Implement collision handling for EVERY types of collisions

- **Types of collisions**
  - Link and walls
  - Link and blocks
  - Link and NPCs (Link bounces back if he touches an enemy)
  - Link and projectiles
  - Link and items (items should disappear when Link picks them up)
  - NPCs and walls
  - NPCs and blocks
  - NPCs and NPCs?
  - NPCs and projectiles
  - Walls and projectiles
- **Suggestions on implementing collision and handling**
  - https://gamedevelopment.tutsplus.com/tutorials/quick-tip-use-quadtrees-to-detect-likely-collisions-in-2d-space--gamedev-374
    - Using quadtrees to implement collision detection.
    - This might be beyond the scope of our project though and should only be considered if we have a lot of time.
    - This method is especially useful if the game is displaying a lot of objects at once where each one requires collision detection.
  - **For our purposes, read below:**
    - Bound each object inside a rectangle and check for collision when two rectangles intersect. This is known as AABB collision detection.
    - Each collidable object will keep track of their size as well as their current position. This will give us a rectangle to use for the purpose of handling collisions.
    - CollisionDetector class that will have a method to check for collisions every time the game updates.
    - Each collidable object can implement a CollisionResolution interface with a method to resolve the different types of collisions shown above.

Create and implement ALL of the rooms in the first dungeon

- Store room information in xml files
  - http://web.cse.ohio-state.edu/~boggus.2/3902/gameframework.html
  - Look under File I/O options
- Have a class called LoadRoom() that is able to read the xml file and load everything from it.
- Each xml file should load the initial state for the room that they are implementing.
- For the scope of this sprint, implement an easy way to scroll through all the rooms for the purpose of testing. This can be done by using key presses.

Continue to implement Link using items

- Bow
- Bombs
- Sword Beam
- Magic staff
- Compass
- Map them to 1, 2, 3, … keys like in Sprint 2

Create Artificial Level

- Include an instance of Link.
- Include all types of objects found within the first dungeon.
- Similar to Sprint 2.