Introduzione al game-development in **C++11/C++14**

Vittorio Romeo
http://vittorioromeo.info
vittorio.romeo@outlook.com

Chi sono?

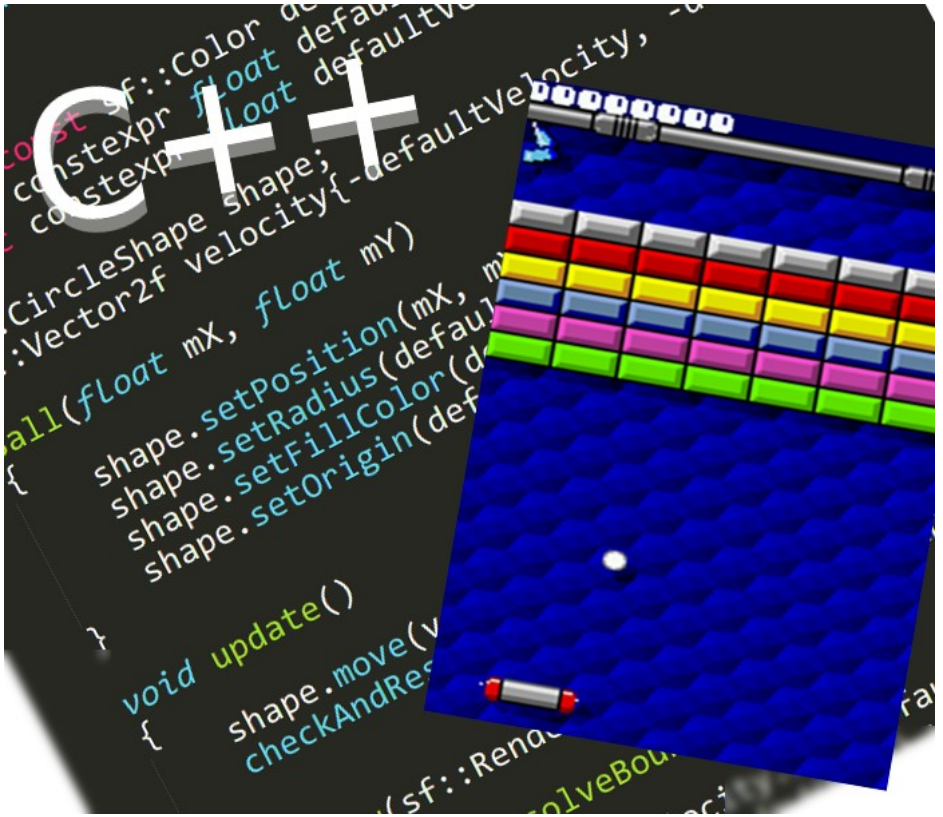
- Studente di **Scienze Informatiche** all'**Università di Messina**
- Programmatore **autodidatta**
- Interessi:
  - **Software development**
  - **Gaming** e **game-development**
  - **C++** e la sua evoluzione
  - Software **open-source**
  - **Condividere** le mie conoscenze

++it Italian C++ Community
meetup
20 Giugno 2015, Firenze

Vittorio Romeo
http://vittorioromeo.info
vittorio.romeo@outlook.com

La **presentazione** di oggi

- **Parte introduttiva:**
  - Game development: **perchè?**
    - **Perchè C++?**
    - **Perchè C++11/C++14?**
- **Parte “live coding”:**
  - Preparazione: **obiettivi, compilatori, risorse**
  - **Live coding**: **analisi dello sviluppo** di un semplice gioco completo

