

# CPSC 427 - Video Game Programming

## Milestone 1: Skeletal Game

For this milestone, you should have a basic version of your target game. It should incorporate basic rendering, input-driven response, 2D motion, correct basic collision handling, event-driven/random response, and a minimal set of assets. Feel free to use the assignment 0 and 1 templates as a starting point for your game code. **Please structure the game code using the ECS design pattern** ([https://en.wikipedia.org/wiki/Entity\\_component\\_system](https://en.wikipedia.org/wiki/Entity_component_system)).(80%)

### *Mandatory Requirements:*

- **Rendering** (30%): Functional rendering code and shaders for background and sprite assets.
  - Loading and rendering of textured geometry with correct blending and consistent asset drawing order (10%).
  - Working basic 2D transformations (10%).
  - Key-frame/state interpolation (smooth movement from point A to point B in Cartesian or angle space) (10%).
- **Gameplay** (30%): Asset control and interaction.
  - Keyboard/mouse control of at least one character sprite. 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) (10%).
- **Stability** (20%): Support **stable** gameplay. In particular, you need to ensure that
  - the game runs without severe lagging;
  - the game code supports continuing execution and graceful termination.

**(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 full creative credits, the game should have either *two basic features or one advanced feature* outside the mandatory requirements.

Examples of basic features you should be able to complete include camera controls and simple rendering effects. Examples of advanced features include enemy group behavior and parallax scrolling backgrounds. Check out the **MilestoneSubmissionForm.pdf** for ideas of other creative tasks you can complete and classifications of basic vs advanced features (please make sure you have the background knowledge to implement chosen features).

Grading here will necessarily be subjective: more complex features or those better fitting into the overall game will be rewarded with more points.

**Note:** You will receive full credit for features only if they are **fully** operational. You will receive points for creative components only if the mandatory ones are fully operational. Points will be deducted for buggy and incomplete implementations.

### **Documentation:**

- Provide a README.md providing entry points to each of the implemented features and explain them where necessary.
- **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.
- **Game Design Documentation:** Document the ECS design pattern used in your game. Enumerate the game entities and actionable components used. Draw a diagram of the interaction between entities and components.
- Please submit a filled **MilestoneSubmissionForm.pdf** with this and all subsequent milestones.

**Submission:** Submit the code and associated documents using the course Git repository that has been set up for your team 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'.*