# Sample Program Files

All sample .next programs are followed by their Next.java files they translate to.

```
/*LRMSample.next by Ernesto Arreguin */

int count;

item the_greatest_sword_ever {(int damage = 100000000)}

character xiaowei_the_greatest_man_ever {(int life = 100000000,
int level=99999, string haha="hahahahaha"), (the_greatest_sword_ever)}

location where_is_this_place {(int sizex = 10000, int sizey=9283), (),
(xiaowei_the_greatest_man_ever)}

start where_is_this_place end (xiaowei_the_greatest_man_ever.life < 0) {
      choose (attack,"hia!","a") (up,"up","u") (fin, "end", "f") {
            when attack
            {
                  xiaowei_the_greatest_man_ever.life+1;

            } next where_is_this_place

            when up
            {
                  [? prob 40 {
                        count = count-1;
                        output "count:";
                        output count;
                  }
                  prob 60 {
                        count = count+1;
                        output "count:";
                        output count;
                  }
                  ?]
            } next where_is_this_place

            when fin
            {
                  xiaowei_the_greatest_man_ever.life = -1;
            } next where_is_this_place


      }
}
```

```
//Generated Next.java file for LRMSample.Next
import java.util.*;

public class Next {
   enum Type {INT, STRING, CHARACTER, ITEM, LOCATION}

   static Random r = new Random();
   Object dummy;
   String currentLocation;
   Map<String, Location> locations = new HashMap<String, Location>();
   Map<String, Character> characters = new HashMap<String, Character>();
   Map<String, Item> items = new HashMap<String, Item>();
   Map<String, Type> types = new HashMap<String, Type>();

   public static void main(String[] args) {
    (new Next()).play();
   }

   public int boolToInt(boolean value) {
        if(value) {
            return 1;
        }
        else {
            return 0;
        }
   }

   public String entitySetString(String key1, Type type1, String key2, String value) {
    boolean valueSet = false;
      if(type1 == Type.LOCATION) {
            Location loc = locations.get(key1);
            if(loc != null) {
                  loc.strAttrs.put(key2, value);
                  valueSet = true;
            }
      }
      else if(type1 == Type.CHARACTER) {
            Character character = characters.get(key1);
            if(character != null) {
                  character.strAttrs.put(key2, value);
                  valueSet = true;
            }
      }
      else if(type1 == Type.ITEM) {
            Item item = items.get(key1);
            if(item != null) {
                  item.strAttrs.put(key2, value);
                  valueSet = true;
            }
      }

      if(!valueSet) {
            throw new RuntimeException();
      }

      return value;
   }

   public int entitySetInt(String key1, Type type1, String key2, int value) {
      boolean foundReturnValue = false;

      if(type1 == Type.LOCATION) {
            Location loc = locations.get(key1);
            if(loc != null) {
            loc.intAttrs.put(key2, value);
                  foundReturnValue = true;
            }
      }
      else if(type1 == Type.CHARACTER) {
            Character character = characters.get(key1);
            if(character != null) {
                  character.intAttrs.put(key2, value);
```

```
                foundReturnValue = true;
          }
    }
    else if(type1 == Type.ITEM) {
          Item item = items.get(key1);
          if(item != null) {
                item.intAttrs.put(key2, value);
                foundReturnValue = true;
          }
    }

    if(foundReturnValue == false) {
          throw new RuntimeException();
    }

    return value;
}

    public boolean isTrue(Object object) {
        if(object instanceof String) {
           if(((String)object).isEmpty()) {
               return false;
           }
        }
        else if(object instanceof Integer) {
           if((Integer)object == 0) {
               return false;
           }
        }
        else {
            if(object == null) {
                return false;
            }
        }

        return true;
    }

  public void killFunction(String varName){
    if (characters.containsKey(varName)){
          characters.remove(varName);
          for (String key: locations.keySet()){
                      locations.get(key).hideCharacter(varName);
          }
    }
    else if (items.containsKey(varName)){
          items.remove(varName);
          for (String key:locations.keySet()){
                locations.get(key).removeItem(varName);
                }
          for (String key : characters.keySet()){
                characters.get(key).removeItem(varName);
          }
    }
  }

  public int entityHasInt(String key1, Type type1, String key2) {
    int returnValue = 0;
    boolean foundReturnValue = false;

    if(type1 == Type.LOCATION) {
          Location loc = locations.get(key1);
          if(loc != null) {
                returnValue = loc.intAttrs.get(key2);
                foundReturnValue = true;
          }
    }
    else if(type1 == Type.CHARACTER) {
          Character character = characters.get(key1);
          if(character != null) {
                returnValue = character.intAttrs.get(key2);
                foundReturnValue = true;
```

```
		}
	}
	else if(type1 == Type.ITEM) {
		Item item = items.get(key1);
		if(item != null) {
			returnValue = item.intAttrs.get(key2);
			foundReturnValue = true;
		}
	}

	if(foundReturnValue == false) {
		throw new RuntimeException();
	}

	return returnValue;
}

public String entityHasString(String key1, Type type1, String key2) {
	String returnValue = null;
	if(type1 == Type.LOCATION) {
		Location loc = locations.get(key1);
		if(loc != null) {
			returnValue = loc.strAttrs.get(key2);
		}
	}
	else if(type1 == Type.CHARACTER) {
		Character character = characters.get(key1);
		if(character != null) {
			returnValue = character.strAttrs.get(key2);
		}
	}
	else if(type1 == Type.ITEM) {
		Item item = items.get(key1);
		if(item != null) {
			returnValue = item.strAttrs.get(key2);
		}
	}

	if(returnValue == null) {
		throw new RuntimeException();
	}

	return returnValue;
}

public Item entityHasItem(String key1, Type type1, String key2) {
	Item returnValue = null;
	if(type1 == Type.LOCATION) {
		Location loc = locations.get(key1);
		if(loc != null) {
			if(loc.items.contains(key2)) {
				returnValue = items.get(key2);
			}
		}
	}
	else if(type1 == Type.CHARACTER) {
		Character character = characters.get(key1);
		if(character != null) {
			if(character.items.contains(key2)) {
				returnValue = items.get(key2);
			}
		}
	}

	if(returnValue == null) {
		throw new RuntimeException();
	}

	return returnValue;
}

public Character entityHasCharacter(String key1, Type type1, String key2) {
```

```
      Character returnValue = null;
      if(type1 == Type.LOCATION) {
            Location loc = locations.get(key1);
            if(loc != null) {
                  if(loc.characters.contains(key2)) {
                        returnValue = characters.get(key2);
                  }
            }
      }

      if(returnValue == null) {
            throw new RuntimeException();
      }

      return returnValue;
    }

    public int entityExistsItem(String key1, Type type1, String key2) {
      Object returnValue = null;
      if(type1 == Type.LOCATION) {
            Location loc = locations.get(key1);
            if(loc != null) {
                  if(loc.items.contains(key2)) {
                        returnValue = items.get(key2);
                  }
            }
      }
      else if(type1 == Type.CHARACTER) {
            Character character = characters.get(key1);
            if(character != null) {
                  if(character.items.contains(key2)) {
                        returnValue = items.get(key2);
                  }
            }
      }

      if(returnValue == null) {
            return 0;
      }

      return 1;
    }

    public int entityExistsCharacter(String key1, Type type1, String key2) {
      Object returnValue = null;
      if(type1 == Type.LOCATION) {
            Location loc = locations.get(key1);
            if(loc != null) {
                  if(loc.characters.contains(key2)) {
                        returnValue = characters.get(key2);
                  }
            }
      }

      if(returnValue == null) {
            return 0;
      }

      return 1;
    }

   public void endGame() {
      System.exit(0);
   }

   public void play() {
items.put("_the_greatest_sword_ever",_the_greatest_sword_ever);
types.put("_the_greatest_sword_ever", Type.ITEM);
_the_greatest_sword_ever.addIntAttr("_damage", (100000000));
types.put("_damage", Type.INT);
characters.put("_xiaowei_the_greatest_man_ever",_xiaowei_the_greatest_man_ever);
types.put("_xiaowei_the_greatest_man_ever", Type.CHARACTER);
```

```
_xiaowei_the_greatest_man_ever.addIntAttr("_life", (100000000));
types.put("_life", Type.INT);
_xiaowei_the_greatest_man_ever.addIntAttr("_level", (99999));
types.put("_level", Type.INT);
_xiaowei_the_greatest_man_ever.addStrAttr("_haha","hahahahaha");
types.put("_haha", Type.STRING);
_xiaowei_the_greatest_man_ever.addItem("_the_greatest_sword_ever");
locations.put("_where_is_this_place",_where_is_this_place);
types.put("_where_is_this_place", Type.LOCATION);
_where_is_this_place.addIntAttr("_sizex", (10000));
types.put("_sizex", Type.INT);
_where_is_this_place.addIntAttr("_sizey", (9283));
types.put("_sizey", Type.INT);
_where_is_this_place.showCharacter("_xiaowei_the_greatest_man_ever");
//Location function call
_where_is_this_place();
   endGame();
   }
//intdec
int _count;
//itemdec
Item _the_greatest_sword_ever = new Item();
//charadec
Character _xiaowei_the_greatest_man_ever = new Character();
//locdec
Location _where_is_this_place = new Location();
//start funtion
public void _where_is_this_place() {
currentLocation = "_where_is_this_place";
while (!((entityHasInt("_xiaowei_the_greatest_man_ever", Type.CHARACTER, "_life") <
(0)))){
{
Map<String,String> keysToActionName0 = new HashMap<String, String>();
if ((entityHasInt("_xiaowei_the_greatest_man_ever", Type.CHARACTER, "_life") < (0)))
endGame();
Map<String, String> actionNameToOutput0 = new HashMap<String, String>();
if ((entityHasInt("_xiaowei_the_greatest_man_ever", Type.CHARACTER, "_life") < (0)))
endGame();
System.out.println("CHOOSE AN ACTION:");
if ((entityHasInt("_xiaowei_the_greatest_man_ever", Type.CHARACTER, "_life") < (0)))
endGame();
keysToActionName0.put("a", "_attack");
if ((entityHasInt("_xiaowei_the_greatest_man_ever", Type.CHARACTER, "_life") < (0)))
endGame();
actionNameToOutput0.put("_attack", "hia!");
if ((entityHasInt("_xiaowei_the_greatest_man_ever", Type.CHARACTER, "_life") < (0)))
endGame();
System.out.println("Type a for hia!");
if ((entityHasInt("_xiaowei_the_greatest_man_ever", Type.CHARACTER, "_life") < (0)))
endGame();
keysToActionName0.put("u", "_up");
if ((entityHasInt("_xiaowei_the_greatest_man_ever", Type.CHARACTER, "_life") < (0)))
endGame();
actionNameToOutput0.put("_up", "up");
if ((entityHasInt("_xiaowei_the_greatest_man_ever", Type.CHARACTER, "_life") < (0)))
endGame();
System.out.println("Type u for up");
if ((entityHasInt("_xiaowei_the_greatest_man_ever", Type.CHARACTER, "_life") < (0)))
endGame();
keysToActionName0.put("f", "_fin");
if ((entityHasInt("_xiaowei_the_greatest_man_ever", Type.CHARACTER, "_life") < (0)))
endGame();
actionNameToOutput0.put("_fin", "end");
if ((entityHasInt("_xiaowei_the_greatest_man_ever", Type.CHARACTER, "_life") < (0)))
endGame();
System.out.println("Type f for end");
if ((entityHasInt("_xiaowei_the_greatest_man_ever", Type.CHARACTER, "_life") < (0)))
endGame();
Scanner in0 = new Scanner(System.in);
if ((entityHasInt("_xiaowei_the_greatest_man_ever", Type.CHARACTER, "_life") < (0)))
endGame();
String input0 = in0.nextLine();
```

```
if ((entityHasInt("_xiaowei_the_greatest_man_ever", Type.CHARACTER, "_life") < (0)))
endGame();
while(!keysToActionName0.containsKey(input0)) {
System.out.println("Invalid input, try again");
if ((entityHasInt("_xiaowei_the_greatest_man_ever", Type.CHARACTER, "_life") < (0)))
endGame();
input0 = in0.nextLine();
if ((entityHasInt("_xiaowei_the_greatest_man_ever", Type.CHARACTER, "_life") < (0)))
endGame();
}
System.out.println("You typed " + input0);
if ((entityHasInt("_xiaowei_the_greatest_man_ever", Type.CHARACTER, "_life") < (0)))
endGame();
String action0 = keysToActionName0.get(input0);
if ((entityHasInt("_xiaowei_the_greatest_man_ever", Type.CHARACTER, "_life") < (0)))
endGame();
if(action0.equals("_attack")) {
{
dummy = ((entityHasInt("_xiaowei_the_greatest_man_ever", Type.CHARACTER, "_life") +
(1)));
if ((entityHasInt("_xiaowei_the_greatest_man_ever", Type.CHARACTER, "_life") < (0)))
endGame();
}
_where_is_this_place();
if ((entityHasInt("_xiaowei_the_greatest_man_ever", Type.CHARACTER, "_life") < (0)))
endGame();
}
if(action0.equals("_up")) {
{
int num = r.nextInt(100);
if ((entityHasInt("_xiaowei_the_greatest_man_ever", Type.CHARACTER, "_life") < (0)))
endGame();
if(num >= 0 && num < 40) {
{
dummy = (_count = (_count - (1)));
if ((entityHasInt("_xiaowei_the_greatest_man_ever", Type.CHARACTER, "_life") < (0)))
endGame();
System.out.println(""+ ("count:"));
if ((entityHasInt("_xiaowei_the_greatest_man_ever", Type.CHARACTER, "_life") < (0)))
endGame();
System.out.println(""+ (_count));
if ((entityHasInt("_xiaowei_the_greatest_man_ever", Type.CHARACTER, "_life") < (0)))
endGame();
}
}
if(num >= 40 && num < 100) {
{
dummy = (_count = (_count + (1)));
if ((entityHasInt("_xiaowei_the_greatest_man_ever", Type.CHARACTER, "_life") < (0)))
endGame();
System.out.println(""+ ("count:"));
if ((entityHasInt("_xiaowei_the_greatest_man_ever", Type.CHARACTER, "_life") < (0)))
endGame();
System.out.println(""+ (_count));
if ((entityHasInt("_xiaowei_the_greatest_man_ever", Type.CHARACTER, "_life") < (0)))
endGame();
}
}
}
_where_is_this_place();
if ((entityHasInt("_xiaowei_the_greatest_man_ever", Type.CHARACTER, "_life") < (0)))
endGame();
}
if(action0.equals("_fin")) {
{
dummy = (entitySetInt("_xiaowei_the_greatest_man_ever", Type.CHARACTER, "_life", (-
(1))));
if ((entityHasInt("_xiaowei_the_greatest_man_ever", Type.CHARACTER, "_life") < (0)))
endGame();
}
_where_is_this_place();
if ((entityHasInt("_xiaowei_the_greatest_man_ever", Type.CHARACTER, "_life") < (0)))
```

```
endGame();
}
}
}
endGame();
}
}

abstract class Entity {

   Map<String, Integer> intAttrs = new HashMap<String, Integer>();
   Map<String, String> strAttrs = new HashMap<String, String>();

   public void addIntAttr(String name, int value) {
      intAttrs.put(name, value);
   }

   public void addStrAttr(String name, String value) {
      strAttrs.put(name, value);
   }
}

class Location extends Entity {
   Set<String> characters = new HashSet<String>();
   Set<String> items = new HashSet<String>();

   public void addItem(String name, Map<String, Item> itemses) {
      if(itemses.containsKey(name))
                  items.add(name);
      else
            System.out.println("Error: The item you attempted to add no longer exists");
   }

   public void addItem(String name){
      items.add(name);
      }

   public void removeItem(String name) {
      items.remove(name);
   }

   public void showCharacter(String name, Map<String, Character> characterses) {
      if(characterses.containsKey(name))
      characters.add(name);
      else
            System.out.println("Error: The character you attempted to use no longer
exists");
   }
   public void showCharacter(String name){

      characters.add(name);
      }

   public void hideCharacter(String name) {
      characters.remove(name);
   }
}

class Character extends Entity {
   Set<String> items = new HashSet<String>();

   public void addItem(String name, String locationNow, Map<String, Location> locations){

      if(locations.get(locationNow).items.contains(name)){
            locations.get(locationNow).removeItem(name);
            items.add(name);
      }
      else
            System.out.println("Error: The item you attempted to grab is not in this
location");
}
```

```
public void removeItem(String name, String locationNow, Map<String, Location> locations)
{
            if (items.contains(name)){
            items.remove(name);
            locations.get(locationNow).addItem(name);
   }
      else
            System.out.println("Error: The character does not have the item you attempted
to drop");
}

   public void addItem(String name) {
      items.add(name);
   }

   public void removeItem(String name) {
      items.remove(name);
   }
}

class Item extends Entity {
}
```

```
/*RockPaperScissors.next by Ernesto */

int won;
int lost;


location game {(int outcome),(),()}


start game end (game.outcome == 4) {
      output "Games won:";
      output won;
      output "";
      output "Games lost:";
      output lost;
      output "";
      output "rock paper scissors says shoot!";

      [?
            prob 33 game.outcome = 1;
            prob 33 game.outcome = 2;
            prob 34 game.outcome = 3;
      ?]