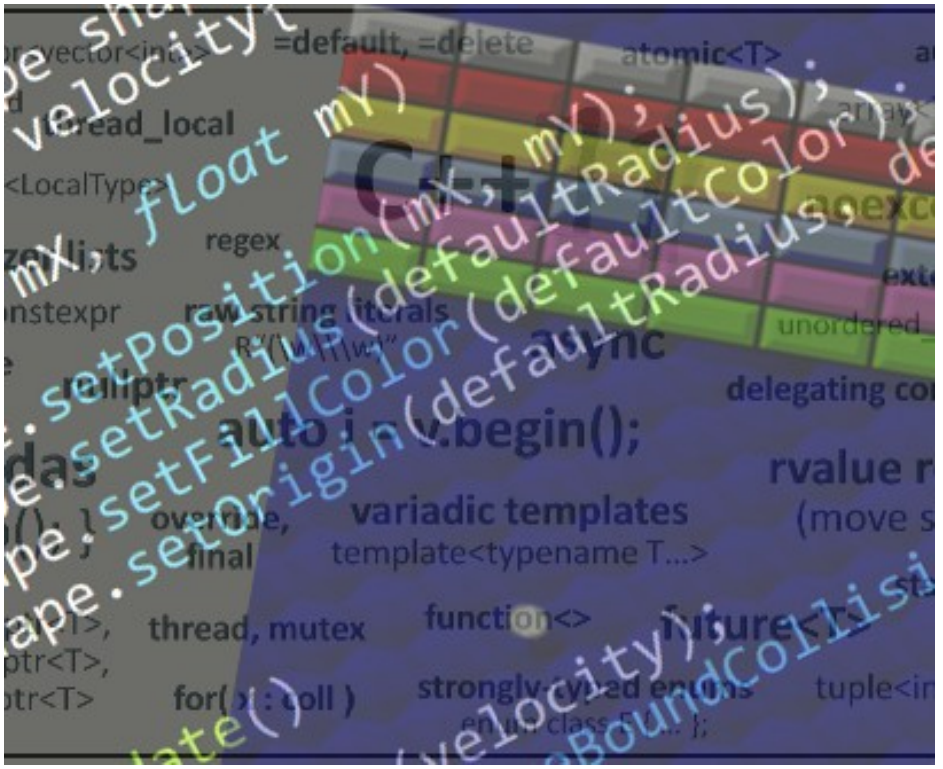


++it Italian C++ Community
meetup
20 Giugno 2015, Firenze

Vittorio Romeo
http://vittorioromeo.info
vittorio.romeo@outlook.com

Obiettivi di oggi

- **Incoraggiare** chiunque a provare il **game development**
- **Dimostrare** quanto i nuovi standard del **C++** semplifichino il processo di **development**



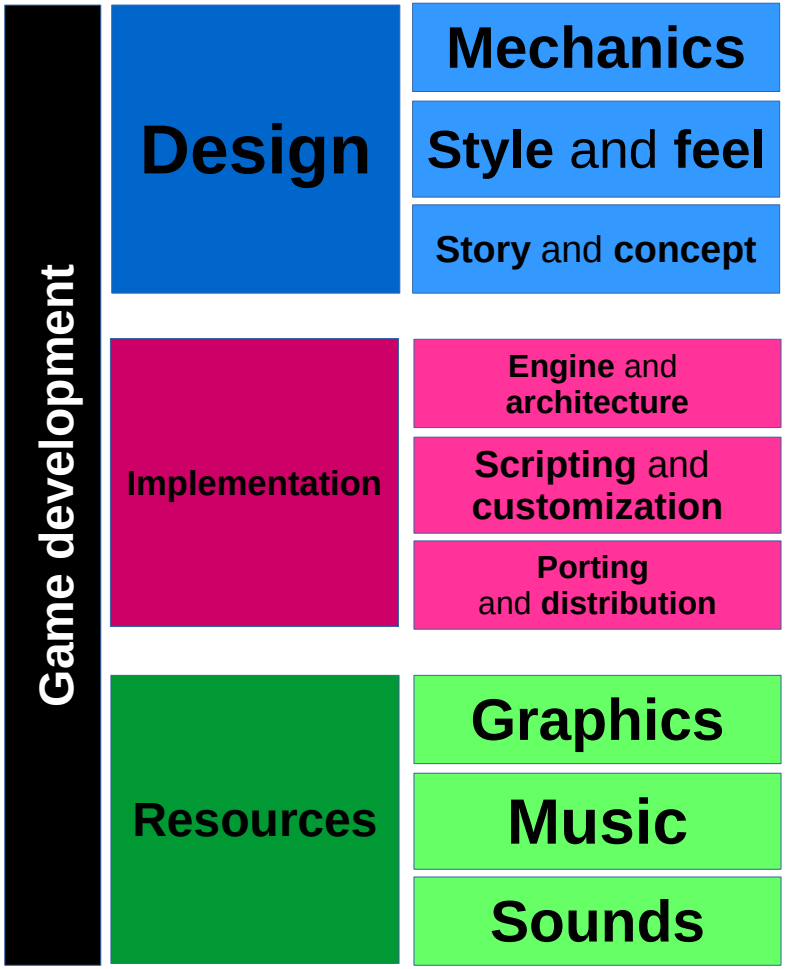
Vittorio Romeo
http://vittorioromeo.info
vittorio.romeo@outlook.com

