

# CPSC 427 - Video Game Programming

## Fall 2019/20

### Milestone 1: Skeletal Game - October 4, 2019

For this milestone, you should have a basic version of your target game. This basic game is expected to have comparable complexity to the first individual assignment. It should incorporate basic rendering, input-driven response, 2D motion, correct basic collision handling, event-driven/random response, and a minimal set of assets.

**(80%) Mandatory requirements:**

- **Rendering (25%):** functional rendering code and shaders for background and sprite assets.
  - Loading and rendering of textured geometry with correct blending and consistent asset drawing order (5%).
  - Working basic 2D transformations (10%).
  - Key-frame/state interpolation (smooth movement from point A to point B in Cartesian or angle space) (10%).
- **Gameplay (35%):** asset control and interaction.
  - Keyboard/mouse control of one or more character sprites. This can include changes in the set of rendered objects, object geometry, position, orientation, textures, colors, and other attributes (10%).
  - Randomized or hard-coded action of one or more characters/assets (5%).
  - Well defined game space boundaries (5%).
  - Correct collision processing including basic detection and avoidance. (preventing obvious penetrations) (15%).
- **Robustness (20%):** Supporting stable gameplay with continuing execution and graceful termination. In particular, you need to ensure that
  - the game runs without severe lagging;
  - the game resolution is consistent across different machines (in particular aspect ratio remains constant);
  - the game does not crash and terminates gracefully.

**(20%) Creative Component:** You should implement one or more additional creative elements. These can include additional integrated assets, rendering effects, or pre-emptive implementation of one or more features from subsequent milestones. To receive the full creative credits, the game

should have at least two small features outside the requirement (e.g. camera controls, simple rendering effects, etc.) or one advanced feature outside the requirement (e.g. enemy group behavior, sprite sheet animation, etc.). Structuring the game code using ECS design pattern ([https://en.wikipedia.org/wiki/Entity\\_component\\_system](https://en.wikipedia.org/wiki/Entity_component_system)) will be counted towards creative credits.

**Note:** You will receive full credit for features only if they are **fully** operational. Points will be deducted for buggy and incomplete implementation.

**Your submission should align with your proposed development plan:** Provide a write-up explaining how your milestone aligns with the plan. Explain all discrepancies and **submit an updated proposal** when such discrepancies occur.

**Submission:** Submit the code and associated documents using the course Git repository that's been set up for you at <https://github.students.cs.ubc.ca/CPSC427/team#>. The repository is hosted on the UBC servers and will be accessible only to enrolled students. *Note that each team member is also expected to submit their individual progress & feedback report via 'handin'.*