      choose (scissors, "scissors", "s")
            (paper, "paper", "p")
            (rock, "rock", "r")
            (exit, "exit", "e")
      {
            when scissors{
                  if(game.outcome == 1) then{
                        output "opponent picked scissors!";
                        output "You tied";
                  }
                  else if ( game.outcome == 2) then{
                        output "opponent picked paper!";
                        output "You won!!";
                        won = won + 1;
                  }
                  else{
                        output "opponent picked rock!";
                        output"You lost";
                        lost = lost + 1;
                  }
            }
            next game

            when paper{
                  if(game.outcome == 2) then{
                        output "opponent picked paper!";
                        output "You tied";
                  }
                  else if ( game.outcome == 3) then{
                        output "opponent picked rock!";
                        output "You won!!";
                        won = won + 1;
                  }
                  else{
                        output "opponent picked scissors!";
                        output"You lost";
                        lost = lost + 1;
                  }
            }
            next game

            when rock{
                  if(game.outcome == 3) then{
                        output "opponent picked rock!";
```

```
                        output "You tied";
                  }
                  else if ( game.outcome == 1) then{
                        output "opponent picked scissors!";
                        output "You won!!";
                        won = won + 1;
                  }
                  else{
                        output "opponent picked paper!";
                        output"You lost";
                        lost = lost + 1;
                  }
            }
            next game


            when exit
                  game.outcome = 4;
            next game
      }

}
```

```
//Generated Next.java file for RockPaperScissors.Next
import java.util.*;

public class Next {
   enum Type {INT, STRING, CHARACTER, ITEM, LOCATION}

   static Random r = new Random();
   Object dummy;
   String currentLocation;
   Map<String, Location> locations = new HashMap<String, Location>();
   Map<String, Character> characters = new HashMap<String, Character>();
   Map<String, Item> items = new HashMap<String, Item>();
   Map<String, Type> types = new HashMap<String, Type>();

   public static void main(String[] args) {
    (new Next()).play();
   }

   public int boolToInt(boolean value) {
        if(value) {
            return 1;
        }
        else {
            return 0;
        }
   }

   public String entitySetString(String key1, Type type1, String key2, String value) {
    boolean valueSet = false;
      if(type1 == Type.LOCATION) {
            Location loc = locations.get(key1);
            if(loc != null) {
                  loc.strAttrs.put(key2, value);
                  valueSet = true;
            }
      }
      else if(type1 == Type.CHARACTER) {
            Character character = characters.get(key1);
            if(character != null) {
                  character.strAttrs.put(key2, value);
                  valueSet = true;
            }
      }
      else if(type1 == Type.ITEM) {
            Item item = items.get(key1);
            if(item != null) {
                  item.strAttrs.put(key2, value);
                  valueSet = true;
            }
      }

      if(!valueSet) {
            throw new RuntimeException();
      }

      return value;
   }

   public int entitySetInt(String key1, Type type1, String key2, int value) {
      boolean foundReturnValue = false;

      if(type1 == Type.LOCATION) {
            Location loc = locations.get(key1);
            if(loc != null) {
            loc.intAttrs.put(key2, value);
                  foundReturnValue = true;
            }
      }
      else if(type1 == Type.CHARACTER) {
            Character character = characters.get(key1);
            if(character != null) {
                  character.intAttrs.put(key2, value);
```

```
                foundReturnValue = true;
          }
    }
    else if(type1 == Type.ITEM) {
          Item item = items.get(key1);
          if(item != null) {
                item.intAttrs.put(key2, value);
                foundReturnValue = true;
          }
    }

    if(foundReturnValue == false) {
          throw new RuntimeException();
    }

    return value;
}

    public boolean isTrue(Object object) {
        if(object instanceof String) {
           if(((String)object).isEmpty()) {
               return false;
           }
        }
        else if(object instanceof Integer) {
           if((Integer)object == 0) {
               return false;
           }
        }
        else {
            if(object == null) {
                return false;
            }
        }

        return true;
    }

  public void killFunction(String varName){
    if (characters.containsKey(varName)){
          characters.remove(varName);
          for (String key: locations.keySet()){
                      locations.get(key).hideCharacter(varName);
          }
    }
    else if (items.containsKey(varName)){
          items.remove(varName);
          for (String key:locations.keySet()){
                locations.get(key).removeItem(varName);
                }
          for (String key : characters.keySet()){
                characters.get(key).removeItem(varName);
          }
    }
  }

  public int entityHasInt(String key1, Type type1, String key2) {
    int returnValue = 0;
    boolean foundReturnValue = false;

    if(type1 == Type.LOCATION) {
          Location loc = locations.get(key1);
          if(loc != null) {
                returnValue = loc.intAttrs.get(key2);
                foundReturnValue = true;
          }
    }
    else if(type1 == Type.CHARACTER) {
          Character character = characters.get(key1);
          if(character != null) {
                returnValue = character.intAttrs.get(key2);
                foundReturnValue = true;
```

```
		}
	}
	else if(type1 == Type.ITEM) {
		Item item = items.get(key1);
		if(item != null) {
			returnValue = item.intAttrs.get(key2);
			foundReturnValue = true;
		}
	}

	if(foundReturnValue == false) {
		throw new RuntimeException();
	}

	return returnValue;
}

public String entityHasString(String key1, Type type1, String key2) {
	String returnValue = null;
	if(type1 == Type.LOCATION) {
		Location loc = locations.get(key1);
		if(loc != null) {
			returnValue = loc.strAttrs.get(key2);
		}
	}
	else if(type1 == Type.CHARACTER) {
		Character character = characters.get(key1);
		if(character != null) {
			returnValue = character.strAttrs.get(key2);
		}
	}
	else if(type1 == Type.ITEM) {
		Item item = items.get(key1);
		if(item != null) {
			returnValue = item.strAttrs.get(key2);
		}
	}

	if(returnValue == null) {
		throw new RuntimeException();
	}

	return returnValue;
}

public Item entityHasItem(String key1, Type type1, String key2) {
	Item returnValue = null;
	if(type1 == Type.LOCATION) {
		Location loc = locations.get(key1);
		if(loc != null) {
			if(loc.items.contains(key2)) {
				returnValue = items.get(key2);
			}
		}
	}
	else if(type1 == Type.CHARACTER) {
		Character character = characters.get(key1);
		if(character != null) {
			if(character.items.contains(key2)) {
				returnValue = items.get(key2);
			}
		}
	}

	if(returnValue == null) {
		throw new RuntimeException();
	}

	return returnValue;
}

public Character entityHasCharacter(String key1, Type type1, String key2) {
```

```
      Character returnValue = null;
      if(type1 == Type.LOCATION) {
            Location loc = locations.get(key1);
            if(loc != null) {
                  if(loc.characters.contains(key2)) {
                        returnValue = characters.get(key2);
                  }
            }
      }

      if(returnValue == null) {
            throw new RuntimeException();
      }

      return returnValue;
    }

    public int entityExistsItem(String key1, Type type1, String key2) {
      Object returnValue = null;
      if(type1 == Type.LOCATION) {
            Location loc = locations.get(key1);
            if(loc != null) {
                  if(loc.items.contains(key2)) {
                        returnValue = items.get(key2);
                  }
            }
      }
      else if(type1 == Type.CHARACTER) {
            Character character = characters.get(key1);
            if(character != null) {
                  if(character.items.contains(key2)) {
                        returnValue = items.get(key2);
                  }
            }
      }

      if(returnValue == null) {
            return 0;
      }

      return 1;
    }

    public int entityExistsCharacter(String key1, Type type1, String key2) {
      Object returnValue = null;
      if(type1 == Type.LOCATION) {
            Location loc = locations.get(key1);
            if(loc != null) {
                  if(loc.characters.contains(key2)) {
                        returnValue = characters.get(key2);
                  }
            }
      }

      if(returnValue == null) {
            return 0;
      }

      return 1;
    }

   public void endGame() {
      System.exit(0);
   }

   public void play() {
locations.put("_game",_game);
types.put("_game", Type.LOCATION);
_game.addIntAttr("_outcome", 0);
types.put("_outcome", Type.INT);
//Location function call
_game();
```

```
   endGame();
   }
//intdec
int _won;
//intdec
int _lost;
//locdec
Location _game = new Location();
//start funtion
public

void _game() {
currentLocation = "_game";
while (!(entityHasInt("_game", Type.LOCATION, "_outcome") ==

(4))){
{
System.out.println(""+ ("Games won:"));
if (entityHasInt("_game", Type.LOCATION, "_outcome") ==

(4))
endGame();
System.out.println(""+ (_won));
if (entityHasInt("_game", Type.LOCATION, "_outcome") ==

(4))
endGame();
System.out.println(""+ (""));
if (entityHasInt("_game", Type.LOCATION, "_outcome") ==

(4))
endGame();
System.out.println(""+ ("Games lost:"));
if (entityHasInt("_game", Type.LOCATION,

"_outcome") == (4))
endGame();
System.out.println(""+ (_lost));
if (entityHasInt("_game", Type.LOCATION,

"_outcome") == (4))
endGame();
System.out.println(""+ (""));
if (entityHasInt("_game", Type.LOCATION,

"_outcome") == (4))
endGame();
System.out.println(""+ ("rock paper scissors says shoot!"));
if

(entityHasInt("_game", Type.LOCATION, "_outcome") == (4))
endGame();
int num = r.nextInt(100);
if

(entityHasInt("_game", Type.LOCATION, "_outcome") == (4))
endGame();
if(num >= 0 && num < 33) {
dummy =

(entitySetInt("_game", Type.LOCATION, "_outcome", (1)));
if (entityHasInt("_game", Type.LOCATION,

"_outcome") == (4))
endGame();
}
if(num >= 33 && num < 66) {
dummy = (entitySetInt("_game", Type.LOCATION,

"_outcome", (2)));
if (entityHasInt("_game", Type.LOCATION, "_outcome") == (4))
endGame();
```

```
}
if(num >= 66

&& num < 100) {
dummy = (entitySetInt("_game", Type.LOCATION, "_outcome", (3)));
if (entityHasInt

("_game", Type.LOCATION, "_outcome") == (4))
endGame();
}
Map<String,String> keysToActionName17 = new

HashMap<String, String>();
if (entityHasInt("_game", Type.LOCATION, "_outcome") == (4))
endGame();
Map<String, String> actionNameToOutput17 = new HashMap<String, String>();
if (entityHasInt("_game",

Type.LOCATION, "_outcome") == (4))
endGame();
System.out.println("CHOOSE AN ACTION:");
if (entityHasInt

("_game", Type.LOCATION, "_outcome") == (4))
endGame();
keysToActionName17.put("s", "_scissors");
if

(entityHasInt("_game", Type.LOCATION, "_outcome") == (4))
endGame();
actionNameToOutput17.put

("_scissors", "scissors");
if (entityHasInt("_game", Type.LOCATION, "_outcome") == (4))
endGame();
System.out.println("Type s for scissors");
if (entityHasInt("_game", Type.LOCATION, "_outcome") == (4))
endGame();
keysToActionName17.put("p", "_paper");
if (entityHasInt("_game", Type.LOCATION, "_outcome")

== (4))
endGame();
actionNameToOutput17.put("_paper", "paper");
if (entityHasInt("_game", Type.LOCATION,

"_outcome") == (4))
endGame();
System.out.println("Type p for paper");
if (entityHasInt("_game",

Type.LOCATION, "_outcome") == (4))
endGame();
keysToActionName17.put("r", "_rock");
if (entityHasInt

("_game", Type.LOCATION, "_outcome") == (4))
endGame();
actionNameToOutput17.put("_rock", "rock");
if

(entityHasInt("_game", Type.LOCATION, "_outcome") == (4))
endGame();
System.out.println("Type r for

rock");
if (entityHasInt("_game", Type.LOCATION, "_outcome") == (4))
endGame();
keysToActionName17.put

("e", "_exit");
if (entityHasInt("_game", Type.LOCATION, "_outcome") == (4))
```

```
endGame();
actionNameToOutput17.put("_exit", "exit");
if (entityHasInt("_game", Type.LOCATION, "_outcome") == (4))
endGame();
System.out.println("Type e for exit");
if (entityHasInt("_game", Type.LOCATION, "_outcome")

== (4))
endGame();
Scanner in17 = new Scanner(System.in);
if (entityHasInt("_game", Type.LOCATION,

"_outcome") == (4))
endGame();
String input17 = in17.nextLine();
if (entityHasInt("_game", Type.LOCATION,

"_outcome") == (4))
endGame();
while(!keysToActionName17.containsKey(input17)) {
System.out.println

("Invalid input, try again");
if (entityHasInt("_game", Type.LOCATION, "_outcome") == (4))
endGame();
input17 = in17.nextLine();
if (entityHasInt("_game", Type.LOCATION, "_outcome") == (4))
endGame();
}
System.out.println("You typed " + input17);
if (entityHasInt("_game", Type.LOCATION, "_outcome") ==

(4))
endGame();
String action17 = keysToActionName17.get(input17);
if (entityHasInt("_game",

Type.LOCATION, "_outcome") == (4))
endGame();
if(action17.equals("_scissors")) {
{
if(entityHasInt

("_game", Type.LOCATION, "_outcome") == (1)) {
{
System.out.println(""+ ("opponent picked scissors!"));
if (entityHasInt("_game", Type.LOCATION, "_outcome") == (4))
endGame();
System.out.println(""+ ("You

tied"));
if (entityHasInt("_game", Type.LOCATION, "_outcome") == (4))
endGame();
}
}
else {
if(entityHasInt

("_game", Type.LOCATION, "_outcome") == (2)) {
{
System.out.println(""+ ("opponent picked paper!"));
if

(entityHasInt("_game", Type.LOCATION, "_outcome") == (4))
endGame();
System.out.println(""+ ("You

won!!"));
if (entityHasInt("_game", Type.LOCATION, "_outcome") == (4))
endGame();
dummy = (_won = (_won +
```

```
(1)));
if (entityHasInt("_game", Type.LOCATION, "_outcome") == (4))
endGame();
}
}
else {
{
System.out.println(""+ ("opponent picked rock!"));
if (entityHasInt("_game", Type.LOCATION, "_outcome")

== (4))
endGame();
System.out.println(""+ ("You lost"));
if (entityHasInt("_game", Type.LOCATION,

"_outcome") == (4))
endGame();
dummy = (_lost = (_lost + (1)));
if (entityHasInt("_game", Type.LOCATION,

"_outcome") == (4))
endGame();
}
}
}
}
_game();
if (entityHasInt("_game", Type.LOCATION, "_outcome") == (4))
endGame();
}
if(action17.equals("_paper")) {
{
if(entityHasInt("_game", Type.LOCATION, "_outcome") == (2))

{
}
{
System.out.println(""+ ("opponent picked paper!"));
if (entityHasInt("_game", Type.LOCATION,

"_outcome") == (4))
endGame();
System.out.println(""+ ("You tied"));
if (entityHasInt("_game",

Type.LOCATION, "_outcome") == (4))
endGame();
}
}
else {
if(entityHasInt("_game", Type.LOCATION,

"_outcome") == (3)) {
{
System.out.println(""+ ("opponent picked rock!"));
if (entityHasInt("_game",