++it Italian C++ Community
Meetup
20 Giugno 2015, Firenze

Game development:
esperienza a 360°

Game development: **perchè?**

- Game development
  - Richiede **conoscenze** e **abilità** in diverse aree
  - **Coinvolge** il programmatore con la **community**
  - **Tocca** un **vasto numero** di specifici **argomenti di sviluppo**



Vittorio Romeo
http://vittorioromeo.info
vittorio.romeo@outlook.com

Game development: **perchè C++?**

- **Efficienza**: *zero-cost abstractions* e codice *“low-level”*
- **Portabilità**: il codice *standard-compliant* funziona su molte architteture
- **Popolarità**: enorme numero di *librerie* e *risorse*

Vittorio Romeo
http://vittorioromeo.info
vittorio.romeo@outlook.com

Game development: **perché C++11/C++14?**

- **Convenienza**, **sicurezza** ed **espressività**
- **Initializer lists** e **uniform initialization**
  - `auto`, **range-based for loops**
  - **Lambdas**, **variadic templates**, `decltype`
  - `override`, `final`, `enum class`, `explicit`, `nullptr`
  - `default`, `delete`
- **Memory management** (**!**)
  - `std::unique_ptr`, `std::shared_ptr`, `offsetof`
- Possibili miglioramenti della **performance**
  - `constexpr`, `std::move`, `noexcept`
- Altre **migliorie/aggiunte**
  - **Multithreading** standardizzato
  - `std::tuple`, **variadic macros**, `<random>`, `<chrono>`
  - **Generic lambdas**, **lambda capture expression**
  - `auto` functions, **relaxed constexpr**, `std::tuple::get<...>`

factory functions & callbacks

entity management

game loop timing

++it Italian C++ Community meetup 20 Giugno 2015, Firenze

Vittorio Romeo
http://vittorioromeo.info
vittorio.romeo@outlook.com

Iniziamo lo sviluppo!

Live coding: **il nostro obiettivo**

- Il nostro obiettivo è **creare** un clone di **arkanoid/breakout** quasi completamente da zero.
- **Passo passo**, dimostrerò quanto è facile creare un gioco completo in poco più di **200 linee di codice.**

Vittorio Romeo
http://vittorioromeo.info
vittorio.romeo@outlook.com

Live coding: **compilatori**

- Il supporto **C++11** è necessario
  - `g++ 4.7.2` o `clang++ 3.0` supportano completamente lo **standard C++11**
  - Qualche feature del **C++14** sarà utilizzata
    - `clang++ 3.4` supporta completamente lo **standard C++14**
    - `g++ 4.9` supporta tutte le features del **C++14** che useremo
- Informazioni sul supporto **C++14** dei compilatori
  - `http://gcc.gnu.org/projects/cxx1y.html`
  - `http://clang.llvm.org/cxx_status.html`

Vittorio Romeo
http://vittorioromeo.info
vittorio.romeo@outlook.com

Live coding: **libreria SFML**

- Per **interfacciarsi** con le componenti **input**, **audio**, **grafiche** del nostro PC e del nostro sistema operativo useremo la libreria open-source **SFML 2.x** disponbili qui: http://sfml-dev.org

Vittorio Romeo
http://vittorioromeo.info
vittorio.romeo@outlook.com

Live coding: **codice** e **risorse**

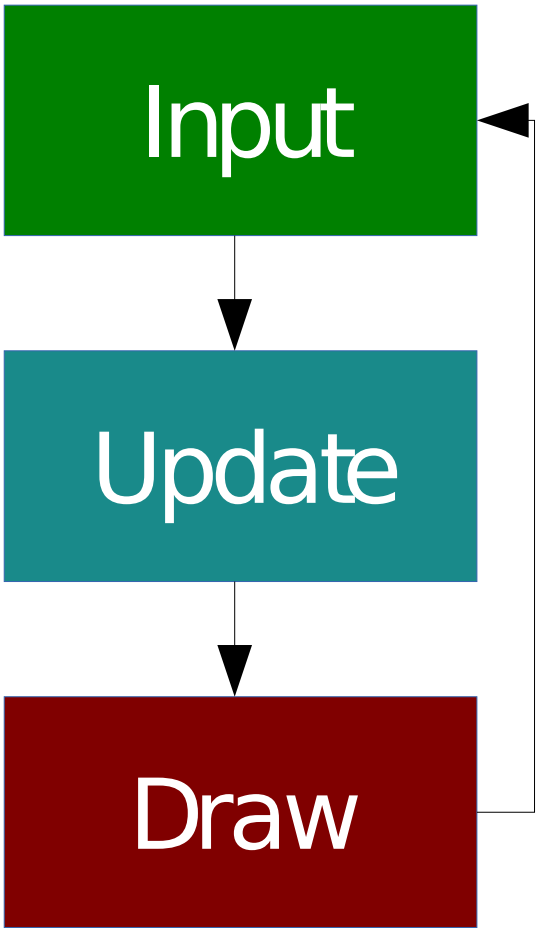
- Il **codice** e le **risorse** che useremo sono disponbili su questo **repository**:
  - `http://github.com/SuperV1234/itcpp2015`

