# CPSC 427 - Video Game Programming

# Winter 21/22

## Milestone 4: Final Game – April 8, 2022

The final version of your game should support robust and continuous gameplay as well as integrated advanced game elements. It should be self-contained – namely players should be able to play the game with no outside help or explanations. You should implement one or more additional advanced gameplay features (AI, physics, geometry, or other) and incorporate one or more advanced play features using either a game engine or alternative tools. The game should fully comply with your game development plan.

**(50%)** *Mandatory Requirements:*

- **Stability** (15%):
  - Include fully completed and playable prior-milestone implementations. Fix all bugs identified in prior marking sessions. *You will lose points if any bugs identified in prior marking sessions remain unfixed..*
  - The game code should support continuing execution and graceful termination, with no crashes, glitches, or other unpredictable behavior.
- **Playability** (15%): Sustain *progressive, non-repetitive* gameplay for at least **10 minutes.** During this time the player should be able to interact with the game and see **new** content for most of the time.
- **User Experience** (20%):
  - Include a tutorial introducing the player to the game mechanics. The game should be self-explanatory with **no verbal explanation required at any point during the gameplay** (10%).
  - Evaluate and optimize user-game interactions (choice of user gestures, ease of navigation, etc.). **Report** on the user testing you performed, including user feedback. Document the changes you implemented in response to the feedback (10%).

**(50%)** *Creative Components*: To obtain full marks you should implement a subset of the advanced features below. **Correctly** implementing multiple features can bring your grade above 100%.

- **External Integration** (10%): Include integration of one or more external tools or libraries: physical simulation (PhysX, Bullet, ODE, etc.), game engines, or alternatives.
- **Advanced Graphics** (20%): Implement an advanced graphics feature such as visual effects (Particle Systems, 2.5D(3D) lighting, 2D dynamic shadows), or advanced 2D geometric modifications (2D deformations, rigged/skinned motion).
- **Advanced Gameplay** (20%): Implement an advanced gameplay feature such as advanced decision-making mechanisms based on goals (path planning, A*, or similar), advanced group behavior (e.g. coordination between enemies), or more complex physical interactions with the environment (e.g. gravity, bouncing, complex dynamics). *To receive full marks, the physical effects implemented should be correctly integrated in time and should not be locked to the machine's speed by correctly handling the simulation time step and integration.*
- **Audio** (10%): Add audio feedback for all meaningful interactions in the game as well as background music with tones reflecting the journey of the game.
- **Other:** As an alternative to the above you can implement a selection of basic (10%) or advanced (20%) features listed in the **MilestoneSubmissionForm.pdf** which were not part of prior milestones.

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 any of the features above 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/or 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. *Highlight any changes versus the previous milestone.*
- 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'*