

L7 Project

Table of contents

- Night
- Day
- Technical Specs
- Next Steps

[Wilderness Survival](#)

Let's make a game! A very, very bleak survival game! Your airplane crashed onto a deserted, desert island. Well, deserted except for all of the ferocious shadow monsters that will tear you limb from limb. The game will be a type of resource management game with two different phases: night and day. There will be two resources: wood and food.

[Night](#)

The shadow monsters are scared of fire, as they should be. In order to survive the night, you must have a bonfire. You must feed the bonfire **2 wood per hour** and the night lasts for **12 hours**. You can only collect wood and food in the day time, and only while you have enough energy (food). If you run out of wood during the night time, it's **game over** (shadow monsters eat you).

🔗 Day

The day time lasts for 12 hours and is the only time you can collect **wood** or **food**. While you are out collecting, you are expending energy and using **up 1 food per hour**. If you are out collecting **wood**, you can collect **3 per hour**. If you are out collecting **food**, you can collect 4 per hour, but you still need to eat food so results to a gain of **2 per hour**. If you run out of food in the day time, it's **game over** (you starve).

🔗 Technical Specs

The goal of the game is to balance collecting **wood** and **food**. The game should initialize your player with at least some wood and food to start off with. Then during the **day phase**, for each of the 12 hours allow the user to collect either wood or food. Change your food and wood stores appropriately, and note that you still eat 2 food while collecting food, resulting in a gain of **2 per hour**.

You can write all of your code in your main, unless you think something will work better as a method. Note that if you keep track of wood and food stores as a variable in your main, **you cannot use a method to change those values**; that logic would need to be in the main itself. A method named **alive** that takes in **food** as a parameter and returns either **true** or **false** is an example of something that would work well as a method.

- Night
 - -= 2 wood per hour
 - if wood <= 0, game over
 - no food gain or loss
- Day
 - prompt user for resource collection choice
 - your choice on implementation could be:
 - 12 choices, one for every hour

- ask for resource followed by number
 - "wood 4" meaning collect wood for 4 hours
 - "food 8" meaning collect food for 8 hours
 - use the String.split() method
 - if food ≤ 0 , game over
 - if collect wood
 - += 3 wood per hour
 - -= 1 food per hour
 - if collect food
 - += 4 food per hour
 - -= 2 food per hour
-

Next Steps

Once you've finished the basic game loop of resource collection and survival, let's think about more features we could add to the game.

- Random events using random number generator
 - Monsters thrash your campsite, lose 50% wood and 50% food
 - Find a boon of food, for a few hours gain double food
 - Level system
 - After 10 days, you learn how to develop tools. +1 to both food and wood gain per hour
 - After 15 days, you learn how to make a torch, allowing you to collect resources at a slower pace during the night
 - After 20 days, you have food store all around the island, gives you an extra life if food ≤ 0
 - Different difficulties/player perks
 - Hungry: you need to eat food during the night as well
 - Engineer: immediately get access to tech tree
 - Runner: +1 to both food and wood per hour, but only for 6 hours
 - Lazy: You only collect items 8 out of the available 12 hours of daytime
-

Some starter code has been provided below. As always, be wary of straight copying and pasting; watch for curly braces and indentation. This starter code is 12 choices for 12 hours of daytime; you'll need to change a few things if you want the shorthand "wood 4", "food 8".

Java

COPY

```
public static void main(String[] args) {
    boolean alive = true;
    int food = 20;
    int wood = 20;
    Scanner scan = new Scanner(System.in);
    System.out.println("Airplane go brrrrr!! (in a bad way)");
    System.out.println("You crash land on a deserted island. Dawn");

    while(alive) {
        // daytime loop
        for(int i = 0; i < 13; i++) {
            if(food > 0){
                System.out.println("It is day time! Time to collect food");
                System.out.println("Should I collect wood or food? ");
                String s = scan.nextLine();

                if(s.equals("wood")) {
                    wood = wood - 1;
                }
                else if (s.equals("food")) {
                    food = food - 1;
                }
            }
            else{
                System.out.println("You run out of food and don't survive");
            }
        }
        // night time loop
        for(int i = 0; i < 13; i++) {
            if(wood > 0){
                System.out.println("It is night time! Time to collect wood");
                System.out.println("Should I collect wood or food? ");
                String s = scan.nextLine();

                if(s.equals("wood")) {
                    wood = wood - 1;
                }
                else if (s.equals("food")) {
                    food = food - 1;
                }
            }
            else{
                System.out.println("You run out of wood and don't survive");
            }
        }
    }
}
```

```
}  
}
```

Published: April 26, 2021

Last updated: April 26, 2021



[← Previous](#)

L7-3 Exercise

[Next →](#)

L7 Summary

© 2021 Coding Minds - All right Reserved.