- La libreria **SFML 2.x** è disponibile qui:
  - `http://sfml-dev.org`

Vittorio Romeo
http://vittorioromeo.info
vittorio.romeo@outlook.com

Info: **game loop**

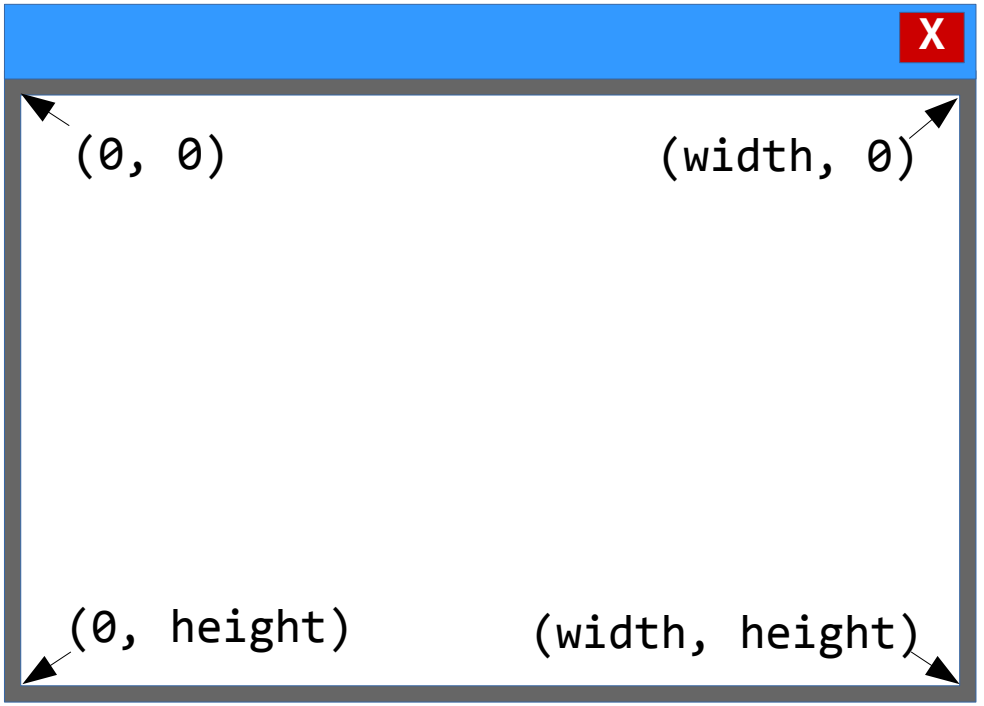
- Il **game loop** è un loop che si ripete finchè il gioco non termina
  - 1. Ottieni **input**
  - 2. **Aggiorna** logica
  - 3. **Disegna** entità



