

# Game Design & Programming I

CMPT 414 / MSCS 565 – Fall 2023

## Game #3 – Variation on Zombie Arena – 100 points

### Goals

To reinforce intermediate C++ and SMFL skills by creating a multi-level video game.

### Preparation

You must already have a Git repository named *{YourLastname}-work*. In Visual Studio, add to your existing *GameProg1* solution a new C++ *Empty Project*. This project will house the *Zombie Arena* clone that you will create and customize as described below.

### Requirements

Follow along in Chapters 8 through 13 of our textbook to build a game in the style of *Zombie Arena*, deviating from the written instructions to distinguish your game from the base version. Your customizations must include three (3) prescribed changes together with your choice of discretionary changes, as described below.

- ★ *Configuration and Correctness* 10pts
  - Project organization and settings.
  - Project resides in the same solution as your other games.
  - Program does not fail to compile or run due to improper config.
  - Program must not fail to compile or run due to syntax or logic errors.
  - Program compiles and runs without error.
- ★ *Code Style* 10pts
  - Program demonstrates consistent, readable coding style and follows best practices for C++ including, but is not limited to, indentation/spacing; naming of variables/functions/classes; architecture/organization; use of STL features.
- ★ *Prescribed Change #1 – Arena Obstacles* 10pts
  - Add randomly placed, static obstacles to your arena levels. You may use the wall texture for the obstacles or find a new graphic.
  - Player and Zombies must not be able to pass through an obstacle.
- ★ *Prescribed Change #2 – Player Upgrade* 15pts
  - Add a new pickup that upgrades the player's weapon to a multi-shot weapon. After obtaining the multi-shot weapon, the player shoots two or more bullets at once in at least two differed directions.
- ★ *Prescribed Change #3 – Enemy Behavior* 15pts
  - Differentiate one kind of enemy through its behavior pattern, not just size, speed, or health. You must choose one of the following:
    - Each time a crawler gets hit, its speed increases slightly.
    - Give Chasers 2 health – when hit they flee briefly before resuming the chase.
    - When Bloaters get down to 1 health, they stop and explode.

★ *Discretionary Changes – Your choice of the options below.*

**Up to 40pts**

- a) **Arena Shape** – change the shape of the arena so that it is rectangular and randomize the aspect ratio for each level.
  - Earn this credit only once. **(10pts)**
- b) **Additional Arena Obstacles** – think of and implement another kind of obstacle; e.g., pool of acid or wall of fire that damage player/zombie, etc.
  - Earn this credit at most twice. **(10pts each)**
- c) **Additional Player Upgrades** – think of and implement another kind of player upgrade besides rate of fire, damage, movement speed, or multi-shot.
  - Earn this credit at most twice. **(10pts each)**
- d) **More Enemy Behaviors** – differentiate another enemy as described above, or by coming up with your own idea.
  - Earn this credit at most twice. **(10pts each)**
- e) **Cosmetic Effects** – add visual and/or audio effects to your game.
  - Earn this credit only once. **(10pts)**
- f) **Your Own Idea!** – think of and implement another idea not listed here... but you must clear it with me first!
  - Earn this credit at most twice. **(10-20pts)**

***Advice***

You should break up your work over multiple days and commits. Your commit history should show your process, not just the final product. Push any local work to your remote GitHub repository regularly. Don't forget to write short but meaningful messages for every commit. (Tip: Consider using wording from these instructions as your commit messages.)

Test, test, test... and test again. Then test some more. When you think you've tested enough, go back and test yet again. Then get someone else to test for you while you test theirs. Etc.

***Submitting***

You must push your changes to GitHub before the due date.

Note: Pushing regularly will reduce the risk of losing your work, so do not wait until after you have made all changes and commits before pushing.