Type.LOCATION, "_outcome") == (4))
endGame();
System.out.println(""+ ("You won!!"));
if (entityHasInt

("_game", Type.LOCATION, "_outcome") == (4))
endGame();
dummy = (_won = (_won + (1)));
if (entityHasInt

("_game", Type.LOCATION, "_outcome") == (4))
endGame();
}
}
else {
{
```

```
System.out.println(""+ ("opponent

picked scissors!"));
if (entityHasInt("_game", Type.LOCATION, "_outcome") == (4))
endGame();
System.out.println(""+ ("You lost"));
if (entityHasInt("_game", Type.LOCATION, "_outcome") == (4))
endGame();
dummy = (_lost = (_lost + (1)));
if (entityHasInt("_game", Type.LOCATION, "_outcome") == (4))
endGame();
}
}
}
}
_game();
if (entityHasInt("_game", Type.LOCATION, "_outcome") == (4))
endGame();
}
if

(action17.equals("_rock")) {
{
if(entityHasInt("_game", Type.LOCATION, "_outcome") == (3)) {
{
System.out.println(""+ ("opponent picked rock!"));
if (entityHasInt("_game", Type.LOCATION, "_outcome")

== (4))
endGame();
System.out.println(""+ ("You tied"));
if (entityHasInt("_game", Type.LOCATION,

"_outcome") == (4))
endGame();
}
}
else {
if(entityHasInt("_game", Type.LOCATION, "_outcome") == (1)) {
{
System.out.println(""+ ("opponent picked scissors!"));
if (entityHasInt("_game", Type.LOCATION,

"_outcome") == (4))
endGame();
System.out.println(""+ ("You won!!"));
if (entityHasInt("_game",

Type.LOCATION, "_outcome") == (4))
endGame();
dummy = (_won = (_won + (1)));
if (entityHasInt("_game",

Type.LOCATION, "_outcome") == (4))
endGame();
}
}
else {
{
System.out.println(""+ ("opponent picked

paper!"));
if (entityHasInt("_game", Type.LOCATION, "_outcome") == (4))
endGame();
System.out.println(""+

("You lost"));
if (entityHasInt("_game", Type.LOCATION, "_outcome") == (4))
endGame();
dummy = (_lost =

(_lost + (1)));
```

```
if (entityHasInt("_game", Type.LOCATION, "_outcome") == (4))
endGame();
}
}
}
}
_game();
if

(entityHasInt("_game", Type.LOCATION, "_outcome") == (4))
endGame();
}
if(action17.equals("_exit")) {
dummy = (entitySetInt("_game", Type.LOCATION, "_outcome", (4)));
if (entityHasInt("_game",

Type.LOCATION, "_outcome") == (4))
endGame();
_game();
if (entityHasInt("_game", Type.LOCATION,

"_outcome") == (4))
endGame();
}
}
}
endGame();
}
}

abstract class Entity {

   Map<String, Integer> intAttrs = new HashMap<String, Integer>();
   Map<String, String> strAttrs = new HashMap<String, String>();

   public void addIntAttr(String name, int value) {
      intAttrs.put(name, value);
   }

   public void addStrAttr(String name, String value) {
      strAttrs.put(name, value);
   }
}

class Location extends Entity {
   Set<String> characters = new HashSet<String>();
   Set<String> items = new HashSet<String>();

   public void addItem(String name, Map<String, Item> itemses) {
      if(itemses.containsKey(name))
                  items.add(name);
      else
            System.out.println("Error: The item you attempted to add no longer exists");
   }

   public void addItem(String name){
      items.add(name);
      }

   public void removeItem(String name) {
      items.remove(name);
   }

   public void showCharacter(String name, Map<String, Character> characterses) {
      if(characterses.containsKey(name))
      characters.add(name);
      else
            System.out.println("Error: The character you attempted to use no longer
exists");
   }
   public void showCharacter(String name){
```

```
      characters.add(name);
      }

   public void hideCharacter(String name) {
      characters.remove(name);
   }
}

class Character extends Entity {
   Set<String> items = new HashSet<String>();

   public void addItem(String name, String locationNow, Map<String, Location> locations){

      if(locations.get(locationNow).items.contains(name)){
            locations.get(locationNow).removeItem(name);
            items.add(name);
      }
      else
            System.out.println("Error: The item you attempted to grab is not in this
location");
}

public void removeItem(String name, String locationNow, Map<String, Location> locations)
{
            if (items.contains(name)){
            items.remove(name);
            locations.get(locationNow).addItem(name);
   }
      else
            System.out.println("Error: The character does not have the item you attempted
to

drop");
}

   public void addItem(String name) {
      items.add(name);
   }

   public void removeItem(String name) {
      items.remove(name);
   }
}

class Item extends Entity {
}
```

```
/*Tutorial1.next by Ernesto */
int num = 0;
string announce = "not in the world:";
character you {(string slogan = "Hello world!"),()}
location here {(),(),(you)}

start here end (num == 1) {
      if (exists here.you) then
            output you.slogan;
      else
            output announce;
      num = 1;

}
```

```
//Next.java file generated for Tutorial1.Next
import java.util.*;

public class Next {
   enum Type {INT, STRING, CHARACTER, ITEM, LOCATION}

   static Random r = new Random();
   Object dummy;
   String currentLocation;
   Map<String, Location> locations = new HashMap<String, Location>();
   Map<String, Character> characters = new HashMap<String, Character>();
   Map<String, Item> items = new HashMap<String, Item>();
   Map<String, Type> types = new HashMap<String, Type>();

   public static void main(String[] args) {
    (new Next()).play();
   }

   public int boolToInt(boolean value) {
        if(value) {
            return 1;
        }
        else {
            return 0;
        }
   }

   public String entitySetString(String key1, Type type1, String key2, String value) {
    boolean valueSet = false;
      if(type1 == Type.LOCATION) {
            Location loc = locations.get(key1);
            if(loc != null) {
                  loc.strAttrs.put(key2, value);
                  valueSet = true;
            }
      }
      else if(type1 == Type.CHARACTER) {
            Character character = characters.get(key1);
            if(character != null) {
                  character.strAttrs.put(key2, value);
                  valueSet = true;
            }
      }
      else if(type1 == Type.ITEM) {
            Item item = items.get(key1);
            if(item != null) {
                  item.strAttrs.put(key2, value);
                  valueSet = true;
            }
      }

      if(!valueSet) {
            throw new RuntimeException();
      }

      return value;
   }

   public int entitySetInt(String key1, Type type1, String key2, int value) {
      boolean foundReturnValue = false;

      if(type1 == Type.LOCATION) {
            Location loc = locations.get(key1);
            if(loc != null) {
            loc.intAttrs.put(key2, value);
                  foundReturnValue = true;
            }
      }
      else if(type1 == Type.CHARACTER) {
            Character character = characters.get(key1);
            if(character != null) {
                  character.intAttrs.put(key2, value);
```

```
                foundReturnValue = true;
          }
    }
    else if(type1 == Type.ITEM) {
          Item item = items.get(key1);
          if(item != null) {
                item.intAttrs.put(key2, value);
                foundReturnValue = true;
          }
    }

    if(foundReturnValue == false) {
          throw new RuntimeException();
    }

    return value;
}

    public boolean isTrue(Object object) {
        if(object instanceof String) {
           if(((String)object).isEmpty()) {
               return false;
           }
        }
        else if(object instanceof Integer) {
           if((Integer)object == 0) {
               return false;
           }
        }
        else {
            if(object == null) {
                return false;
            }
        }

        return true;
    }

  public void killFunction(String varName){
    if (characters.containsKey(varName)){
          characters.remove(varName);
          for (String key: locations.keySet()){
                      locations.get(key).hideCharacter(varName);
          }
    }
    else if (items.containsKey(varName)){
          items.remove(varName);
          for (String key:locations.keySet()){
                locations.get(key).removeItem(varName);
                }
          for (String key : characters.keySet()){
                characters.get(key).removeItem(varName);
          }
    }
  }

  public int entityHasInt(String key1, Type type1, String key2) {
    int returnValue = 0;
    boolean foundReturnValue = false;

    if(type1 == Type.LOCATION) {
          Location loc = locations.get(key1);
          if(loc != null) {
                returnValue = loc.intAttrs.get(key2);
                foundReturnValue = true;
          }
    }
    else if(type1 == Type.CHARACTER) {
          Character character = characters.get(key1);
          if(character != null) {
                returnValue = character.intAttrs.get(key2);
                foundReturnValue = true;
```

```
		}
	}
	else if(type1 == Type.ITEM) {
		Item item = items.get(key1);
		if(item != null) {
			returnValue = item.intAttrs.get(key2);
			foundReturnValue = true;
		}
	}

	if(foundReturnValue == false) {
		throw new RuntimeException();
	}

	return returnValue;
}

public String entityHasString(String key1, Type type1, String key2) {
	String returnValue = null;
	if(type1 == Type.LOCATION) {
		Location loc = locations.get(key1);
		if(loc != null) {
			returnValue = loc.strAttrs.get(key2);
		}
	}
	else if(type1 == Type.CHARACTER) {
		Character character = characters.get(key1);
		if(character != null) {
			returnValue = character.strAttrs.get(key2);
		}
	}
	else if(type1 == Type.ITEM) {
		Item item = items.get(key1);
		if(item != null) {
			returnValue = item.strAttrs.get(key2);
		}
	}

	if(returnValue == null) {
		throw new RuntimeException();
	}

	return returnValue;
}

public Item entityHasItem(String key1, Type type1, String key2) {
	Item returnValue = null;
	if(type1 == Type.LOCATION) {
		Location loc = locations.get(key1);
		if(loc != null) {
			if(loc.items.contains(key2)) {
				returnValue = items.get(key2);
			}
		}
	}
	else if(type1 == Type.CHARACTER) {
		Character character = characters.get(key1);
		if(character != null) {
			if(character.items.contains(key2)) {
				returnValue = items.get(key2);
			}
		}
	}

	if(returnValue == null) {
		throw new RuntimeException();
	}

	return returnValue;
}

public Character entityHasCharacter(String key1, Type type1, String key2) {
```

```
      Character returnValue = null;
      if(type1 == Type.LOCATION) {
            Location loc = locations.get(key1);
            if(loc != null) {
                  if(loc.characters.contains(key2)) {
                        returnValue = characters.get(key2);
                  }
            }
      }

      if(returnValue == null) {
            throw new RuntimeException();
      }

      return returnValue;
    }

    public int entityExistsItem(String key1, Type type1, String key2) {
      Object returnValue = null;
      if(type1 == Type.LOCATION) {
            Location loc = locations.get(key1);
            if(loc != null) {
                  if(loc.items.contains(key2)) {
                        returnValue = items.get(key2);
                  }
            }
      }
      else if(type1 == Type.CHARACTER) {
            Character character = characters.get(key1);
            if(character != null) {
                  if(character.items.contains(key2)) {
                        returnValue = items.get(key2);
                  }
            }
      }

      if(returnValue == null) {
            return 0;
      }

      return 1;
    }

    public int entityExistsCharacter(String key1, Type type1, String key2) {
      Object returnValue = null;
      if(type1 == Type.LOCATION) {
            Location loc = locations.get(key1);
            if(loc != null) {
                  if(loc.characters.contains(key2)) {
                        returnValue = characters.get(key2);
                  }
            }
      }

      if(returnValue == null) {
            return 0;
      }

      return 1;
    }

   public void endGame() {
      System.exit(0);
   }

   public void play() {
_num = (0);
_announce = "not in the world:";
characters.put("_you",_you);
types.put("_you",

Type.CHARACTER);
```

```
_you.addStrAttr("_slogan","Hello world!");
types.put("_slogan", Type.STRING);
locations.put("_here",_here);
types.put("_here", Type.LOCATION);
_here.showCharacter("_you");
//Location

function call
_here();
   endGame();
   }
//intdecinit
int _num;
//strdecinit
String _announce;
//charadec
Character _you = new Character();
//locdec
Location _here = new Location();
//start funtion
public void _here() {
currentLocation = "_here";
while (!(_num == (1))){
{
if(isTrue(entityExistsCharacter("_here", Type.LOCATION, "_you"))) {
System.out.println(""+ (entityHasString("_you", Type.CHARACTER, "_slogan")));
if (_num == (1))
endGame

();
}
else {
System.out.println(""+ (_announce));
if (_num == (1))
endGame();
}
dummy = (_num = (1));
if (_num

== (1))
endGame();
}
}
endGame();
}
}

abstract class Entity {

   Map<String, Integer> intAttrs = new HashMap<String, Integer>();
   Map<String, String> strAttrs = new HashMap<String, String>();

   public void addIntAttr(String name, int value) {
      intAttrs.put(name, value);
   }

   public void addStrAttr(String name, String value) {
      strAttrs.put(name, value);
   }
}

class Location extends Entity {
   Set<String> characters = new HashSet<String>();
   Set<String> items = new HashSet<String>();

   public void addItem(String name, Map<String, Item> itemses) {
      if(itemses.containsKey(name))
                  items.add(name);
      else
            System.out.println("Error: The item you attempted to add no longer exists");
   }
```

```
   public void addItem(String name){
      items.add(name);
      }

   public void removeItem(String name) {
      items.remove(name);
   }

   public void showCharacter(String name, Map<String, Character> characterses) {
      if(characterses.containsKey(name))
      characters.add(name);
      else
            System.out.println("Error: The character you attempted to use no longer
exists");
   }
   public void showCharacter(String name){

      characters.add(name);
      }

   public void hideCharacter(String name) {
      characters.remove(name);
   }
}

class Character extends Entity {
   Set<String> items = new HashSet<String>();

   public void addItem(String name, String locationNow, Map<String, Location> locations){

      if(locations.get(locationNow).items.contains(name)){
            locations.get(locationNow).removeItem(name);
            items.add(name);
      }
      else
            System.out.println("Error: The item you attempted to grab is not in this
location");
}

public void removeItem(String name, String locationNow, Map<String, Location> locations)
{
            if (items.contains(name)){
            items.remove(name);
            locations.get(locationNow).addItem(name);
   }
      else
            System.out.println("Error: The character does not have the item you attempted
to

drop");
}

   public void addItem(String name) {
      items.add(name);
   }

   public void removeItem(String name) {
      items.remove(name);
   }
}

class Item extends Entity {
}
```

```
/*Tutorial2.next by Ernesto */
int num;

character person {(),()}
location here {(),(),()}
item object {()}

start here end (num == 1) {

      [?
            prob 50 output "This is line 1 of a possible 2";
            prob 50 output "This is line 2 of a possible 2";
      ?]