Vittorio Romeo
http://vittorioromeo.info
vittorio.romeo@outlook.com

Info: **sistema** di **coordinate**

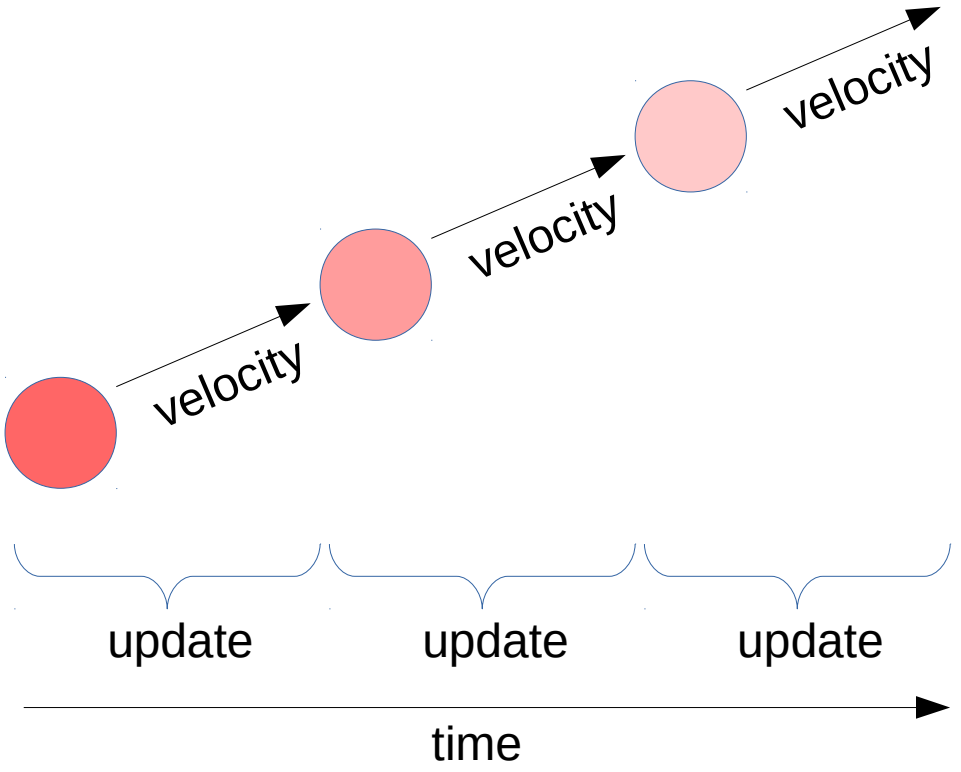
- Il **sistema** di **coordinate SFML** setta l'origine nell'angolo in alto a sinistra della finestra



Vittorio Romeo
http://vittorioromeo.info
vittorio.romeo@outlook.com

Info: **movimento** della **pallina**

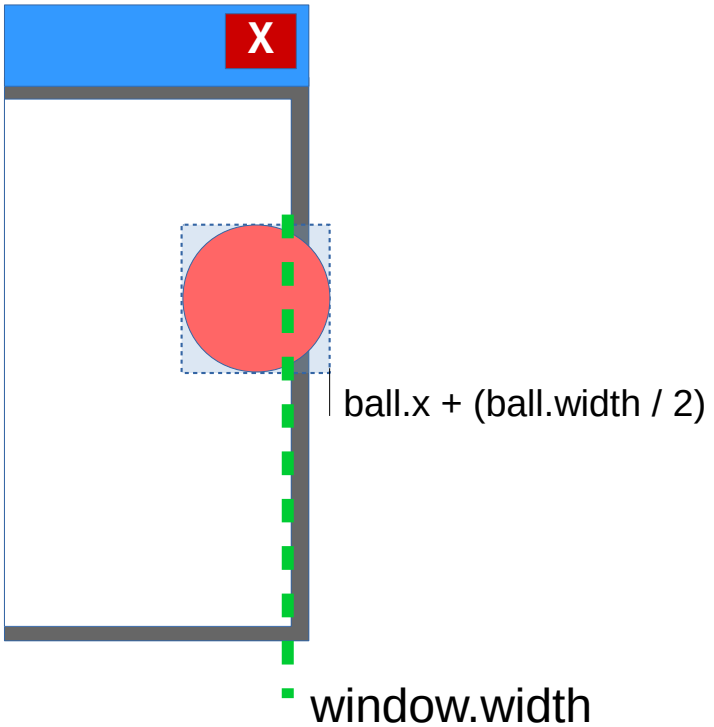
- Sommando il **vettore velocità** della pallina al suo **vettore posizione ogni frame**, essa si muoverà



