CS 210 - Week 04 Game Activity - Inheritance and Polymorphism

- Working Individually but can help each other
- Download the “ZombieGame.zip”
- Open empty folder in VS Studio Code (it is a VS Studio Project)
- Review the code / Run the code (watch instructions first)
- **Have fun, take a look - not graded!**

- ***Optional** - Create a character for yourself (to replace Ellie.java). Be sure to update the instance properties to match your new character. Instantiate your custom Character.
- Make Player abstract. We will only instantiate a subclass (Ellie or your own custom class).
- Add 2-4 new Zombie **subclasses** (example FungalZombie)
    - Give them a different draw character (other than Z which is the base’s)
    - Give them a different attack message with their specific attack.
    - Fulfill the interface required methods
    - Add a means to change the attack damage in your new zombie (the default for Zombie was 20)
- Make Zombie “abstract” and only use your new Zombie subclasses
- Add a new **interface** “IAttacker.java” or “Attacker.java”
    - Add the Attack(ILiving livingEntity) “WHAT” method to the interface
    - Implement the IAttacker interface in the Zombie base class. The Attack(ILiving livingEntity) is already defined.
    - Make sure the game still runs with this change (we are just defining an Attack ability in the interface to use the interface later)
- Add a new “Spike.java” **class** to represent another obstacle in the game
    - It will look more like Exit.java than Zombie.java, however it will implement the IAttacker
    - Implement the IAttacker interface and fulfill requirements
        - Adjust the damage per attack on the spike (any number, currently the player’s health is 100)
    - Give it a symbol / string on the map ^ (caret?)
    - Add code to instantiate a random number of spikes (they don’t move) and add them to the world. (see addZombies() for an example to copy/paste and change to Spike)
    - Test your code at this point to confirm the Spikes in the world.
    - Update the collision handling to **not be specific to instanceof Zombie** but switch to IAttacker and let the Spike OR Zombie perform their attack.
    - Confirm the collision works when the Player hits the Spike to reduce health (just like Zombie attack)

Demonstrate

- Inheritance
  - Show the Zombie subclasses and their overrides. Describe their custom attacks and different instance variables.
- Abstract
  - Show Player is abstract, state why and what class(es) extends it?
  - Show how Zombie is abstract and requires a subclass to be instantiated (new) in the world.
  - Show how the IAttacker interface allows 2 different classes to utilize a method definition where “HOW” implementation differs.
- Interfaces
  - Discuss the IAttacker interface and its purpose (related / similar to question above).
  - Discuss the ILiving interface and its purpose.

Discussion

- What could make the game better?
  - Could you add items? Health restore? Speed boost (move 2 spaces)?
  - Could the movements be more strategic? Move in the direction of the player? Only start moving within a certain distance from the player?
- Could there be levels?
  - What are some ideas for other levels?
- Where else could interfaces be used?
- What could be more random/less random?
- Would a bigger map/world be possible?
- Can you see how potentially the “model” could be separated from the “graphics” and you could upgrade this to 2D sprites? The model is a sort of simulation and can occur without the “draw” or graphics.
- Are there too many Zombie games? 🙂