      if (exists person.object) then
            output "The person has the object";
      else
            output "The person does not have the object";

      if(exists here.object) then
            output "The object is in the location";
      else
            output "The object is not in the location";


      choose (grabItem, "character grab item", "g")
            (dropItem, "character drop item", "d")
            (showItem, "show item in location", "s")
            (hideItem, "hide item from location", "h")
            (exit, "exit", "e")
      {
            when grabItem
                  grab person.object;
            next here

            when dropItem
                  drop person.object;
            next here

            when showItem
                  show here.object;
            next here

            when hideItem
                  hide here.object;
            next here

            when exit
                  num = 1;
            next here
      }

}
```

```
//Next.java file generated for Tutorial2.next
import java.util.*;

public class Next {
   enum Type {INT, STRING, CHARACTER, ITEM, LOCATION}

   static Random r = new Random();
   Object dummy;
   String currentLocation;
   Map<String, Location> locations = new HashMap<String, Location>();
   Map<String, Character> characters = new HashMap<String, Character>();
   Map<String, Item> items = new HashMap<String, Item>();
   Map<String, Type> types = new HashMap<String, Type>();

   public static void main(String[] args) {
    (new Next()).play();
   }

   public int boolToInt(boolean value) {
        if(value) {
            return 1;
        }
        else {
            return 0;
        }
   }

   public String entitySetString(String key1, Type type1, String key2, String value) {
    boolean valueSet = false;
      if(type1 == Type.LOCATION) {
            Location loc = locations.get(key1);
            if(loc != null) {
                  loc.strAttrs.put(key2, value);
                  valueSet = true;
            }
      }
      else if(type1 == Type.CHARACTER) {
            Character character = characters.get(key1);
            if(character != null) {
                  character.strAttrs.put(key2, value);
                  valueSet = true;
            }
      }
      else if(type1 == Type.ITEM) {
            Item item = items.get(key1);
            if(item != null) {
                  item.strAttrs.put(key2, value);
                  valueSet = true;
            }
      }

      if(!valueSet) {
            throw new RuntimeException();
      }

      return value;
   }

   public int entitySetInt(String key1, Type type1, String key2, int value) {
      boolean foundReturnValue = false;

      if(type1 == Type.LOCATION) {
            Location loc = locations.get(key1);
            if(loc != null) {
            loc.intAttrs.put(key2, value);
                  foundReturnValue = true;
            }
      }
      else if(type1 == Type.CHARACTER) {
            Character character = characters.get(key1);
            if(character != null) {
                  character.intAttrs.put(key2, value);
```

```
                foundReturnValue = true;
          }
    }
    else if(type1 == Type.ITEM) {
          Item item = items.get(key1);
          if(item != null) {
                item.intAttrs.put(key2, value);
                foundReturnValue = true;
          }
    }

    if(foundReturnValue == false) {
          throw new RuntimeException();
    }

    return value;
}

    public boolean isTrue(Object object) {
        if(object instanceof String) {
           if(((String)object).isEmpty()) {
               return false;
           }
        }
        else if(object instanceof Integer) {
           if((Integer)object == 0) {
               return false;
           }
        }
        else {
            if(object == null) {
                return false;
            }
        }

        return true;
    }

  public void killFunction(String varName){
    if (characters.containsKey(varName)){
          characters.remove(varName);
          for (String key: locations.keySet()){
                      locations.get(key).hideCharacter(varName);
          }
    }
    else if (items.containsKey(varName)){
          items.remove(varName);
          for (String key:locations.keySet()){
                locations.get(key).removeItem(varName);
                }
          for (String key : characters.keySet()){
                characters.get(key).removeItem(varName);
          }
    }
  }

  public int entityHasInt(String key1, Type type1, String key2) {
    int returnValue = 0;
    boolean foundReturnValue = false;

    if(type1 == Type.LOCATION) {
          Location loc = locations.get(key1);
          if(loc != null) {
                returnValue = loc.intAttrs.get(key2);
                foundReturnValue = true;
          }
    }
    else if(type1 == Type.CHARACTER) {
          Character character = characters.get(key1);
          if(character != null) {
                returnValue = character.intAttrs.get(key2);
                foundReturnValue = true;
```

```
		}
	}
	else if(type1 == Type.ITEM) {
		Item item = items.get(key1);
		if(item != null) {
			returnValue = item.intAttrs.get(key2);
			foundReturnValue = true;
		}
	}

	if(foundReturnValue == false) {
		throw new RuntimeException();
	}

	return returnValue;
}

public String entityHasString(String key1, Type type1, String key2) {
	String returnValue = null;
	if(type1 == Type.LOCATION) {
		Location loc = locations.get(key1);
		if(loc != null) {
			returnValue = loc.strAttrs.get(key2);
		}
	}
	else if(type1 == Type.CHARACTER) {
		Character character = characters.get(key1);
		if(character != null) {
			returnValue = character.strAttrs.get(key2);
		}
	}
	else if(type1 == Type.ITEM) {
		Item item = items.get(key1);
		if(item != null) {
			returnValue = item.strAttrs.get(key2);
		}
	}

	if(returnValue == null) {
		throw new RuntimeException();
	}

	return returnValue;
}

public Item entityHasItem(String key1, Type type1, String key2) {
	Item returnValue = null;
	if(type1 == Type.LOCATION) {
		Location loc = locations.get(key1);
		if(loc != null) {
			if(loc.items.contains(key2)) {
				returnValue = items.get(key2);
			}
		}
	}
	else if(type1 == Type.CHARACTER) {
		Character character = characters.get(key1);
		if(character != null) {
			if(character.items.contains(key2)) {
				returnValue = items.get(key2);
			}
		}
	}

	if(returnValue == null) {
		throw new RuntimeException();
	}

	return returnValue;
}

public Character entityHasCharacter(String key1, Type type1, String key2) {
```

```
      Character returnValue = null;
      if(type1 == Type.LOCATION) {
            Location loc = locations.get(key1);
            if(loc != null) {
                  if(loc.characters.contains(key2)) {
                        returnValue = characters.get(key2);
                  }
            }
      }

      if(returnValue == null) {
            throw new RuntimeException();
      }

      return returnValue;
    }

    public int entityExistsItem(String key1, Type type1, String key2) {
      Object returnValue = null;
      if(type1 == Type.LOCATION) {
            Location loc = locations.get(key1);
            if(loc != null) {
                  if(loc.items.contains(key2)) {
                        returnValue = items.get(key2);
                  }
            }
      }
      else if(type1 == Type.CHARACTER) {
            Character character = characters.get(key1);
            if(character != null) {
                  if(character.items.contains(key2)) {
                        returnValue = items.get(key2);
                  }
            }
      }

      if(returnValue == null) {
            return 0;
      }

      return 1;
    }

    public int entityExistsCharacter(String key1, Type type1, String key2) {
      Object returnValue = null;
      if(type1 == Type.LOCATION) {
            Location loc = locations.get(key1);
            if(loc != null) {
                  if(loc.characters.contains(key2)) {
                        returnValue = characters.get(key2);
                  }
            }
      }

      if(returnValue == null) {
            return 0;
      }

      return 1;
    }

   public void endGame() {
      System.exit(0);
   }

   public void play() {
characters.put("_person",_person);
types.put("_person", Type.CHARACTER);
locations.put("_here",_here);
types.put("_here", Type.LOCATION);
items.put("_object",_object);
types.put("_object", Type.ITEM);
```

```
//Location function call
_here();
   endGame();
   }
//intdec
int _num;
//charadec
Character _person = new Character();
//locdec
Location _here = new

Location();
//itemdec
Item _object = new Item();
//start funtion
public void _here() {
currentLocation =

"_here";
while (!(_num == (1))){
{
int num = r.nextInt(100);
if (_num == (1))
endGame();
if(num >= 0 && num

< 50) {
System.out.println(""+ ("This is line 1 of a possible 2"));
if (_num == (1))
endGame();
}
if(num >=

50 && num < 100) {
System.out.println(""+ ("This is line 2 of a possible 2"));
if (_num == (1))
endGame

();
}
if(isTrue(entityExistsItem("_person", Type.CHARACTER, "_object"))) {
System.out.println(""+ ("The

person has the object"));
if (_num == (1))
endGame();
}
else {
System.out.println(""+ ("The person does not

have the object"));
if (_num == (1))
endGame();
}
if(isTrue(entityExistsItem("_here", Type.LOCATION,

"_object"))) {
System.out.println(""+ ("The object is in the location"));
if (_num == (1))
endGame();
}
else {
System.out.println(""+ ("The object is not in the location"));
if (_num == (1))
endGame();
}
Map<String,String> keysToActionName19 = new HashMap<String, String>();
if (_num == (1))
endGame();
Map<String, String> actionNameToOutput19 = new HashMap<String, String>();
if (_num == (1))
endGame();
```

```
System.out.println("CHOOSE AN ACTION:");
if (_num == (1))
endGame();
keysToActionName19.put("g",

"_grabItem");
if (_num == (1))
endGame();
actionNameToOutput19.put("_grabItem", "character grab item");
if

(_num == (1))
endGame();
System.out.println("Type g for character grab item");
if (_num == (1))
endGame();
keysToActionName19.put("d", "_dropItem");
if (_num == (1))
endGame();
actionNameToOutput19.put

("_dropItem", "character drop item");
if (_num == (1))
endGame();
System.out.println("Type d for

character drop item");
if (_num == (1))
endGame();
keysToActionName19.put("s", "_showItem");
if (_num ==

(1))
endGame();
actionNameToOutput19.put("_showItem", "show item in location");
if (_num == (1))
endGame

();
System.out.println("Type s for show item in location");
if (_num == (1))
endGame();
keysToActionName19.put("h", "_hideItem");
if (_num == (1))
endGame();
actionNameToOutput19.put

("_hideItem", "hide item from location");
if (_num == (1))
endGame();
System.out.println("Type h for hide

item from location");
if (_num == (1))
endGame();
keysToActionName19.put("e", "_exit");
if (_num == (1))
endGame();
actionNameToOutput19.put("_exit", "exit");
if (_num == (1))
endGame();
System.out.println("Type

e for exit");
if (_num == (1))
endGame();
Scanner in19 = new Scanner(System.in);
if (_num == (1))
endGame

();
String input19 = in19.nextLine();
```

```
if (_num == (1))
endGame();
while(!keysToActionName19.containsKey

(input19)) {
System.out.println("Invalid input, try again");
if (_num == (1))
endGame();
input19 =

in19.nextLine();
if (_num == (1))
endGame();
}
System.out.println("You typed " + input19);
if (_num == (1))
endGame();
String action19 = keysToActionName19.get(input19);
if (_num == (1))
endGame();
if

(action19.equals("_grabItem")) {
_person.addItem ("_object", currentLocation, locations);
if (_num ==

(1))
endGame();
_here();
if (_num == (1))
endGame();
}
if(action19.equals("_dropItem")) {
_person.removeItem

("_object", currentLocation, locations);
if (_num == (1))
endGame();
_here();
if (_num == (1))
endGame();
}
if(action19.equals("_showItem")) {
_here.addItem ("_object",items);
if (_num == (1))
endGame();
_here();
if

(_num == (1))
endGame();
}
if(action19.equals("_hideItem")) {
_here.removeItem ("_object");
if (_num ==

(1))
endGame();
_here();
if (_num == (1))
endGame();
}
if(action19.equals("_exit")) {
dummy = (_num = (1));
if

(_num == (1))
endGame();
_here();
if (_num == (1))
endGame();
}
```

```
}
}
endGame();
}
}

abstract class Entity {

   Map<String, Integer> intAttrs = new HashMap<String, Integer>();
   Map<String, String> strAttrs = new HashMap<String, String>();

   public void addIntAttr(String name, int value) {
      intAttrs.put(name, value);
   }

   public void addStrAttr(String name, String value) {
      strAttrs.put(name, value);
   }
}

class Location extends Entity {
   Set<String> characters = new HashSet<String>();
   Set<String> items = new HashSet<String>();

   public void addItem(String name, Map<String, Item> itemses) {
      if(itemses.containsKey(name))
                  items.add(name);
      else
            System.out.println("Error: The item you attempted to add no longer exists");
   }

   public void addItem(String name){
      items.add(name);
      }

   public void removeItem(String name) {
      items.remove(name);
   }

   public void showCharacter(String name, Map<String, Character> characterses) {
      if(characterses.containsKey(name))
      characters.add(name);
      else
            System.out.println("Error: The character you attempted to use no longer
exists");
   }
   public void showCharacter(String name){

      characters.add(name);
      }

   public void hideCharacter(String name) {
      characters.remove(name);
   }
}

class Character extends Entity {
   Set<String> items = new HashSet<String>();

   public void addItem(String name, String locationNow, Map<String, Location> locations){

      if(locations.get(locationNow).items.contains(name)){
            locations.get(locationNow).removeItem(name);
            items.add(name);
      }
      else
            System.out.println("Error: The item you attempted to grab is not in this
location");
}

public void removeItem(String name, String locationNow, Map<String, Location> locations)
{
```

```
            if (items.contains(name)){
            items.remove(name);
            locations.get(locationNow).addItem(name);
   }
      else
            System.out.println("Error: The character does not have the item you attempted
to

drop");
}

   public void addItem(String name) {
      items.add(name);
   }

   public void removeItem(String name) {
      items.remove(name);
   }
}

class Item extends Entity {
}
```

```
/*ProfEdwardsAdventure.next by Ernesto */

item laser{(string color = "Green", int damage = 10)}
item PDP_8{(int memory = 10)}
item not_windows_computer {(string OS = "Not-Windows")}
item fish_sandwich {(int portion_left = 100, string taste = "yummy")}

character prof_edwards {(int life = 10000, int awesomeness = 300, string slogan = "Add
another layer

of indirection!"), ()}

item dragon_book {(int damage = 5)}
character prof_aho{(int life = 20, int awesomeness = 20, string slogan = "I believe thats
on page 42

of the Dragon Book!"), (dragon_book)}


character godzilla{(int life = 100), ()}
character ronald{(),()}

character plt_mob {(int life = 3, int damage = 6), ()}


character bill_gates {(int life = 2, string money = "$$$54 billion$$$", string slogan =
"It just

works"), ()}


location dragon_cave {(int depth = 500), (), (prof_edwards, prof_aho)}
location ucb {(), (not_windows_computer), (bill_gates, plt_mob)}
location mcdonalds {(), (fish_sandwich), (ronald)}
location taiwan {(), (PDP_8), ()}

int itemNum = 0;
int enemy;
int intro = 0;

location columbia{(), (), (prof_edwards)}

start columbia end(prof_edwards.life < 0 or itemNum == 4) {
      if intro == 0 then{
            output "Professor Edwards has woken up this morning from a dream!";
            output "In the dream a GCD function tells him to go forth and create the
greatest

compiler ever created!";
            output "It is a sign and he knows he must do it. He will go forth and create
the most

premiums compiler eeeeeeever";
            output "There is a problem, however...";
            output "His items of compiler power have been stolen. He must go forth and
find them";
            intro = 1;
      }
      else ;

      enemy = 0;
      output "You are in columbia";
      output "Where to next?";

      choose (dcave, "The dragon cave", "d")
               (tai, "Taiwan", "t")
               (u, " UC Berkeley", "u")
               (mcd, " McDonalds", "m")
      {

            when dcave {
                        output "Off to the Dragon Cave to battle Professor Aho";
```

```
            } next dragon_cave

            when tai {
                        output "Off to Taiwan in search of the powerful PDP-8";
            } next taiwan

            when u {
                        output "Off to UC Berkeley! ... for some reason.";
            } next ucb

            when mcd {
                        output "Off to McDonalds for a fish burger";
            } next mcdonalds
      }
}



int moment = 0;

start dragon_cave end (prof_edwards.life <= 0  or itemNum == 4) {
      if(moment == 0) then {
            output "You are in Prof. Aho's dragon cave!!";
            output "Aho is ready to fight you.. You must fight him to get his dragon
book!!";
            output "... or maybe not?...";
            moment = 1;
      }
      else ;
      choose (fight, "fight", "f")
            (runaway, "run away", "r")
            (ask, " ask nicely", "a") {

            when fight{
                  [? prob 70 {
                        output "You say: ";
                        output "behold Aho, my shiny laser pointer";
                        prof_aho.life = prof_aho.life - laser.damage;
                        output "You caused Prof. Aho a damage of:";
                        output laser.damage;

                        if (prof_aho.life <= 1) then{
                              output "You have defeated Aho... well done!!";
                              drop prof_aho.dragon_book;
                        }
                        else ;
                  }
                  prob 20 {
                        output "As he hits you Prof. Aho lets you know:";
                        output prof_aho.slogan;
                        prof_edwards.life = prof_edwards.life - dragon_book.damage;
                        output "Prof Aho hurt you this much:";
                        output dragon_book.damage;
                  }
                  prob 10{
                        output "wuah wuah wuah";
                        output "Aho surrenders.";
                        output "YOu have won the dragon book";
                        drop prof_aho.dragon_book;

                  }?]

