# C++ Final Project Proposal

*Denis-Patrick Odagiu*

denis-patrick.odagiu@student.manchester.ac.uk | github.com/bb511

**Text Based Adventure Game**

## Brief Description

The project will be a short action adventure game in which the player must escape a prison. At the start of the game, the main (playable) character is spawned inside a cell. Then, one out of three possible random events occur, making the escape possible. Each such event sets the player on a distinct path for the rest of the game, where he must face different challenges to escape. All paths converge at the end of the game into a final battle. However, the paths might intertwine at various points during the game.

In his attempts to escape, the player can interact with different objects which might aid or hinder him. For example, the player might start with a rusty spoon in his inventory, which he may use in a number of ways, such as causing noise by striking it against the cell bars or silently starting to dig an ambitious escape tunnel. Current choices will influence what kind of challenges and choices the player will have to face next. Therefore, the game will be relatively short, but very replayable.

## Used C++ Features

By default, the game will use a main character class. Furthermore, each object will be part of a class hierarchy, depending on its use. For example, the spoon might inherit from a class called 'kitchenUtensils' which inherits from an abstract base class named 'escapeTools'. The game will also use random number generators (for the starting scenario), error detection/exception handling (to prohibit restricted player actions), lambda functions, templates (to improve code readability and functionality) and dynamic memory allocation (player inventory). The output might contain some ascii art while the input will determine character actions in a simple but interesting way.

## Minimal Layout

Minimum class design specification:

- Abstract base classes for escape items, utility items and weapons. Derived classes for each item present in the game. Pure virtual functions that allow the basic moves with a certain type of item.
- Abstract character class with pure virtual function to allow basic movement. Inventory class with dynamic storage.
- Player class that inherits from both. Final boss class derived from character class.

Minimum functionality:

- Allowed actions for each item and the player.
- Input function that allows the player to make choices. Output function that shows inventory, what each item does and that displays current possible options to the player.