

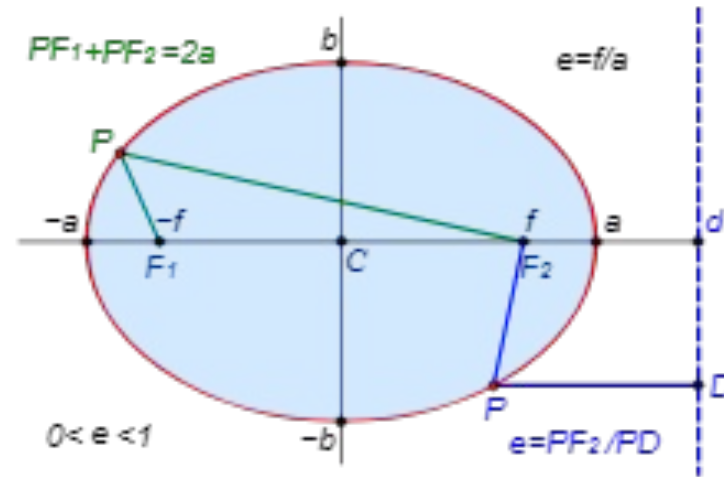
Esercizi su classi e oggetti

Facciamo un po' di pratica



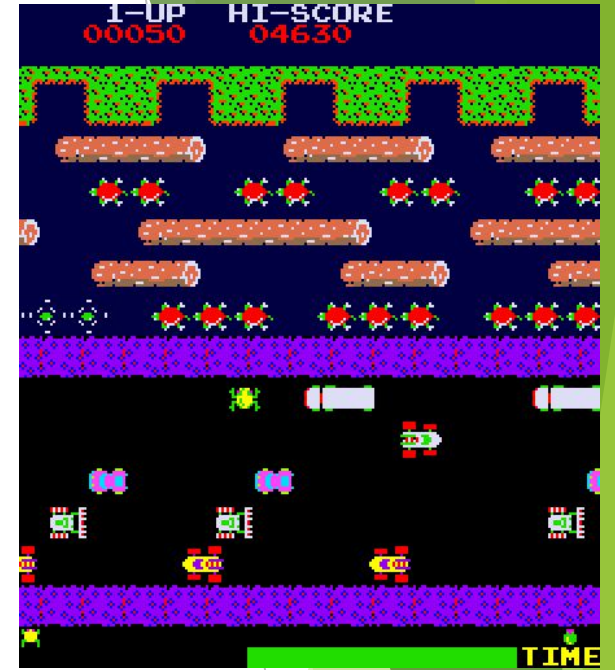
Classe per l'ellisse

- Classe che modella un'ellisse
- Campi privati (parametri del costruttore)
 - Semiassi: a , b
- Metodi pubblici per ottenere...
 - Area: $\pi \cdot a \cdot b$
 - Distanza focale: $2 \cdot \sqrt{|a^2 - b^2|}$
- Nel corpo principale del programma...
 - Creare un oggetto con dati forniti dall'utente
 - Visualizzare area e distanza focale dell'ellisse



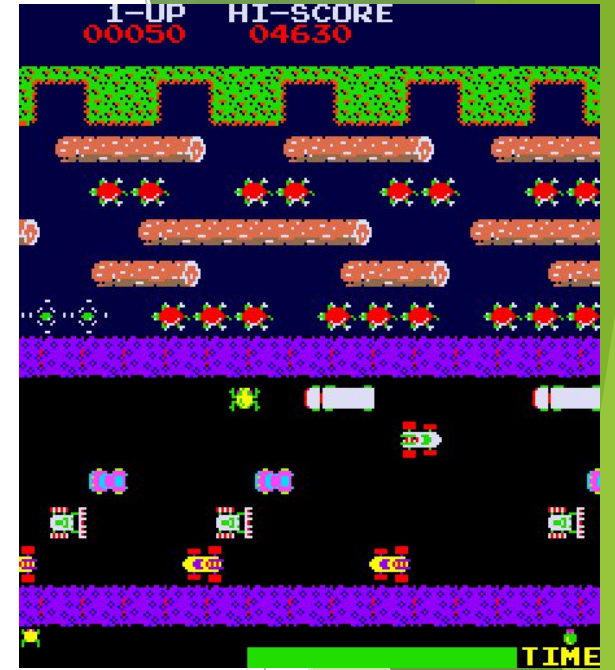
Animazione di un veicolo

- Creare un veicolo modificando la classe `Ball` vista a lezione
- Eseguire l'animazione
 - Per ogni frame, chiamare il metodo `move` del veicolo
 - Rappresentare un rettangolo nella posizione aggiornata del veicolo
- Modificare però il metodo `move`
 - Il veicolo si sposta sempre di pochi pixel, solo in orizzontale
 - Se supera 100px oltre il bordo destro, ricompare a 100px prima del bordo sinistro e viceversa



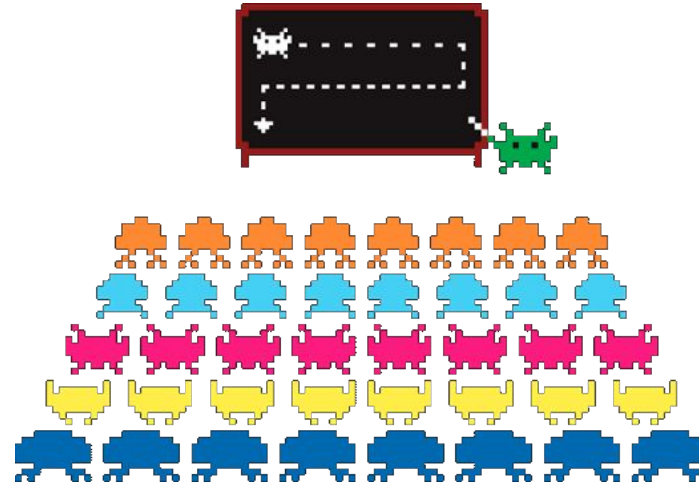
Pallina con conteggio

- Partire di nuovo dalla classe `Ball` vista a lezione
- La pallina si muove solo dopo il click del mouse
- Esegue i suoi comportamenti già definiti, ma solo per 10 fotogrammi
- Dopo 10 fotogrammi si ferma, in attesa di una nuova pressione del tasto



Animazione di un alieno

- Partire di nuovo dalla classe `Ball` vista a lezione
- Il movimento di base però è solo orizzontale
- Arrivato al bordo, il personaggio:
 - a. Si sposta di qualche pixel verso il basso
 - b. Poi cambia direzione orizzontale
- Fare in modo che, in ogni frame, lo spostamento sia solo orizzontale, o solo verticale, ma *non* diagonale



Spirale ad oggetti

- Mostrare l'*animazione* di un cerchio lungo una spirale
- Realizzare una classe per gestire dati e comportamento del cerchio
- Implementare il movimento in un metodo `move()`
- Campi: `xc`, `yc`, `i`
 - `i` conta i passi; se eccede il limite, torna a 0