                  if(exists dragon_cave.dragon_book) then{
                        output "In victory you say your famous slogan:";
                        output prof_edwards.slogan;
                        output "Would you like to pick up the dragon book?";
                        choose(grabbook, "grab", "g")
                              (notgrab, "not grab", "n"){

                              when grabbook { output "You have the dragon book!!";
                                    grab prof_edwards.dragon_book;
                                    if (itemNum == 3) then{
```

```
                                    output "You did it!!";
                                    output "You have collected al four items";
                                    output "With them you build the most premiums

compiler eveeeeer!!";
                              }
                              else ;

                              itemNum = itemNum + 1;
                              output "Lets go back to columbia";
                        } next columbia

                        when notgrab {
                              output "You dont have the dragon book!!!";
                              grab prof_aho.dragon_book;
                        }
                        next dragon_cave
                  }
            }
            else ;
      } next dragon_cave

      when runaway {
            output "You run away to columbia.. aaaawww";
            moment = 0;

      }next columbia


      when ask {
            if(exists prof_aho.dragon_book) then{
                  drop prof_aho.dragon_book;
                  grab prof_edwards.dragon_book;
                  output "He actually just gave you the dragon book!";
                  output "... after 3000 hours of telling you about Awk...";

                  if (itemNum == 3) then{
                        output "You did it!!";
                        output "You have collected all four items";
                        output "With them you build the most premiums compiler

eveeeeer!!";
                  }
                  else ;

                  itemNum = itemNum + 1;
                  output "Anyway, time to go back to columbia and choose the next

location";
            }
            else output "The professor does not have the dragon book!";

      }next columbia
}



}

int attack;

start ucb end (prof_edwards.life <= 0 or itemNum == 4) {
      if (bill_gates.life <= 0 or plt_mob.life <= 0) then {
            if (exists ucb.not_windows_computer) then {
                  output "You have won!";
                  output "You have received the not-windows computer!";
                  grab prof_edwards.not_windows_computer;
                  if (itemNum == 3) then{
                        output "You did it!!";
                        output "You have collected all four items";
                        output "With them you build the most premiums compiler
eveeeeer!!";
```

```
                }
                else ;
                itemNum = itemNum + 1;
                output "Time to return to Columbia";
                      choose (col, "Columbia", "c"){
                            when col ;
                            next columbia
                      }
          }
          else{
                output "You have what you needed from here";
                output "Time to return to Columbia";
                choose (col, "Columbia", "c"){
                      when col{

                      } next columbia
                }
          }
    }
    else ;


    if(enemy == 0) then{
          [? prob 50 enemy = 1;
          prob 50 enemy = 2;
          ?]
    }
    else if (enemy == 1) then{

          output "Get ready to fight Bill Gates!!";
          [? prob 40 {
                attack = 1;
                output "Bill throws Microsoft Office at your head!";
          }
          prob 30 {
                 attack = 2;
                output "Bill throws Windows 7 at your feet!";
          }
          prob 30 {
                attack = 3;
                output "Bill is trying to tell you the new features of c#!";
          }

          ?]

          choose(duck, "duck", "d")
                (jump, "jump", "j")
                (cover, "cover your ears", "c")
                {
                when duck {
                      if (attack == 1) then {
                            output "You escaped the horror!";
                            output "Gates is hyperventilating!";
                            bill_gates.life = bill_gates.life - 1;
                      }
                      else if (attack == 2) then {
                            output "You got hit by windows.. you will never function

correctly again!";
                            output "You loose 5 life points!!";
                            prof_edwards.life = prof_edwards.life - 5;
                      }
                      else {
                            output "You heard it all... You are utterly confused!";
                            output "You loose 5 life points!!";
                            prof_edwards.life = prof_edwards.life - 5;
                      }
                } next  ucb
                when jump {
                      if (attack == 2) then {
                            output "You escaped that which makes many stumble and crack
```

```
their heards!";
                              output "Gates is sweating so much!";
                              bill_gates.life = bill_gates.life - 1;
                        }
                        else if (attack == 1) then {
                              output "You got hit by office square in the face... just
like

every user!!";
                              output "You loose 5 life points!!";
                              prof_edwards.life = prof_edwards.life - 5;

                        }
                        else {
                              output "You heard it all... You are utterly confused!";
                              output "You loose 5 life points!!";
                              prof_edwards.life = prof_edwards.life - 5;
                        }
                  } next ucb
                  when cover {
                        if (attack == 3) then {
                              output "Good! You would have never been able to make a good

compiler again!";
                              output "Gates looks dizzy!";
                              bill_gates.life = bill_gates.life - 1;
                        }
                        else if (attack == 1) then {
                              output "You got hit by office square in the face... just
like

every user!!";
                              output "You loose 5 life points!!";
                              prof_edwards.life = prof_edwards.life - 5;

                        }
                        else{
                              output "You got hit by windows.. you will never function

correctly again!";
                              output "You loose 5 life points!!";
                              prof_edwards.life = prof_edwards.life - 5;
                        }

                  } next ucb
            }


      }
      else{

            output "Get ready to fight a huge PLT mob!!";

            [? prob 40 {
                  attack = 1;
                  output "Someone throws a C Reference Manual at you!";
            }
            prob 30 {
                   attack = 2;
                  output "someone is trying to throw the Ocaml camel at your feet!!";
            }
            prob 30 {
                  attack = 3;
                  output "They are all screaming about a new language that is scanned

backwards!";
            }

            ?]
            choose(duck, "duck", "d")
                  (jump, "jump", "j")
                  (cover, "cover your ears", "c")
```

```
                  {
                  when duck {
                        if (attack == 1) then {
                              output "You escaped the horror!";
                              output "They are getting tired!";
                              plt_mob.life = plt_mob.life - 1;
                        }
                        else if (attack == 2) then {
                              output "You got hit by the camel.. you appreciate O'Caml a

little less!";
                              output "You loose 5 life points!!";
                              prof_edwards.life = prof_edwards.life - plt_mob.damage;
                        }
                        else {
                              output "You heard it all... You are utterly confused!";
                              output "You loose 5 life points!!";
                              prof_edwards.life = prof_edwards.life - plt_mob.damage;
                        }
                  } next  ucb
                  when jump {
                        if (attack == 2) then {
                              output "You escaped that which makes many stumble and crack

their heards!";
                              output "The mob is getting bored!";
                              plt_mob.life = plt_mob.life - 1;
                        }
                        else if (attack == 1) then {
                              output "You got hit by the manual square in the face...
your

brain is now a big null pointer!!";
                              output "You loose 5 life points!!";
                              prof_edwards.life = prof_edwards.life - plt_mob.damage;

                        }
                        else {
                              output "You heard it all... You are utterly confused!";
                              output "You loose 5 life points!!";
                              prof_edwards.life = prof_edwards.life - plt_mob.damage;
                        }
                  } next ucb
                  when cover {
                        if (attack == 3) then {
                              output "Good! You would have never been able to make a good

compiler again!";
                              output "Every head in that mob hurts!";
                              bill_gates.life = bill_gates.life - 1;
                        }
                        else if (attack == 1) then {
                              output "You got hit by the manual square in the face...
your

brain is now a big null pointer!!";
                              output "You loose 5 life points!!";
                              prof_edwards.life = prof_edwards.life - plt_mob.damage;

                        }
                        else{
                              output "You got hit by the camel.. you appreciate O'Caml a

little less!";
                              output "You loose 5 life points!!";
                              prof_edwards.life = prof_edwards.life - plt_mob.damage;
                        }

                  } next ucb
            }
      }
```

```
}

start mcdonalds end  (prof_edwards.life <= 0 or itemNum == 4) {
      if (exists mcdonalds.fish_sandwich) then{

            output "Ronald gives you a riddle!";
            output  "What Burger was featured in the greatest advertising campaign
ever?";
            choose (fish, " fish sandwitch", "f")
                  (burger, " Big Mac", "b")
                  (caesar, " Caesar Salad", "c"){

                  when fish {
                        output "Yes, you do believe in magic!";
                        output "YOu have earned a  fish Burger";
                        output "Lets go back to Columbia to keep on with the
adventure!!";
                        grab prof_edwards.fish_sandwich;
                        if (itemNum == 3) then{
                              output "You did it!!";
                              output "You have collected all four items";
                              output "With them you build the most premiums compiler

eveeeeer!!";
                        }
                        else ;
                        itemNum = itemNum + 1;
                  } next columbia

                  when burger
                        output "no... come on... you were there!!";
                  next mcdonalds

                  when caesar
                        output "Thats not even a burger!!";
                  next mcdonalds
            }
      }

      else{
            output "you have collected what you needed here";
            choose(col, "Columbia", "c"){
                  when col{} next columbia
            }
      }

}


int narrated = 0;

start taiwan end (prof_edwards.life <= 0 or itemNum == 4) {
      if (godzilla.life <= 0) then{
            output "...oh no wait, you beat Godzilla!! holy shhh... wow!";
            output "You have received the PDP-8!!";
            if( not exists prof_edwards.PDP_8) then{
                  grab prof_edwards.PDP_8;
                  if (itemNum == 3) then{
                        output "You did it!!";
                        output "You have collected all four items";
                        output "With them you build the most premiums compiler
eveeeeer!!";
                  }
                  else ;
            itemNum = itemNum + 1;
            }
            else ;
      }
      else ;
```

```
      if (exists taiwan.PDP_8) then{
            if (narrated == 0) then{
                  output "You have traveled to Taiwan to get the biggest PDP-8 in the
world";
                  output "You need it for its humongous stack!!";
                  output "The problem is, it belongs to Godzilla!!";
                  narrated = 1;
            }
            else ;

            output "Godzilla is about to step on you!!";

            choose (shoot,"shooting your laser pointer at his foot", "s")
                  (die, "dying squished between its toes", "d")
                  (tickle, "tickling the monster's foot", "t"){

                  when shoot {
                        output "You hit the monster in the foot";
                        godzilla.life = godzilla.life - laser.damage;
                        output " You did 10 points damage!!";
                        output "Keep fighting!!";

                  }next taiwan

                  when die{
                        output "You have died between its green toes";
                        prof_edwards.life = 0;
                  }next columbia

                  when tickle{
                        output "the monster fell back laughing";
                        output "You have done 50 damage!!";
                        godzilla.life = godzilla.life - 50;
                  }next taiwan
            }
      }
      else{
            output "You have the calculator. Time to return to columbia";
            choose(col, "Columbia", "c"){
                  when col; next columbia
            }
      }



}
```

```
//Next.java file generated for ProfEdwardsAdventure.next
import java.util.*;

public class Next {
   enum Type {INT, STRING, CHARACTER, ITEM, LOCATION}

   static Random r = new Random();
   Object dummy;
   String currentLocation;
   Map<String, Location> locations = new HashMap<String, Location>();
   Map<String, Character> characters = new HashMap<String, Character>();
   Map<String, Item> items = new HashMap<String, Item>();
   Map<String, Type> types = new HashMap<String, Type>();

   public static void main(String[] args) {
    (new Next()).play();
   }

   public int boolToInt(boolean value) {
        if(value) {
            return 1;
        }
        else {
            return 0;
        }
   }

   public String entitySetString(String key1, Type type1, String key2, String value) {
    boolean valueSet = false;
      if(type1 == Type.LOCATION) {
            Location loc = locations.get(key1);
            if(loc != null) {
                  loc.strAttrs.put(key2, value);
                  valueSet = true;
            }
      }
      else if(type1 == Type.CHARACTER) {
            Character character = characters.get(key1);
            if(character != null) {
                  character.strAttrs.put(key2, value);
                  valueSet = true;
            }
      }
      else if(type1 == Type.ITEM) {
            Item item = items.get(key1);
            if(item != null) {
                  item.strAttrs.put(key2, value);
                  valueSet = true;
            }
      }

      if(!valueSet) {
            throw new RuntimeException();
      }

      return value;
   }

   public int entitySetInt(String key1, Type type1, String key2, int value) {
      boolean foundReturnValue = false;

      if(type1 == Type.LOCATION) {
            Location loc = locations.get(key1);
            if(loc != null) {
            loc.intAttrs.put(key2, value);
                  foundReturnValue = true;
            }
      }
      else if(type1 == Type.CHARACTER) {
            Character character = characters.get(key1);
            if(character != null) {
                  character.intAttrs.put(key2, value);
```

```
                foundReturnValue = true;
          }
    }
    else if(type1 == Type.ITEM) {
          Item item = items.get(key1);
          if(item != null) {
                item.intAttrs.put(key2, value);
                foundReturnValue = true;
          }
    }

    if(foundReturnValue == false) {
          throw new RuntimeException();
    }

    return value;
