further.
  - What you learned or concluded from this assignment.

- References:
  - **Clearly state references** for any work not created by your team. (e.g., something from tech blogs, Stack Overflow posts, etc.)
  - **If any part of your assignment is found to be someone else's work without proper references, it will be considered cheating.**
- **AI-assisted coding references**:
  - If you used any AI-assisted coding tools in writing your program (e.g., OpenAI Codex, Claude Code, Google Gemini, Cursor, Amazon Kiro, GitHub Copilot, etc.), clearly state which tools you used, how you used them, and where in your program they were applied.
  - You should also mention AI tools you consulted, even if their outputs were ultimately not included in your final program.
  - Provide an estimate of what percentage of your assignment was completed with AI assistance, and explain your reasoning. *This percentage will not affect your credit at all*.

Program

- Include a brief `README.md` file that describes your source code files.
- Your source code is expected to meet the technical requirements specified in the provided setup document.

4. Scoring Criteria

Basic Rules

- **Minimum requirements** (40%)
  - The program must run properly and meet the minimum requirements.
- **Additional requirements** (20% + *10% extra credit*)
  - You must define and implement additional requirements on your own.
  - Two valid additional requirements → +20% credit.
  - Three or more valid additional requirements → up to +30% total (20% base + 10% extra).
  - You may propose more than the minimum if you are unsure whether certain requirements will qualify for full credit.
- **Program design and implementation** (30%)
  - The program must be designed and implemented properly to meet the requirements using OpenGL.

- **Report structure and `README.md`** (10%)
    - The report and `README.md` must be clear and concise.
    - The structure and formatting of the report will also be evaluated.

Deduction

- If you miss the submission deadline, your score will be reduced by 10%.
- For every additional 24 hours late, your score will be reduced by another 10%p.
- For example,
    - Submitting 8 hours late: -10%
    - Submitting 25 hours late: -20%
    - Submitting 216 hours (9 days) late: no credit will be given.
- Note: After the submission deadline, you may only submit via the TA's email (`yoonha.hwang@postech.ac.kr`).

If you have any further questions, please post your question on Q&A board on PLMS.