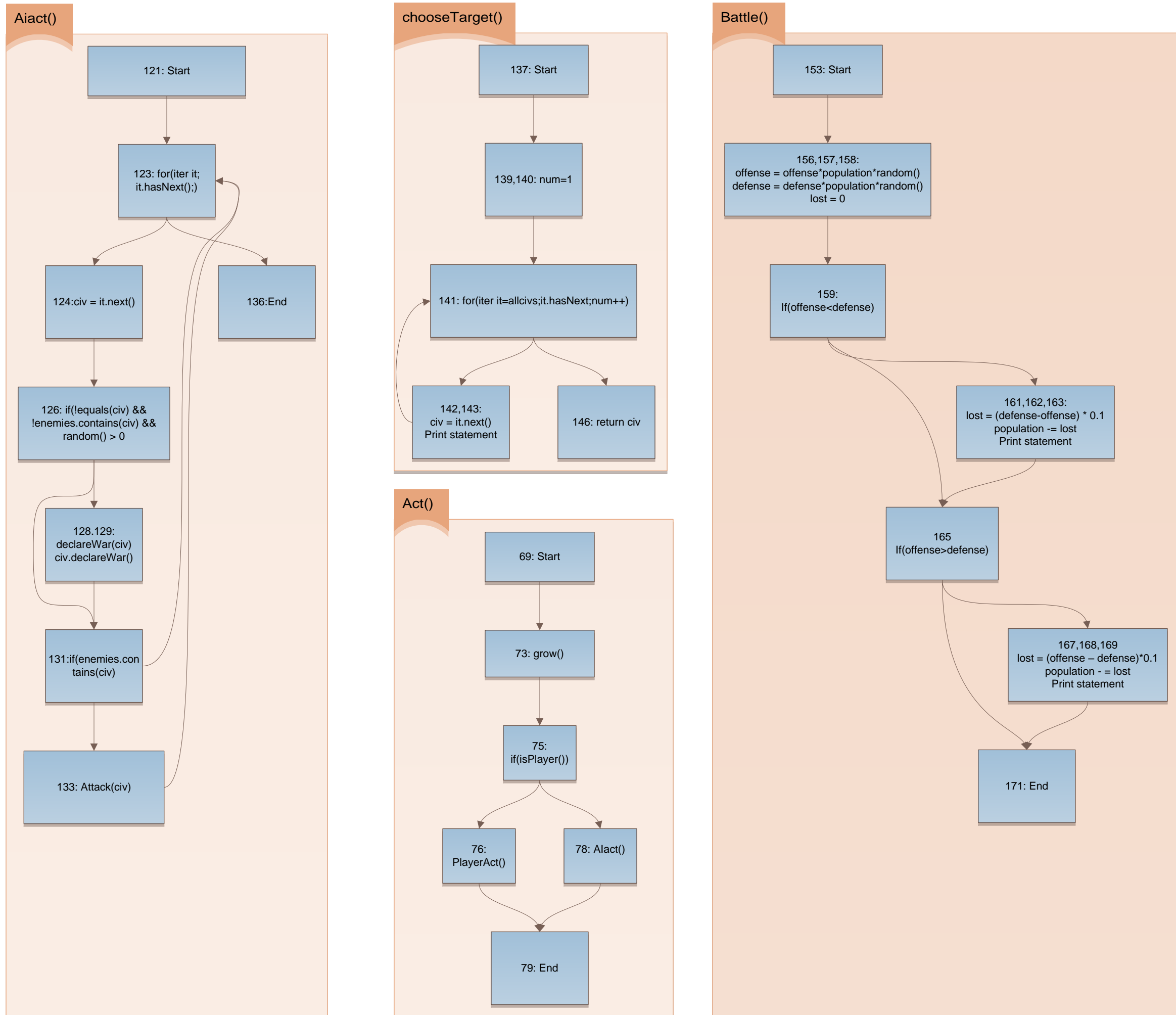
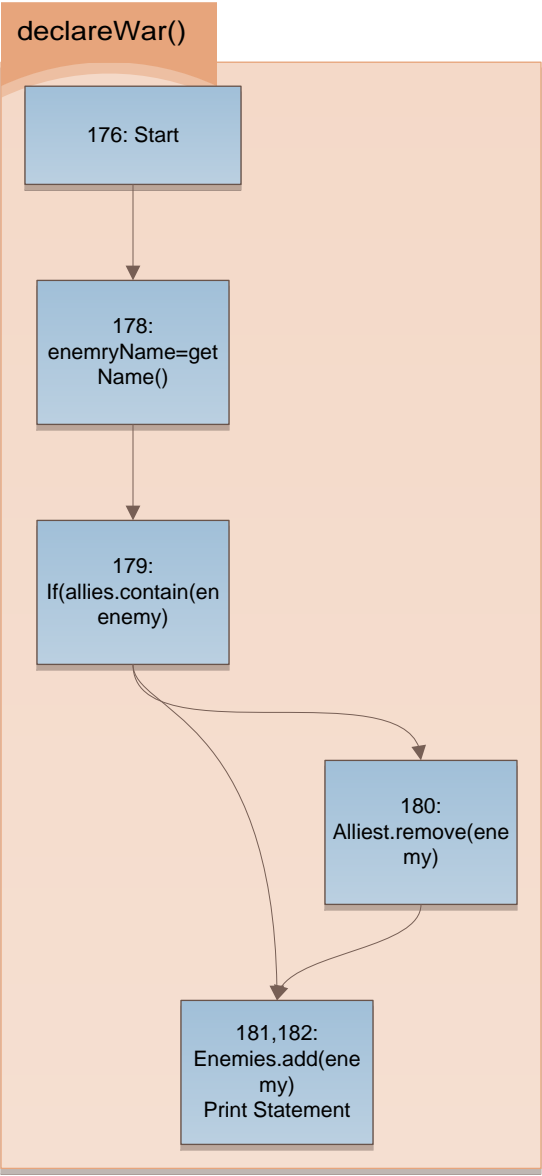


CFG: Civilization.java





```

80 public void PlayerAct(ArrayList<Civilization> allCivs)
81 {
82     System.out.println("Your turn. Select an action");
83     System.out.println("1: Check my team.");
84     System.out.println("2: Attack another team.");
85     System.out.println("3: Declare War on another team.");
86     System.out.println("4: End Your turn.");
87     int num = 0;
88     int attacks = 0;
89     while (num==0)
90     {
91         num = Reader.pickOption(4);
92         switch (num){
93             case 1:{
94                 System.out.println("Population: " + this.population)
95                 num=0;
96                 break;
97             }
98             case 2:{
99                 this.attack(this.chooseTarget(allCivs));
100                 num=0;
101                 break;
102             }
103             case 3:{
104                 this.declareWar(this.chooseTarget(allCivs));
105                 num=0;
106                 break;
107             }
108             case 4:{
109                 break;
110             }
111             default: {
112                 num=0;
113                 break;
114             }
115         }
116     }
117 }
118
119 }
120
121 public void AIact(ArrayList<Civilization> allCivs)
122 {
123     for(Iterator<Civilization> it = allCivs.iterator(); it.hasNext(); ) {
124         Civilization civ = it.next();
125
126         if (!this.equals(civ) && !this.enemies.contains(civ.getName()) && Math.random()>0.9)
127         {
128             this.declareWar(civ);
129             civ.declareWar(this);