Vittorio Romeo
http://vittorioromeo.info
vittorio.romeo@outlook.com

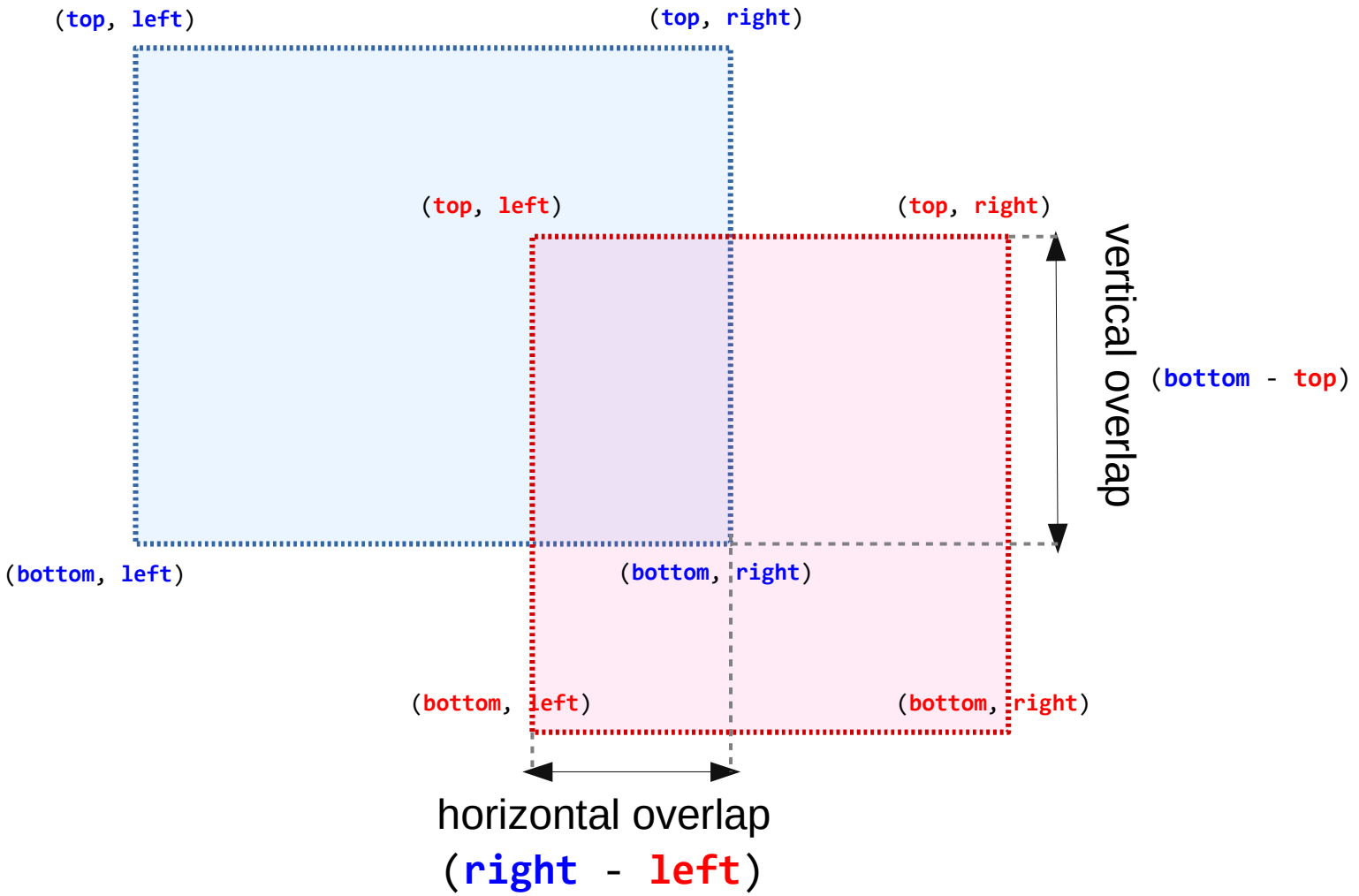
Info: collisione **pallina vs finestra**

- Controllando se una delle coordinate della pallina è **maggiore o inferiore** ai confini della finestra, possiamo determinare se essa è uscita fuori.



