# COMP 62542
# Pattern-Based Software Development

**Group 5** Aurélie Pallas(9839067), Bing Xu (9649124), Carlos Valarezo (9312392), Dingxin Yu (9763008), Olivier Staub (9835426), Salim Salim (9520050).

## Application: Space Invaders Game

We decided to implement a customised (Star War themed) version of the famous Space Invaders game. Our version contains most of the original features. We implemented multiple levels which challenges the player with different enemies.

## Patterns

**Singleton pattern** We only need one instance of the factory throughout the game. We achieve this by using singleton to make sure only one instance of factory is used anytime needed.

**Abstract factory pattern** A factory help us separate the object creation from their implementation and client. In our design, all object creation are handled by a concrete factory class.

**Composite pattern** Composition helps the game treat a collection of objects as single. In each level there is a game board which contains player, enemies and bullets instances. The game class needs only to interact with the composite instance of a game board in order to control those objects.

**Flyweight pattern** Images used by enemy and bullet should be loaded only once and shared between objects that uses them. Since the game has many enemy objects and bullets of the same type, loading the shared part (image) only once saves memory usage for storing enemies and bullets.

**State pattern** The game behaviour depends on its current state, each phase of the game is a state and each state implements its own behaviour. Our game contains six different states (initialization, play, pause, level up, game over and finish). The transition between the states occurs according to a certain keystroke or event (for instance when player is killed or when all enemies have been shot).

**Strategy pattern** We use strategy to encapsulate different movement algorithms into different classes. Then use dynamic binding during Entity creation to assign an instance a movement strategy. Entities have different types of movement. A, player depends on key press to move and enemies move differently in each level.