130         }
131         if (this.enemies.contains(civ.getName()))
132         {
133             this.attack(civ);
134         }
135     }
136 }
137 private Civilization chooseTarget(ArrayList<Civilization> allCivs)
138 {
139     int num = 1;
140     Civilization civ;
141     for(Iterator<Civilization> it = allCivs.iterator(); it.hasNext();num++ ) {
142         civ = it.next();
143
144         System.out.println(num + ": " + civ);
145     }
146     return allCivs.get(Reader.pickOption(num));
147 }
148 private void attack(Civilization civ)
149 {
150     System.out.print(this + " attacked " + civ + "! ");
151     this.battle(civ);
152 }

```

```
153 private void battle(Civilization def)
154 {
155
156     double offense = this.offense*this.population*Math.random();
157     double defense = def.defense*def.population*Math.random();
158     int lost = 0;
159     if (offense < defense)
160     {
161         lost = (int) Math.round( (defense - offense)*0.1);
162         this.population -= lost;
163         System.out.println(this + " lost " + lost + " people.");
164     }
165     if (offense > defense)
166     {
167         lost = (int) Math.round((offense - defense)*0.1);
168         def.population -= lost;
169         System.out.println(def + " lost " + lost + " people.");
170     }
171 }
172
173
174
175
176 public void declareWar(Civilization enemy)
177 {
178     String enemyName = enemy.getName();
179     if (this.allies.contains(enemyName))
180         this.allies.remove(enemyName);
181     this.enemies.add(enemyName);
182     System.out.println(this.civName + " declared war on " + enemy + "!");
183 }
```