Vittorio Romeo
http://vittorioromeo.info
vittorio.romeo@outlook.com

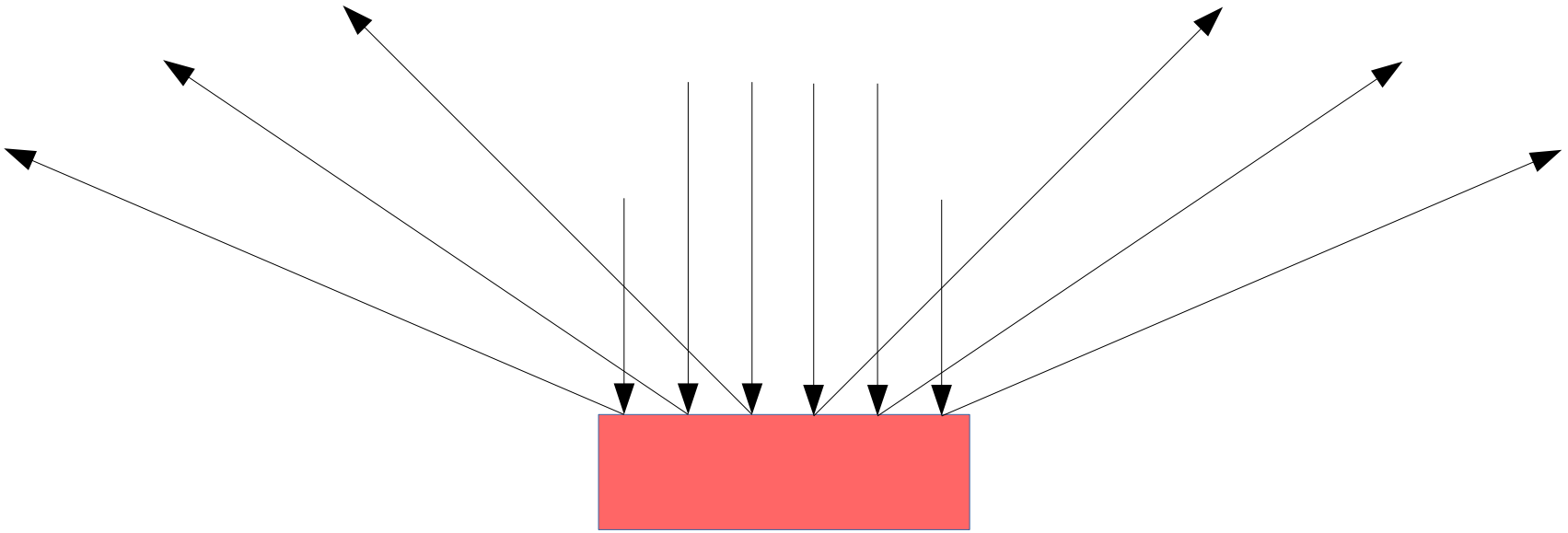
Info: collisione **AABB vs AABB**



Vittorio Romeo
http://vittorioromeo.info
vittorio.romeo@outlook.com

Info: collisione **pallina vs paddle**

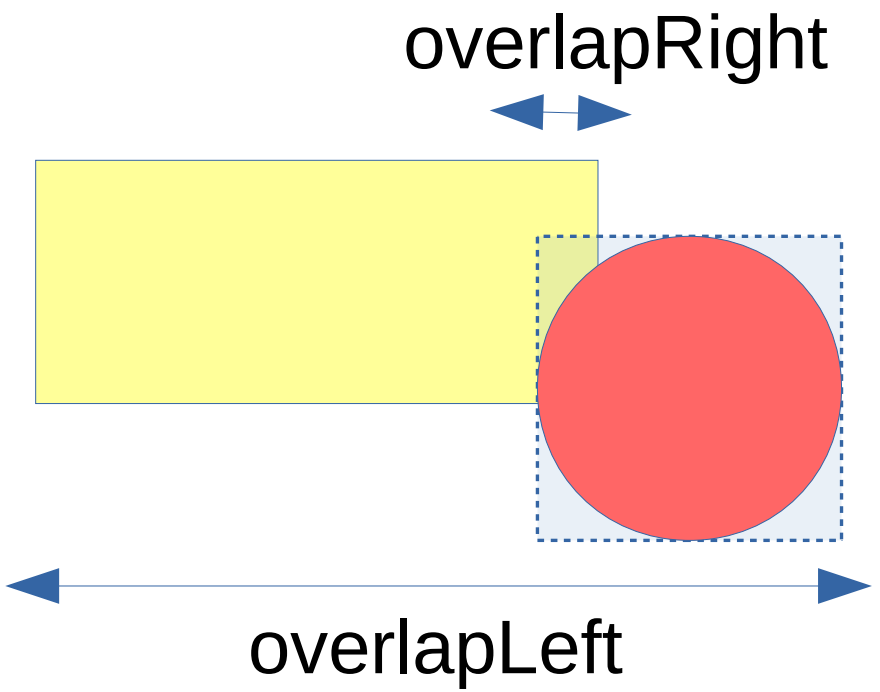
- In base alla **posizione colpita** ed alla **velocità del paddle**, la pallina verrà riflessa come in figura.