}

    public boolean isTrue(Object object) {
        if(object instanceof String) {
           if(((String)object).isEmpty()) {
               return false;
           }
        }
        else if(object instanceof Integer) {
           if((Integer)object == 0) {
               return false;
           }
        }
        else {
            if(object == null) {
                return false;
            }
        }

        return true;
    }

  public void killFunction(String varName){
    if (characters.containsKey(varName)){
          characters.remove(varName);
          for (String key: locations.keySet()){
                      locations.get(key).hideCharacter(varName);
          }
    }
    else if (items.containsKey(varName)){
          items.remove(varName);
          for (String key:locations.keySet()){
                locations.get(key).removeItem(varName);
                }
          for (String key : characters.keySet()){
                characters.get(key).removeItem(varName);
          }
    }
  }

  public int entityHasInt(String key1, Type type1, String key2) {
    int returnValue = 0;
    boolean foundReturnValue = false;

    if(type1 == Type.LOCATION) {
          Location loc = locations.get(key1);
          if(loc != null) {
                returnValue = loc.intAttrs.get(key2);
                foundReturnValue = true;
          }
    }
    else if(type1 == Type.CHARACTER) {
          Character character = characters.get(key1);
          if(character != null) {
                returnValue = character.intAttrs.get(key2);
                foundReturnValue = true;
```

```
		}
	}
	else if(type1 == Type.ITEM) {
		Item item = items.get(key1);
		if(item != null) {
			returnValue = item.intAttrs.get(key2);
			foundReturnValue = true;
		}
	}

	if(foundReturnValue == false) {
		throw new RuntimeException();
	}

	return returnValue;
}

public String entityHasString(String key1, Type type1, String key2) {
	String returnValue = null;
	if(type1 == Type.LOCATION) {
		Location loc = locations.get(key1);
		if(loc != null) {
			returnValue = loc.strAttrs.get(key2);
		}
	}
	else if(type1 == Type.CHARACTER) {
		Character character = characters.get(key1);
		if(character != null) {
			returnValue = character.strAttrs.get(key2);
		}
	}
	else if(type1 == Type.ITEM) {
		Item item = items.get(key1);
		if(item != null) {
			returnValue = item.strAttrs.get(key2);
		}
	}

	if(returnValue == null) {
		throw new RuntimeException();
	}

	return returnValue;
}

public Item entityHasItem(String key1, Type type1, String key2) {
	Item returnValue = null;
	if(type1 == Type.LOCATION) {
		Location loc = locations.get(key1);
		if(loc != null) {
			if(loc.items.contains(key2)) {
				returnValue = items.get(key2);
			}
		}
	}
	else if(type1 == Type.CHARACTER) {
		Character character = characters.get(key1);
		if(character != null) {
			if(character.items.contains(key2)) {
				returnValue = items.get(key2);
			}
		}
	}

	if(returnValue == null) {
		throw new RuntimeException();
	}

	return returnValue;
}

public Character entityHasCharacter(String key1, Type type1, String key2) {
```

```
      Character returnValue = null;
      if(type1 == Type.LOCATION) {
            Location loc = locations.get(key1);
            if(loc != null) {
                  if(loc.characters.contains(key2)) {
                        returnValue = characters.get(key2);
                  }
            }
      }

      if(returnValue == null) {
            throw new RuntimeException();
      }

      return returnValue;
    }

    public int entityExistsItem(String key1, Type type1, String key2) {
      Object returnValue = null;
      if(type1 == Type.LOCATION) {
            Location loc = locations.get(key1);
            if(loc != null) {
                  if(loc.items.contains(key2)) {
                        returnValue = items.get(key2);
                  }
            }
      }
      else if(type1 == Type.CHARACTER) {
            Character character = characters.get(key1);
            if(character != null) {
                  if(character.items.contains(key2)) {
                        returnValue = items.get(key2);
                  }
            }
      }

      if(returnValue == null) {
            return 0;
      }

      return 1;
    }

    public int entityExistsCharacter(String key1, Type type1, String key2) {
      Object returnValue = null;
      if(type1 == Type.LOCATION) {
            Location loc = locations.get(key1);
            if(loc != null) {
                  if(loc.characters.contains(key2)) {
                        returnValue = characters.get(key2);
                  }
            }
      }

      if(returnValue == null) {
            return 0;
      }

      return 1;
    }

   public void endGame() {
      System.exit(0);
   }

   public void play() {
items.put("_laser",_laser);
types.put("_laser", Type.ITEM);
_laser.addStrAttr("_color","Green");
types.put("_color", Type.STRING);
_laser.addIntAttr("_damage", (10));
types.put("_damage", Type.INT);
```

```
items.put("_PDP_8",_PDP_8);
types.put("_PDP_8", Type.ITEM);
_PDP_8.addIntAttr("_memory", (10));
types.put("_memory", Type.INT);
items.put("_not_windows_computer",_not_windows_computer);
types.put

("_not_windows_computer", Type.ITEM);
_not_windows_computer.addStrAttr("_OS","Not-Windows");
types.put

("_OS", Type.STRING);
items.put("_fish_sandwich",_fish_sandwich);
types.put("_fish_sandwich",

Type.ITEM);
_fish_sandwich.addIntAttr("_portion_left", (100));
types.put("_portion_left", Type.INT);
_fish_sandwich.addStrAttr("_taste","yummy");
types.put("_taste", Type.STRING);
characters.put

("_prof_edwards",_prof_edwards);
types.put("_prof_edwards", Type.CHARACTER);
_prof_edwards.addIntAttr

("_life", (10000));
types.put("_life", Type.INT);
_prof_edwards.addIntAttr("_awesomeness", (300));
types.put("_awesomeness", Type.INT);
_prof_edwards.addStrAttr("_slogan","Add another layer of

indirection!");
types.put("_slogan", Type.STRING);
items.put("_dragon_book",_dragon_book);
types.put

("_dragon_book", Type.ITEM);
_dragon_book.addIntAttr("_damage", (5));
types.put("_damage", Type.INT);
characters.put("_prof_aho",_prof_aho);
types.put("_prof_aho", Type.CHARACTER);
_prof_aho.addIntAttr

("_life", (20));
types.put("_life", Type.INT);
_prof_aho.addIntAttr("_awesomeness", (20));
types.put

("_awesomeness", Type.INT);
_prof_aho.addStrAttr("_slogan","I believe thats on page 42 of the Dragon

Book!");
types.put("_slogan", Type.STRING);
_prof_aho.addItem("_dragon_book");
characters.put

("_godzilla",_godzilla);
types.put("_godzilla", Type.CHARACTER);
_godzilla.addIntAttr("_life", (100));
types.put("_life", Type.INT);
characters.put("_ronald",_ronald);
types.put("_ronald", Type.CHARACTER);
characters.put("_plt_mob",_plt_mob);
types.put("_plt_mob", Type.CHARACTER);
_plt_mob.addIntAttr("_life",

(3));
types.put("_life", Type.INT);
_plt_mob.addIntAttr("_damage", (6));
types.put("_damage", Type.INT);
characters.put("_bill_gates",_bill_gates);
```

```
types.put("_bill_gates", Type.CHARACTER);
_bill_gates.addIntAttr("_life", (2));
types.put("_life", Type.INT);
_bill_gates.addStrAttr

("_money","$$$54 billion$$$");
types.put("_money", Type.STRING);
_bill_gates.addStrAttr("_slogan","It

just works");
types.put("_slogan", Type.STRING);
locations.put("_dragon_cave",_dragon_cave);
types.put

("_dragon_cave", Type.LOCATION);
_dragon_cave.addIntAttr("_depth", (500));
types.put("_depth",

Type.INT);
_dragon_cave.showCharacter("_prof_edwards");
_dragon_cave.showCharacter("_prof_aho");
locations.put("_ucb",_ucb);
types.put("_ucb", Type.LOCATION);
_ucb.addItem("_not_windows_computer");
_ucb.showCharacter("_bill_gates");
_ucb.showCharacter("_plt_mob");
locations.put

("_mcdonalds",_mcdonalds);
types.put("_mcdonalds", Type.LOCATION);
_mcdonalds.addItem("_fish_sandwich");
_mcdonalds.showCharacter("_ronald");
locations.put("_taiwan",_taiwan);
types.put("_taiwan",

Type.LOCATION);
_taiwan.addItem("_PDP_8");
_itemNum = (0);
_intro = (0);
locations.put

("_columbia",_columbia);
types.put("_columbia", Type.LOCATION);
_columbia.showCharacter

("_prof_edwards");
//Location function call
_columbia();
_moment = (0);
//Location function call
_dragon_cave();
//Location function call
_ucb();
//Location function call
_mcdonalds();
_narrated = (0);
//Location function call
_taiwan();
   endGame();
   }
//itemdec
Item _laser = new Item();
//itemdec
Item _PDP_8 = new Item();
//itemdec
Item

_not_windows_computer = new Item();
//itemdec
Item _fish_sandwich = new Item();
//charadec
```

```
Character

_prof_edwards = new Character();
//itemdec
Item _dragon_book = new Item();
//charadec
Character _prof_aho

= new Character();
//charadec
Character _godzilla = new Character();
//charadec
Character _ronald = new

Character();
//charadec
Character _plt_mob = new Character();
//charadec
Character _bill_gates = new

Character();
//locdec
Location _dragon_cave = new Location();
//locdec
Location _ucb = new Location();
//locdec
Location _mcdonalds = new Location();
//locdec
Location _taiwan = new Location();
//intdecinit
int

_itemNum;
//intdec
int _enemy;
//intdecinit
int _intro;
//locdec
Location _columbia = new Location();
//start

funtion
public void _columbia() {
currentLocation = "_columbia";
while (!((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") < (0)) || _itemNum == (4))){
{
if(_intro == (0)) {
{
System.out.println(""+

("Professor Edwards has woken up this morning from a dream!"));
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") < (0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("In the dream a

GCD function tells him to go forth and create the greatest compiler ever created!"));
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") < (0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("It is a sign and he knows he must do it. He will go forth and
create the most

premiums compiler eeeeeeever"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") < (0)) ||

_itemNum == (4))
endGame();
```

```
System.out.println(""+ ("There is a problem, however..."));
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") < (0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("His items of compiler power have been stolen. He must go forth
and find

them"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") < (0)) || _itemNum == (4))
endGame

();
dummy = (_intro = (1));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") < (0)) ||

_itemNum == (4))
endGame();
}
}
else {
//Empty stmt
}
dummy = (_enemy = (0));
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") < (0)) || _itemNum == (4))
endGame();
System.out.println(""+

("You are in columbia"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") < (0)) ||

_itemNum == (4))
endGame();
System.out.println(""+ ("Where to next?"));
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") < (0)) || _itemNum == (4))
endGame();
Map<String,String>

keysToActionName16 = new HashMap<String, String>();
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") < (0)) || _itemNum == (4))
endGame();
Map<String, String> actionNameToOutput16 = new

HashMap<String, String>();
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") < (0)) ||

_itemNum == (4))
endGame();
System.out.println("CHOOSE AN ACTION:");
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") < (0)) || _itemNum == (4))
endGame();
keysToActionName16.put("d", "_dcave");
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") < (0)) || _itemNum == (4))
endGame();
actionNameToOutput16.put("_dcave", "The dragon cave");
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") < (0)) || _itemNum == (4))
endGame();
System.out.println("Type d for The dragon

cave");
```

```
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") < (0)) || _itemNum == (4))
endGame

();
keysToActionName16.put("t", "_tai");
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <

(0)) || _itemNum == (4))
endGame();
actionNameToOutput16.put("_tai", "Taiwan");
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") < (0)) || _itemNum == (4))
endGame();
System.out.println

("Type t for Taiwan");
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") < (0)) || _itemNum

== (4))
endGame();
keysToActionName16.put("u", "_u");
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") < (0)) || _itemNum == (4))
endGame();
actionNameToOutput16.put("_u", " UC Berkeley");
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") < (0)) || _itemNum == (4))
endGame();
System.out.println("Type u for  UC Berkeley");
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") < (0)) || _itemNum == (4))
endGame();
keysToActionName16.put("m", "_mcd");
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") < (0)) || _itemNum == (4))
endGame();
actionNameToOutput16.put("_mcd", " McDonalds");
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") < (0)) || _itemNum == (4))
endGame();
System.out.println("Type m for  McDonalds");
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") < (0)) || _itemNum == (4))
endGame();
Scanner

in16 = new Scanner(System.in);
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") < (0)) ||

_itemNum == (4))
endGame();
String input16 = in16.nextLine();
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") < (0)) || _itemNum == (4))
endGame();
while(!keysToActionName16.containsKey

(input16)) {
System.out.println("Invalid input, try again");
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") < (0)) || _itemNum == (4))
endGame();
input16 = in16.nextLine();
```

```
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") < (0)) || _itemNum == (4))
endGame();
}
System.out.println("You typed " + input16);
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life")

< (0)) || _itemNum == (4))
endGame();
String action16 = keysToActionName16.get(input16);
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") < (0)) || _itemNum == (4))
endGame();
if

(action16.equals("_dcave")) {
{
System.out.println(""+ ("Off to the Dragon Cave to battle Professor

Aho"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") < (0)) || _itemNum == (4))
endGame

();
}
_dragon_cave();
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") < (0)) || _itemNum ==

(4))
endGame();
}
if(action16.equals("_tai")) {
{
System.out.println(""+ ("Off to Taiwan in search of the

powerful PDP-8"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") < (0)) || _itemNum ==

(4))
endGame();
}
_taiwan();
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") < (0)) ||

_itemNum == (4))
endGame();
}
if(action16.equals("_u")) {
{
System.out.println(""+ ("Off to UC Berkeley!

... for some reason."));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") < (0)) || _itemNum

== (4))
endGame();
}
_ucb();
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") < (0)) ||

_itemNum == (4))
endGame();
}
if(action16.equals("_mcd")) {
{
System.out.println(""+ ("Off to McDonalds for

a fish burger"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") < (0)) || _itemNum ==
```

```
(4))
endGame();
}
_mcdonalds();
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") < (0)) ||

_itemNum == (4))
endGame();
}
}
}
endGame();
}
//intdecinit
int _moment;
//start funtion
public void

_dragon_cave() {
currentLocation = "_dragon_cave";
while (!((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))){
{
if(_moment == (0)) {
{
System.out.println(""+

("You are in Prof. Aho's dragon cave!!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life")

<= (0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("Aho is ready to fight you.. You must

fight him to get his dragon book!!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <=

(0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("... or maybe not?..."));
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
dummy = (_moment =

(1));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
}
}
else {
//Empty stmt
}
Map<String,String> keysToActionName11 = new HashMap<String, String>();
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
Map<String, String> actionNameToOutput11 = new HashMap<String, String>();
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println

("CHOOSE AN ACTION:");
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum

== (4))
endGame();
```

```
keysToActionName11.put("f", "_fight");
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
actionNameToOutput11.put("_fight",

"fight");
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println("Type f for fight");
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") <= (0)) || _itemNum == (4))
endGame();
keysToActionName11.put("r", "_runaway");
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
actionNameToOutput11.put("_runaway", "run away");
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println("Type r for run away");
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
keysToActionName11.put("a", "_ask");
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <=

(0)) || _itemNum == (4))
endGame();
actionNameToOutput11.put("_ask", " ask nicely");
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println

("Type a for  ask nicely");
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) ||

_itemNum == (4))
endGame();
Scanner in11 = new Scanner(System.in);
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
String input11 = in11.nextLine();
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
while(!

