

# Programming Project

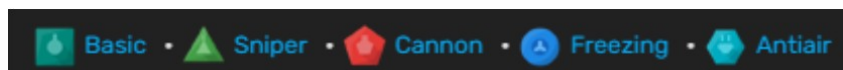
In this project, you will program, in pairs, your own light version of the game **Infini-  
todes 2** in modern C++ with SDL rendering.

The game is available for free on Steam, iOS, and Android if you want to test it.  
The features and properties are described in the wiki.

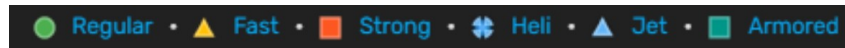


You will mainly have to implement:

- A real-time game
- A map with:
  - Start and end points
  - Path(s) connecting them
  - Locations reserved for towers
  - Loaded from an ASCII file
- Towers:
  - Basic, Sniper, Cannon, Freezing, Anti-air
  - With stats:
    - \* Range, Damage, Attack speed, Rotation speed, Projectile speed
  - Firing projectiles
  - That can be improved by paying



- Enemies:
  - Regular, Fast, Strong, Heli, Jet, Armored
  - With properties such as hit points, speed, resistance, and the fact of flying or not



Controls with keyboard shortcuts (for example, arrow keys to select a tile, 1 to 5 for towers, Enter to launch a wave, Space for play pause)

For points:

Tower experience and level

with boosts at levels 4, 7, 10, etc.

Tower rotation animations and enemy orientation

Mouse controls with GUI

Create an editor to create, modify, and save maps

Add modifiers placed near towers to provide bonuses.

Do not worry about the research trees, menus, resources, or persistence/continuity between game sessions.

The objective of this project is to evaluate your code and your application according to these criteria, not your ability to reproduce the original exactly. Whether or not you are inspired by the graphic style, stats, or gameplay, you are free as long as you respect the spirit of the criteria.

You may replace SDL with another graphics library if you wish.

The evaluation will consider the content, but above all the modernity, cleanliness, and organization of the code.

You must follow these criteria:

Use the STL as much as possible.

No arrays! Use STL containers.

No raw pointers! Use unique/shared/weak\_ptr.

Use STL algorithms.

Create classes for your objects and encapsulate SDL2.

Use C++ 11/14/17/20/etc. features (Lambda, range-for).

Use inheritance and polymorphism.

Be const-correct and efficient; no unnecessary copies!

Document your code.

Use a coding convention.

A quick reminder regarding generative AI:

Generated code can be easily recognized.

The purpose of this project is to evaluate your C++ skills, not your ability to ask Copilot or GPT to do your homework.

Using AI to help explain documentation is acceptable and useful.

Using AI to do your project is not.

Using AI to generate your code or to “fix” your code is therefore strictly forbidden.