Vittorio Romeo
http://vittorioromeo.info
vittorio.romeo@outlook.com

Info: collisione **ball vs brick**

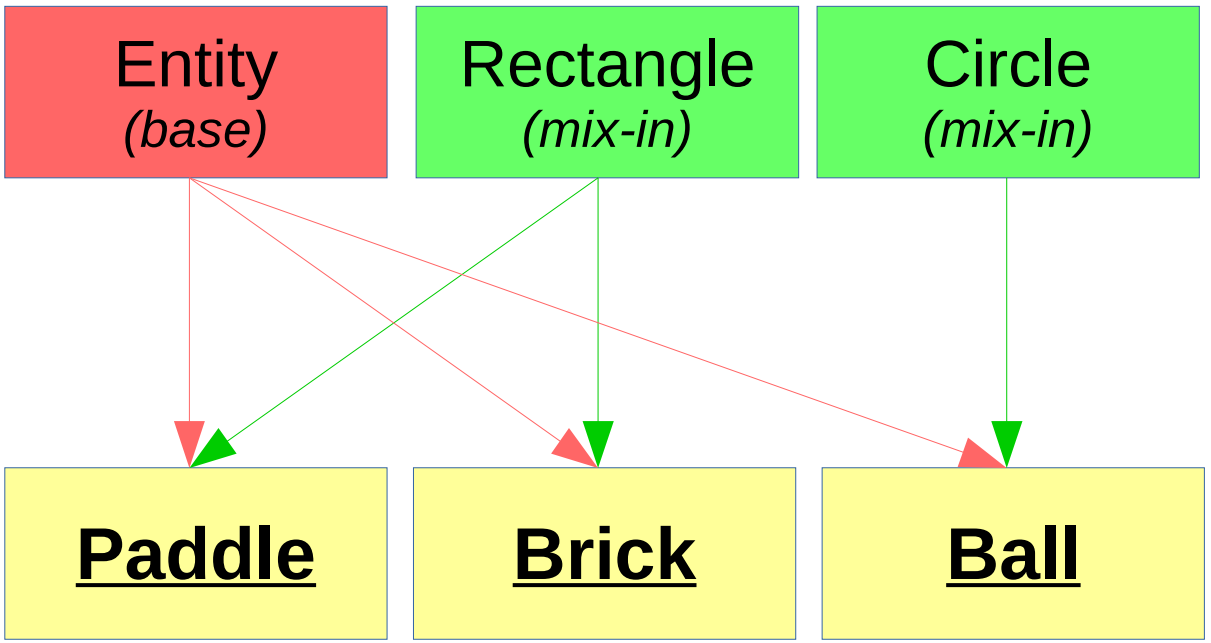
- Dobbiamo capire **la direzione** dalla quale il mattoncino è stato colpito.



Vittorio Romeo
http://vittorioromeo.info
vittorio.romeo@outlook.com

Info: **gerarchia classi**

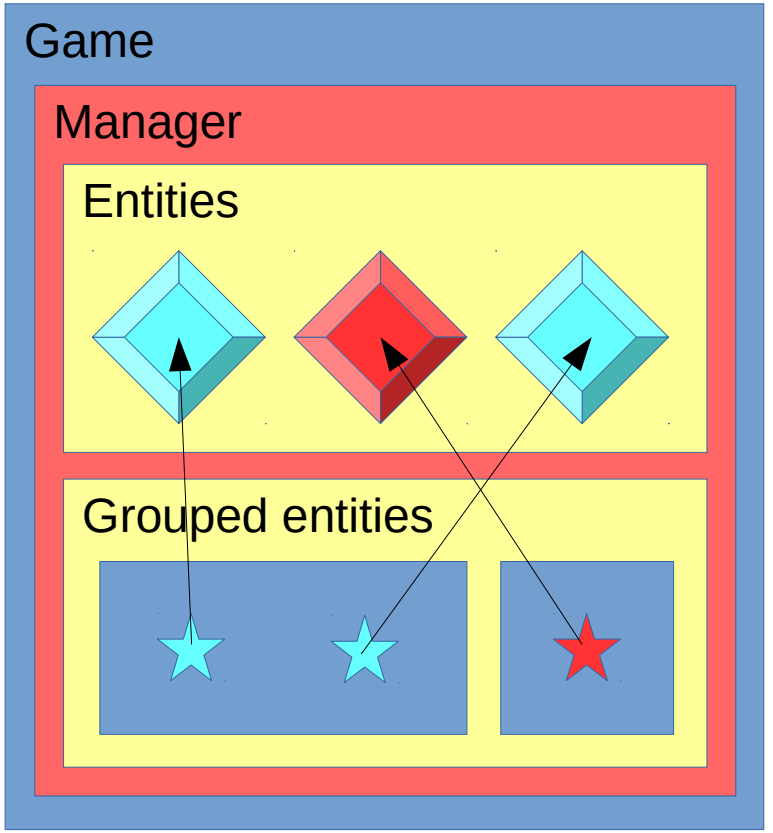
- Questa è la **gerarchia finale** dei nostri elementi di gioco.



Vittorio Romeo
http://vittorioromeo.info
vittorio.romeo@outlook.com

Info: **architettura** delle **entità**

- Questa è l'architettura finale del nostro **entity manager**.



Vittorio Romeo
http://vittorioromeo.info
vittorio.romeo@outlook.com

Grazie!

Vittorio Romeo
http://vittorioromeo.info
vittorio.romeo@outlook.com