keysToActionName11.containsKey(input11)) {
System.out.println("Invalid input, try again");
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
input11

= in11.nextLine();
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum ==

(4))
endGame();
```

```
}
System.out.println("You typed " + input11);
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
String action11 =

keysToActionName11.get(input11);
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) ||

_itemNum == (4))
endGame();
if(action11.equals("_fight")) {
{
int num = r.nextInt(100);
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
if(num >= 0 && num <

70) {
{
System.out.println(""+ ("You say: "));
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("behold Aho, my shiny laser

pointer"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
dummy = (entitySetInt("_prof_aho", Type.CHARACTER, "_life", (entityHasInt("_prof_aho",

Type.CHARACTER, "_life") - entityHasInt("_laser", Type.ITEM, "_damage"))));
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println(""+

("You caused Prof. Aho a damage of:"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <=

(0)) || _itemNum == (4))
endGame();
System.out.println(""+ (entityHasInt("_laser", Type.ITEM,

"_damage")));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
if((entityHasInt("_prof_aho", Type.CHARACTER, "_life") <= (1))) {
{
System.out.println(""+

("You have defeated Aho... well done!!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life")

<= (0)) || _itemNum == (4))
endGame();
_prof_aho.removeItem ("_dragon_book", currentLocation,

locations);
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
}
}
else {
//Empty stmt
}
```

```
}
}
if(num >= 70 && num < 90) {
{
System.out.println(""+ ("As he hits you

Prof. Aho lets you know:"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) ||

_itemNum == (4))
endGame();
System.out.println(""+ (entityHasString("_prof_aho", Type.CHARACTER,

"_slogan")));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
dummy = (entitySetInt("_prof_edwards", Type.CHARACTER, "_life", (entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") - entityHasInt("_dragon_book", Type.ITEM,
"_damage"))));
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("Prof Aho hurt you this much:"));
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println(""+ (entityHasInt

("_dragon_book", Type.ITEM, "_damage")));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life")

<= (0)) || _itemNum == (4))
endGame();
}
}
if(num >= 90 && num < 100) {
{
System.out.println(""+ ("wuah wuah

wuah"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("Aho surrenders."));
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("YOu have won

the dragon book"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum ==

(4))
endGame();
_prof_aho.removeItem ("_dragon_book", currentLocation, locations);
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
}
}
if(isTrue

(entityExistsItem("_dragon_cave", Type.LOCATION, "_dragon_book"))) {
{
System.out.println(""+ ("In

victory you say your famous slogan:"));
```

```
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <=

(0)) || _itemNum == (4))
endGame();
System.out.println(""+ (entityHasString("_prof_edwards",

Type.CHARACTER, "_slogan")));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) ||

_itemNum == (4))
endGame();
System.out.println(""+ ("Would you like to pick up the dragon book?"));
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
Map<String,String> keysToActionName3 = new HashMap<String, String>();
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
Map<String, String>

actionNameToOutput3 = new HashMap<String, String>();
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println("CHOOSE AN ACTION:");
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
keysToActionName3.put("g", "_grabbook");
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <=

(0)) || _itemNum == (4))
endGame();
actionNameToOutput3.put("_grabbook", "grab");
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println

("Type g for grab");
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum ==

(4))
endGame();
keysToActionName3.put("n", "_notgrab");
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
actionNameToOutput3.put("_notgrab", "not

grab");
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame

();
System.out.println("Type n for not grab");
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") <= (0)) || _itemNum == (4))
endGame();
Scanner in3 = new Scanner(System.in);
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
```

```
String input3 =

in3.nextLine();
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
while(!keysToActionName3.containsKey(input3)) {
System.out.println("Invalid input, try

again");
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
input3 = in3.nextLine();
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0))

|| _itemNum == (4))
endGame();
}
System.out.println("You typed " + input3);
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
String action3 =

keysToActionName3.get(input3);
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) ||

_itemNum == (4))
endGame();
if(action3.equals("_grabbook")) {
{
System.out.println(""+ ("You have the

dragon book!!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum ==

(4))
endGame();
_prof_edwards.addItem ("_dragon_book", currentLocation, locations);
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
if(_itemNum == (3)) {
{
System.out.println(""+ ("You did it!!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life")

<= (0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("You have collected al four items"));
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("With them you build the most premiums compiler eveeeeer!!"));
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
}
}
else

{
//Empty stmt
}
dummy = (_itemNum = (_itemNum + (1)));
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") <= (0)) || _itemNum == (4))
endGame();
```

```
System.out.println(""+ ("Lets go back to columbia"));
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
}
_columbia();
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
}
if(action3.equals("_notgrab")) {
{
System.out.println(""+ ("You dont have the dragon

book!!!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
_prof_aho.addItem ("_dragon_book", currentLocation, locations);
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
}
_dragon_cave();
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
}
}
}
else

{
//Empty stmt
}
}
_dragon_cave();
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) ||

_itemNum == (4))
endGame();
}
if(action11.equals("_runaway")) {
{
System.out.println(""+ ("You run away to

columbia.. aaaawww"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum

== (4))
endGame();
dummy = (_moment = (0));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life")

<= (0)) || _itemNum == (4))
endGame();
}
_columbia();
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") <= (0)) || _itemNum == (4))
endGame();
}
if(action11.equals("_ask")) {
{
if(isTrue

(entityExistsItem("_prof_aho", Type.CHARACTER, "_dragon_book"))) {
{
_prof_aho.removeItem
```

```
("_dragon_book", currentLocation, locations);
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") <= (0)) || _itemNum == (4))
endGame();
_prof_edwards.addItem ("_dragon_book", currentLocation,

locations);
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("He actually just gave you the dragon book!"));
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println(""+

("... after 3000 hours of telling you about Awk..."));
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
if(_itemNum == (3)) {
{
System.out.println(""+ ("You did it!!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life")

<= (0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("You have collected all four items"));
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("With them you build the most premiums compiler eveeeeer!!"));
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
}
}
else

{
//Empty stmt
}
dummy = (_itemNum = (_itemNum + (1)));
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("Anyway, time to go back to

columbia and choose the next location"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life")

<= (0)) || _itemNum == (4))
endGame();
}
}
else {
System.out.println(""+ ("The professor does not have the

dragon book!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum ==

(4))
endGame();
}
}
_columbia();
```

```
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) ||

_itemNum == (4))
endGame();
}
}
}
endGame();
}
//intdec
int _attack;
//start funtion
public void _ucb() {
currentLocation = "_ucb";
while (!((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) ||

_itemNum == (4))){
{
if((entityHasInt("_bill_gates", Type.CHARACTER, "_life") <= (0)) || (entityHasInt

("_plt_mob", Type.CHARACTER, "_life") <= (0))) {
{
if(isTrue(entityExistsItem("_ucb", Type.LOCATION,

"_not_windows_computer"))) {
{
System.out.println(""+ ("You have won!"));
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println(""+

("You have received the not-windows computer!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") <= (0)) || _itemNum == (4))
endGame();
_prof_edwards.addItem ("_not_windows_computer",

currentLocation, locations);
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) ||

_itemNum == (4))
endGame();
if(_itemNum == (3)) {
{
System.out.println(""+ ("You did it!!"));
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("You have collected all four items"));
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("With them you

build the most premiums compiler eveeeeer!!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") <= (0)) || _itemNum == (4))
endGame();
}
}
else {
//Empty stmt
}
dummy = (_itemNum = (_itemNum +

(1)));
```

```
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame

();
System.out.println(""+ ("Time to return to Columbia"));
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
Map<String,String> keysToActionName15 =

new HashMap<String, String>();
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) ||

_itemNum == (4))
endGame();
Map<String, String> actionNameToOutput15 = new HashMap<String, String>();
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println("CHOOSE AN ACTION:");
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <=

(0)) || _itemNum == (4))
endGame();
keysToActionName15.put("c", "_col");
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
actionNameToOutput15.put("_col", "Columbia");
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println("Type c for Columbia");
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
Scanner

in15 = new Scanner(System.in);
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) ||

_itemNum == (4))
endGame();
String input15 = in15.nextLine();
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
while(!keysToActionName15.containsKey

(input15)) {
System.out.println("Invalid input, try again");
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
input15 = in15.nextLine();
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
}
System.out.println("You typed " + input15);
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life")

<= (0)) || _itemNum == (4))
endGame();
```

```
String action15 = keysToActionName15.get(input15);
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
if

(action15.equals("_col")) {
//Empty stmt
_columbia();
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") <= (0)) || _itemNum == (4))
endGame();
}
}
}
else {
{
System.out.println(""+ ("You have what you

needed from here"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum ==

(4))
endGame();
System.out.println(""+ ("Time to return to Columbia"));
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
Map<String,String>

keysToActionName2 = new HashMap<String, String>();
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") <= (0)) || _itemNum == (4))
endGame();
Map<String, String> actionNameToOutput2 = new

HashMap<String, String>();
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) ||

_itemNum == (4))
endGame();
System.out.println("CHOOSE AN ACTION:");
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
keysToActionName2.put("c", "_col");
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
actionNameToOutput2.put("_col", "Columbia");
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println("Type c for Columbia");
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
Scanner

in2 = new Scanner(System.in);
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) ||

_itemNum == (4))
endGame();
```

```
String input2 = in2.nextLine();
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
while(!keysToActionName2.containsKey

(input2)) {
System.out.println("Invalid input, try again");
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
input2 = in2.nextLine();
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
}
System.out.println("You typed " + input2);
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life")

<= (0)) || _itemNum == (4))
endGame();
String action2 = keysToActionName2.get(input2);
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
if(action2.equals

("_col")) {
//Empty stmt
_columbia();
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0))

|| _itemNum == (4))
endGame();
}
}
}
}
}
else {
//Empty stmt
}
if(_enemy == (0)) {
{
int num = r.nextInt(100);
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
if(num

>= 0 && num < 50) {
dummy = (_enemy = (1));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life")

<= (0)) || _itemNum == (4))
endGame();
}
if(num >= 50 && num < 100) {
dummy = (_enemy = (2));
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
}
}
}
```

```
else

{
if(_enemy == (1)) {
{
System.out.println(""+ ("Get ready to fight Bill Gates!!"));
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
int num = r.nextInt

(100);
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame

();
if(num >= 0 && num < 40) {
{
dummy = (_attack = (1));
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("Bill throws

Microsoft Office at your head!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0))

|| _itemNum == (4))
endGame();
}
}
if(num >= 40 && num < 70) {
{
dummy = (_attack = (2));
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println(""+

("Bill throws Windows 7 at your feet!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life")

<= (0)) || _itemNum == (4))
endGame();
}
}
if(num >= 70 && num < 100) {
{
dummy = (_attack = (3));
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("Bill is trying to tell you the new features of c#!"));
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
}
}
Map<String,String>

keysToActionName20 = new HashMap<String, String>();
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") <= (0)) || _itemNum == (4))
endGame();
Map<String, String> actionNameToOutput20 = new
```

```
HashMap<String, String>();
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) ||

_itemNum == (4))
endGame();
System.out.println("CHOOSE AN ACTION:");
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
keysToActionName20.put("d", "_duck");
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
actionNameToOutput20.put("_duck", "duck");
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life")

<= (0)) || _itemNum == (4))
endGame();
System.out.println("Type d for duck");
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
keysToActionName20.put("j", "_jump");
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <=

(0)) || _itemNum == (4))
endGame();
actionNameToOutput20.put("_jump", "jump");
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println

("Type j for jump");
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum ==

(4))
endGame();
keysToActionName20.put("c", "_cover");
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") <= (0)) || _itemNum == (4))
endGame();
actionNameToOutput20.put("_cover", "cover your ears");
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println("Type c for cover your ears");
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") <= (0)) || _itemNum == (4))
endGame();
Scanner in20 = new Scanner(System.in);
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
String input20 =

in20.nextLine();
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum ==

(4))
endGame();
while(!keysToActionName20.containsKey(input20)) {
System.out.println("Invalid input, try
```

```
again");
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
input20 = in20.nextLine();
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <=

(0)) || _itemNum == (4))
endGame();
}
System.out.println("You typed " + input20);
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
String action20 =

keysToActionName20.get(input20);
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) ||

_itemNum == (4))
endGame();
if(action20.equals("_duck")) {
{
if(_attack == (1)) {
{
System.out.println(""+

("You escaped the horror!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) ||

_itemNum == (4))
endGame();
System.out.println(""+ ("Gates is hyperventilating!"));
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
dummy = (entitySetInt

("_bill_gates", Type.CHARACTER, "_life", (entityHasInt("_bill_gates", Type.CHARACTER,
"_life") -

(1))));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame

();
}
}
else {
if(_attack == (2)) {
{
System.out.println(""+ ("You got hit by windows.. you will never

function correctly again!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) ||

_itemNum == (4))
endGame();
System.out.println(""+ ("You loose 5 life points!!"));
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
dummy = (entitySetInt

("_prof_edwards", Type.CHARACTER, "_life", (entityHasInt("_prof_edwards", Type.CHARACTER,
"_life") -

(5))));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
```

```
endGame

();
}
}
else {
{
System.out.println(""+ ("You heard it all... You are utterly confused!"));
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("You loose 5 life points!!"));
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
dummy = (entitySetInt("_prof_edwards",

Type.CHARACTER, "_life", (entityHasInt("_prof_edwards", Type.CHARACTER, "_life") -
(5))));
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
}
}
}
}
_ucb();
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame

();
}
if(action20.equals("_jump")) {
{
if(_attack == (2)) {
{
System.out.println(""+ ("You escaped that

which makes many stumble and crack their heards!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("Gates is sweating so much!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
dummy

= (entitySetInt("_bill_gates", Type.CHARACTER, "_life", (entityHasInt("_bill_gates",
Type.CHARACTER,

"_life") - (1))));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum ==

(4))
endGame();
}
}
else {
if(_attack == (1)) {
{
System.out.println(""+ ("You got hit by office square in

the face... just like every user!!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <=

(0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("You loose 5 life points!!"));
```

```
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
dummy =

(entitySetInt("_prof_edwards", Type.CHARACTER, "_life", (entityHasInt("_prof_edwards",
Type.CHARACTER,

"_life") - (5))));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum ==

(4))
endGame();
}
}
else {
{
System.out.println(""+ ("You heard it all... You are utterly confused!"));
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("You loose 5 life points!!"));
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
dummy = (entitySetInt("_prof_edwards",

Type.CHARACTER, "_life", (entityHasInt("_prof_edwards", Type.CHARACTER, "_life") -
(5))));
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
}
}
}
}
_ucb();
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame

();
}
if(action20.equals("_cover")) {
{
if(_attack == (3)) {
{
System.out.println(""+ ("Good! You would have

never been able to make a good compiler again!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("Gates looks dizzy!"));
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
dummy =

(entitySetInt("_bill_gates", Type.CHARACTER, "_life", (entityHasInt("_bill_gates",
Type.CHARACTER,

"_life") - (1))));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum ==

(4))
endGame();
```

```
}
}
else {
if(_attack == (1)) {
{
System.out.println(""+ ("You got hit by office square in

the face... just like every user!!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <=

(0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("You loose 5 life points!!"));
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
dummy =

(entitySetInt("_prof_edwards", Type.CHARACTER, "_life", (entityHasInt("_prof_edwards",
Type.CHARACTER,

"_life") - (5))));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum ==

(4))
endGame();
}
}
else {
{
System.out.println(""+ ("You got hit by windows.. you will never function

correctly again!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum ==

(4))
endGame();
System.out.println(""+ ("You loose 5 life points!!"));
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
dummy = (entitySetInt("_prof_edwards",

Type.CHARACTER, "_life", (entityHasInt("_prof_edwards", Type.CHARACTER, "_life") -
(5))));
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
}
}
}
}
_ucb();
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame

();
}
}
}
else {
{
System.out.println(""+ ("Get ready to fight a huge PLT mob!!"));
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
int num = r.nextInt
```

```
(100);
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame

();
if(num >= 0 && num < 40) {
{
dummy = (_attack = (1));
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("Someone throws

a C Reference Manual at you!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) ||

_itemNum == (4))
endGame();
}
}
if(num >= 40 && num < 70) {
{
dummy = (_attack = (2));
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println(""+

("someone is trying to throw the Ocaml camel at your feet!!"));
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
}
}
if(num >= 70 && num < 100) {
{
dummy =

(_attack = (3));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum ==

(4))
endGame();
System.out.println(""+ ("They are all screaming about a new language that is scanned

backwards!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
}
}
Map<String,String> keysToActionName20 = new HashMap<String, String>();
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
Map<String, String>

actionNameToOutput20 = new HashMap<String, String>();
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println("CHOOSE AN ACTION:");
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
keysToActionName20.put("d", "_duck");
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <=

(0)) || _itemNum == (4))
```

```
endGame();
actionNameToOutput20.put("_duck", "duck");
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println

("Type d for duck");
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum ==

(4))
endGame();
keysToActionName20.put("j", "_jump");
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") <= (0)) || _itemNum == (4))
endGame();
actionNameToOutput20.put("_jump", "jump");
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println("Type j for jump");
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <=

(0)) || _itemNum == (4))
endGame();
keysToActionName20.put("c", "_cover");
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
actionNameToOutput20.put("_cover", "cover your ears");
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println("Type c for cover

your ears");
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
Scanner in20 = new Scanner(System.in);
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") <= (0)) || _itemNum == (4))
endGame();
String input20 = in20.nextLine();
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
while(!

