Day6: Greenfoot Skeleton for Platform game with image manipulation

Pulling it all together!
Review

- Game concepts
- Movement
  - Position, velocity, acceleration
  - Direction, image substitution
- Object interaction
- Distance, Find closest prey
- Levels
- Score, Health, Timer
- Images and sound
- GameOver
- Image Manipulation
  - Color, brightness, size
  - Combining, cropping
Game-Starter Scenario

• Let’s look at the class hierarchy

• Remember for **Picture** we can follow the link off Techgirls page:
  • [Picture JavaDocs](#)

• **MyPicture** has the transforms you have been working on

• Notice **Mover**’s subclasses are **Gateway, Sprite** and **SceneryObject**
Let’s run it!

- What do you notice?
- How does the camel land on the next level?
- How does the camel get caught by the platform?
- How does the jumping work?
- How does the ladder work?
Let’s look at the code!

What does a **Mover** do?
- has location, velocity and acceleration, so there are setters and getters for the x and y version for all of that
- stops
- fall and stop falling

- **What does a Sprite do?**
  - inherits methods from **Mover**
  - hover
  - overrides `setLocation(...)` so it can land on sceneryObjects

- **What does a SceneryObject do?**
  - inherits methods from **Mover**
  - checks if a mover is on top of it
  - overrides `setLocation(...)` so it can bring other movers along
Let’s look at the code!

• What about other classes?
• Which methods did we see when we ran it?
• Add some System.out.println() statements
• What else could we do in this game?
Play around...

- Write an endGame method in **Player**
- Rearrange the scenery objects: **Ladder, Ground and Platform**
- Add charms
- Add sound effects
- Keep score