Towerfull

Authors :

* Thomas Vuilleumier
* Sebastian Diaz
* Lionel Pollien

Une image contenant bâtiment, maison, pixel, tour

Le contenu généré par l’IA peut être incorrect.

Source : [Isometric tower defense game tower sprite on Craiyon](https://www.craiyon.com/image/TE1RQ3J6R-a4ZzwBaKe8hA)

# Project context

During the HEIG-VD’s MCR course, we are asked to create a project using a certain prototype, one per group. Our model is Prototype, and aims to create instances of a class by cloning another instance. To exploit this, we chose to create a tower defense we chose to call Towerfull.

The use of Prototype in our project resides in two main parts:

* The saves: in the rest of the report, you’ll hear about the GameArea. It’s the center of our game, it’s the conceptual board, and it’s a Prototype. We clone the gameArea, storing it as a save, which we can load later. As such, these aren’t copies that are saved to the disk.
* The factories: our enemies and protections are standardized, and as such we store their instances in factories and clone them when we need them.

This project was made using the library libGDX, coming from a recommendation from the teacher.

# Launching the project

**Clone the repository**

$ git clone [git@github.com:Cobora2001/Towerfull.git](mailto:git@github.com:Cobora2001/Towerfull.git)

**Go in the folder**

$ cd Towerfull

**Launch the game**

From an IDE:

- Open *./lwjgl3/src/main/java/io.github.towerfull.lwjgl3/Lwjgl3Launcher* in your IDE

- From there, there is a **public static void main(String[] args)** method that you can use. This will launch the game.

From the command line:

- Launch the command: $ ./gradlew lwjgl3:run

- If needed, you can do a clean build: ./gradlew

# UML

To be fixed with the current one

Une image contenant capture d’écran, texte, Rectangle, carré

Le contenu généré par l’IA peut être incorrect.