keysToActionName20.containsKey(input20)) {
System.out.println("Invalid input, try again");
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
input20

= in20.nextLine();
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum ==

(4))
endGame();
}
System.out.println("You typed " + input20);
if ((entityHasInt("_prof_edwards",
```

```
Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
String action20 =

keysToActionName20.get(input20);
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) ||

_itemNum == (4))
endGame();
if(action20.equals("_duck")) {
{
if(_attack == (1)) {
{
System.out.println(""+

("You escaped the horror!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) ||

_itemNum == (4))
endGame();
System.out.println(""+ ("They are getting tired!"));
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
dummy = (entitySetInt

("_plt_mob", Type.CHARACTER, "_life", (entityHasInt("_plt_mob", Type.CHARACTER, "_life")
- (1))));
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
}
}
else

{
if(_attack == (2)) {
{
System.out.println(""+ ("You got hit by the camel.. you appreciate O'Caml a

little less!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum ==

(4))
endGame();
System.out.println(""+ ("You loose 5 life points!!"));
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
dummy = (entitySetInt("_prof_edwards",

Type.CHARACTER, "_life", (entityHasInt("_prof_edwards", Type.CHARACTER, "_life") -
entityHasInt

("_plt_mob", Type.CHARACTER, "_damage"))));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life")

<= (0)) || _itemNum == (4))
endGame();
}
}
else {
{
System.out.println(""+ ("You heard it all... You are

utterly confused!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum

== (4))
```

```
endGame();
System.out.println(""+ ("You loose 5 life points!!"));
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
dummy = (entitySetInt

("_prof_edwards", Type.CHARACTER, "_life", (entityHasInt("_prof_edwards", Type.CHARACTER,
"_life") -

entityHasInt("_plt_mob", Type.CHARACTER, "_damage"))));
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
}
}
}
}
_ucb();
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
}
if(action20.equals

("_jump")) {
{
if(_attack == (2)) {
{
System.out.println(""+ ("You escaped that which makes many stumble

and crack their heards!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) ||

_itemNum == (4))
endGame();
System.out.println(""+ ("The mob is getting bored!"));
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
dummy = (entitySetInt

("_plt_mob", Type.CHARACTER, "_life", (entityHasInt("_plt_mob", Type.CHARACTER, "_life")
- (1))));
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
}
}
else

{
if(_attack == (1)) {
{
System.out.println(""+ ("You got hit by the manual square in the face... your

brain is now a big null pointer!!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <=

(0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("You loose 5 life points!!"));
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
```

```
dummy =

(entitySetInt("_prof_edwards", Type.CHARACTER, "_life", (entityHasInt("_prof_edwards",
Type.CHARACTER,

"_life") - entityHasInt("_plt_mob", Type.CHARACTER, "_damage"))));
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
}
}
else {
{
System.out.println(""+ ("You

heard it all... You are utterly confused!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("You loose 5 life points!!"));
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
dummy =

(entitySetInt("_prof_edwards", Type.CHARACTER, "_life", (entityHasInt("_prof_edwards",
Type.CHARACTER,

"_life") - entityHasInt("_plt_mob", Type.CHARACTER, "_damage"))));
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
}
}
}
}
_ucb();
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
}
if(action20.equals

("_cover")) {
{
if(_attack == (3)) {
{
System.out.println(""+ ("Good! You would have never been able to

make a good compiler again!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) ||

_itemNum == (4))
endGame();
System.out.println(""+ ("Every head in that mob hurts!"));
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
dummy = (entitySetInt

("_bill_gates", Type.CHARACTER, "_life", (entityHasInt("_bill_gates", Type.CHARACTER,
"_life") -

(1))));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
```

```
endGame

();
}
}
else {
if(_attack == (1)) {
{
System.out.println(""+ ("You got hit by the manual square in the

face... your brain is now a big null pointer!!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("You loose 5 life points!!"));
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
dummy =

(entitySetInt("_prof_edwards", Type.CHARACTER, "_life", (entityHasInt("_prof_edwards",
Type.CHARACTER,

"_life") - entityHasInt("_plt_mob", Type.CHARACTER, "_damage"))));
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
}
}
else {
{
System.out.println(""+ ("You

got hit by the camel.. you appreciate O'Caml a little less!"));
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("You loose 5

life points!!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum ==

(4))
endGame();
dummy = (entitySetInt("_prof_edwards", Type.CHARACTER, "_life", (entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") - entityHasInt("_plt_mob", Type.CHARACTER,
"_damage"))));
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
}
}
}
}
_ucb();
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame

();
}
}
}
}
}
}
endGame();
```

```
}
//start funtion
public void _mcdonalds() {
currentLocation = "_mcdonalds";
while (!

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))){
{
if(isTrue

(entityExistsItem("_mcdonalds", Type.LOCATION, "_fish_sandwich"))) {
{
System.out.println(""+ ("Ronald

gives you a riddle!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum

== (4))
endGame();
System.out.println(""+ ("What Burger was featured in the greatest advertising

campaign ever?"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum ==

(4))
endGame();
Map<String,String> keysToActionName2 = new HashMap<String, String>();
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
Map<String, String>

actionNameToOutput2 = new HashMap<String, String>();
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println("CHOOSE AN ACTION:");
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
keysToActionName2.put("f", "_fish");
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <=

(0)) || _itemNum == (4))
endGame();
actionNameToOutput2.put("_fish", " fish sandwitch");
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println("Type f for  fish sandwitch");
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") <= (0)) || _itemNum == (4))
endGame();
keysToActionName2.put("b", "_burger");
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
actionNameToOutput2.put("_burger", " Big Mac");
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println("Type b for  Big Mac");
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
```

```
endGame();
keysToActionName2.put("c", "_caesar");
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <=

(0)) || _itemNum == (4))
endGame();
actionNameToOutput2.put("_caesar", " Caesar Salad");
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println("Type c for  Caesar Salad");
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") <= (0)) || _itemNum == (4))
endGame();
Scanner in2 = new Scanner(System.in);
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
String input2 =

in2.nextLine();
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
while(!keysToActionName2.containsKey(input2)) {
System.out.println("Invalid input, try

again");
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
input2 = in2.nextLine();
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0))

|| _itemNum == (4))
endGame();
}
System.out.println("You typed " + input2);
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
String action2 =

keysToActionName2.get(input2);
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) ||

_itemNum == (4))
endGame();
if(action2.equals("_fish")) {
{
System.out.println(""+ ("Yes, you do believe

in magic!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("YOu have earned a  fish Burger"));
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println(""+

("Lets go back to Columbia to keep on with the adventure!!"));
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
_prof_edwards.addItem ("_fish_sandwich",

currentLocation, locations);
```

```
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) ||

_itemNum == (4))
endGame();
if(_itemNum == (3)) {
{
System.out.println(""+ ("You did it!!"));
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("You have collected all four items"));
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("With them you

build the most premiums compiler eveeeeer!!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") <= (0)) || _itemNum == (4))
endGame();
}
}
else {
//Empty stmt
}
dummy = (_itemNum = (_itemNum +

(1)));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame

();
}
_columbia();
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum ==

(4))
endGame();
}
if(action2.equals("_burger")) {
System.out.println(""+ ("no... come on... you were

there!!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
_mcdonalds();
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum

== (4))
endGame();
}
if(action2.equals("_caesar")) {
System.out.println(""+ ("Thats not even a

burger!!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
_mcdonalds();
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum

== (4))
endGame();
}
}
}
else {
{
System.out.println(""+ ("you have collected what you needed here"));
```

```
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
Map<String,String> keysToActionName1 = new HashMap<String, String>();
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
Map<String, String>

actionNameToOutput1 = new HashMap<String, String>();
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println("CHOOSE AN ACTION:");
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
keysToActionName1.put("c", "_col");
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0))

|| _itemNum == (4))
endGame();
actionNameToOutput1.put("_col", "Columbia");
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println

("Type c for Columbia");
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) ||

_itemNum == (4))
endGame();
Scanner in1 = new Scanner(System.in);
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
String input1 = in1.nextLine();
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
while(!

keysToActionName1.containsKey(input1)) {
System.out.println("Invalid input, try again");
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
input1 =

in1.nextLine();
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
}
System.out.println("You typed " + input1);
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
String action1 = keysToActionName1.get

(input1);
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
```

```
if(action1.equals("_col")) {
//Empty stmt
_columbia();
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
}
}
}
}
}
endGame();
}
//intdecinit
int

_narrated;
//start funtion
public void _taiwan() {
currentLocation = "_taiwan";
while (!((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))){
{
if((entityHasInt("_godzilla",

Type.CHARACTER, "_life") <= (0))) {
{
System.out.println(""+ ("...oh no wait, you beat Godzilla!! holy

shhh... wow!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum ==

(4))
endGame();
System.out.println(""+ ("You have received the PDP-8!!"));
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
if(!isTrue

(entityExistsItem("_prof_edwards", Type.CHARACTER, "_PDP_8"))) {
{
_prof_edwards.addItem ("_PDP_8",

currentLocation, locations);
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) ||

_itemNum == (4))
endGame();
if(_itemNum == (3)) {
{
System.out.println(""+ ("You did it!!"));
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("You have collected all four items"));
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("With them you

build the most premiums compiler eveeeeer!!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") <= (0)) || _itemNum == (4))
endGame();
}
```

```
}
else {
//Empty stmt
}
dummy = (_itemNum = (_itemNum +

(1)));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame

();
}
}
else {
//Empty stmt
}
}
}
else {
//Empty stmt
}
if(isTrue(entityExistsItem("_taiwan", Type.LOCATION,

"_PDP_8"))) {
{
if(_narrated == (0)) {
{
System.out.println(""+ ("You have traveled to Taiwan to get the

biggest PDP-8 in the world"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) ||

_itemNum == (4))
endGame();
System.out.println(""+ ("You need it for its humongous stack!!"));
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("The problem is, it belongs to Godzilla!!"));
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
dummy = (_narrated =

(1));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
}
}
else {
//Empty stmt
}
System.out.println(""+ ("Godzilla is about to step on you!!"));
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
Map<String,String>

keysToActionName12 = new HashMap<String, String>();
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") <= (0)) || _itemNum == (4))
endGame();
Map<String, String> actionNameToOutput12 = new

HashMap<String, String>();
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) ||

_itemNum == (4))
```

```
endGame();
System.out.println("CHOOSE AN ACTION:");
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
keysToActionName12.put("s", "_shoot");
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
actionNameToOutput12.put("_shoot", "shooting your laser pointer at his foot");
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println

("Type s for shooting your laser pointer at his foot");
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
keysToActionName12.put("d", "_die");
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
actionNameToOutput12.put("_die", "dying squished between its toes");
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println("Type d for dying

squished between its toes");
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) ||

_itemNum == (4))
endGame();
keysToActionName12.put("t", "_tickle");
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
actionNameToOutput12.put("_tickle",

"tickling the monster's foot");
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) ||

_itemNum == (4))
endGame();
System.out.println("Type t for tickling the monster's foot");
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
Scanner

in12 = new Scanner(System.in);
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) ||

_itemNum == (4))
endGame();
String input12 = in12.nextLine();
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
while(!keysToActionName12.containsKey

(input12)) {
```

```
System.out.println("Invalid input, try again");
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
input12 = in12.nextLine();
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
}
System.out.println("You typed " + input12);
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life")

<= (0)) || _itemNum == (4))
endGame();
String action12 = keysToActionName12.get(input12);
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
if

(action12.equals("_shoot")) {
{
System.out.println(""+ ("You hit the monster in the foot"));
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
dummy =

(entitySetInt("_godzilla", Type.CHARACTER, "_life", (entityHasInt("_godzilla",
Type.CHARACTER,

"_life") - entityHasInt("_laser", Type.ITEM, "_damage"))));
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println(""+ (" You did 10

points damage!!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum ==

(4))
endGame();
System.out.println(""+ ("Keep fighting!!"));
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
}
_taiwan();
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
}
if(action12.equals

("_die")) {
{
System.out.println(""+ ("You have died between its green toes"));
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
dummy = (entitySetInt

("_prof_edwards", Type.CHARACTER, "_life", (0)));
```

```
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") <= (0)) || _itemNum == (4))
endGame();
}
_columbia();
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
}
if(action12.equals("_tickle")) {
{
System.out.println(""+ ("the monster fell back laughing"));
if ((entityHasInt("_prof_edwards",

Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println(""+ ("You have done

50 damage!!"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
dummy = (entitySetInt("_godzilla", Type.CHARACTER, "_life", (entityHasInt("_godzilla",

Type.CHARACTER, "_life") - (50))));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0))

|| _itemNum == (4))
endGame();
}
_taiwan();
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <=

(0)) || _itemNum == (4))
endGame();
}
}
}
else {
{
System.out.println(""+ ("You have the calculator. Time to

return to columbia"));
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum

== (4))
endGame();
Map<String,String> keysToActionName1 = new HashMap<String, String>();
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
Map<String, String> actionNameToOutput1 = new HashMap<String, String>();
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println

("CHOOSE AN ACTION:");
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum

== (4))
endGame();
keysToActionName1.put("c", "_col");
if ((entityHasInt("_prof_edwards", Type.CHARACTER,

"_life") <= (0)) || _itemNum == (4))
endGame();
actionNameToOutput1.put("_col", "Columbia");
```

```
if

((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
System.out.println("Type c for Columbia");
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life")

<= (0)) || _itemNum == (4))
endGame();
Scanner in1 = new Scanner(System.in);
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
String input1 =

in1.nextLine();
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
while(!keysToActionName1.containsKey(input1)) {
System.out.println("Invalid input, try

again");
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
input1 = in1.nextLine();
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0))

|| _itemNum == (4))
endGame();
}
System.out.println("You typed " + input1);
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
String action1 =

keysToActionName1.get(input1);
if ((entityHasInt("_prof_edwards", Type.CHARACTER, "_life") <= (0)) ||

_itemNum == (4))
endGame();
if(action1.equals("_col")) {
//Empty stmt
_columbia();
if ((entityHasInt

("_prof_edwards", Type.CHARACTER, "_life") <= (0)) || _itemNum == (4))
endGame();
}
}
}
}
}
endGame();
}
}

abstract class Entity {

   Map<String, Integer> intAttrs = new HashMap<String, Integer>();
   Map<String, String> strAttrs = new HashMap<String, String>();

   public void addIntAttr(String name, int value) {
      intAttrs.put(name, value);
   }

   public void addStrAttr(String name, String value) {
      strAttrs.put(name, value);
   }
}
```

```
class Location extends Entity {
   Set<String> characters = new HashSet<String>();
   Set<String> items = new HashSet<String>();

   public void addItem(String name, Map<String, Item> itemses) {
      if(itemses.containsKey(name))
                  items.add(name);
      else
            System.out.println("Error: The item you attempted to add no longer exists");
   }

   public void addItem(String name){
      items.add(name);
      }

   public void removeItem(String name) {
      items.remove(name);
   }

   public void showCharacter(String name, Map<String, Character> characterses) {
      if(characterses.containsKey(name))
      characters.add(name);
      else
            System.out.println("Error: The character you attempted to use no longer
exists");
   }
   public void showCharacter(String name){

      characters.add(name);
      }

   public void hideCharacter(String name) {
      characters.remove(name);
   }
}

class Character extends Entity {
   Set<String> items = new HashSet<String>();

   public void addItem(String name, String locationNow, Map<String, Location> locations){

      if(locations.get(locationNow).items.contains(name)){
            locations.get(locationNow).removeItem(name);
            items.add(name);
      }
      else
            System.out.println("Error: The item you attempted to grab is not in this
location");
}

public void removeItem(String name, String locationNow, Map<String, Location> locations)
{
            if (items.contains(name)){
            items.remove(name);
            locations.get(locationNow).addItem(name);
   }
      else
            System.out.println("Error: The character does not have the item you attempted
to

drop");
}

   public void addItem(String name) {
      items.add(name);
   }

   public void removeItem(String name) {
      items.remove(name);
   }
}
```

```
class Item extends Entity {
}
```