Forest

Game by Tricia Glassborow

Game premise

Forest is a text based game where you start in a plane crash that leaves you stranded in the middle of a forest. You start walking to try get help and as you go you will have to make decisions about which path to take. Each decision effects your journey.

Your aim is to make it out of the forest alive. Different challenges will stand in your way though.

- Managing your health, eating berries will increase it.
- Fighting animals that block your path, picking up weapons will increase your chances of winning.
- Environmental dangers

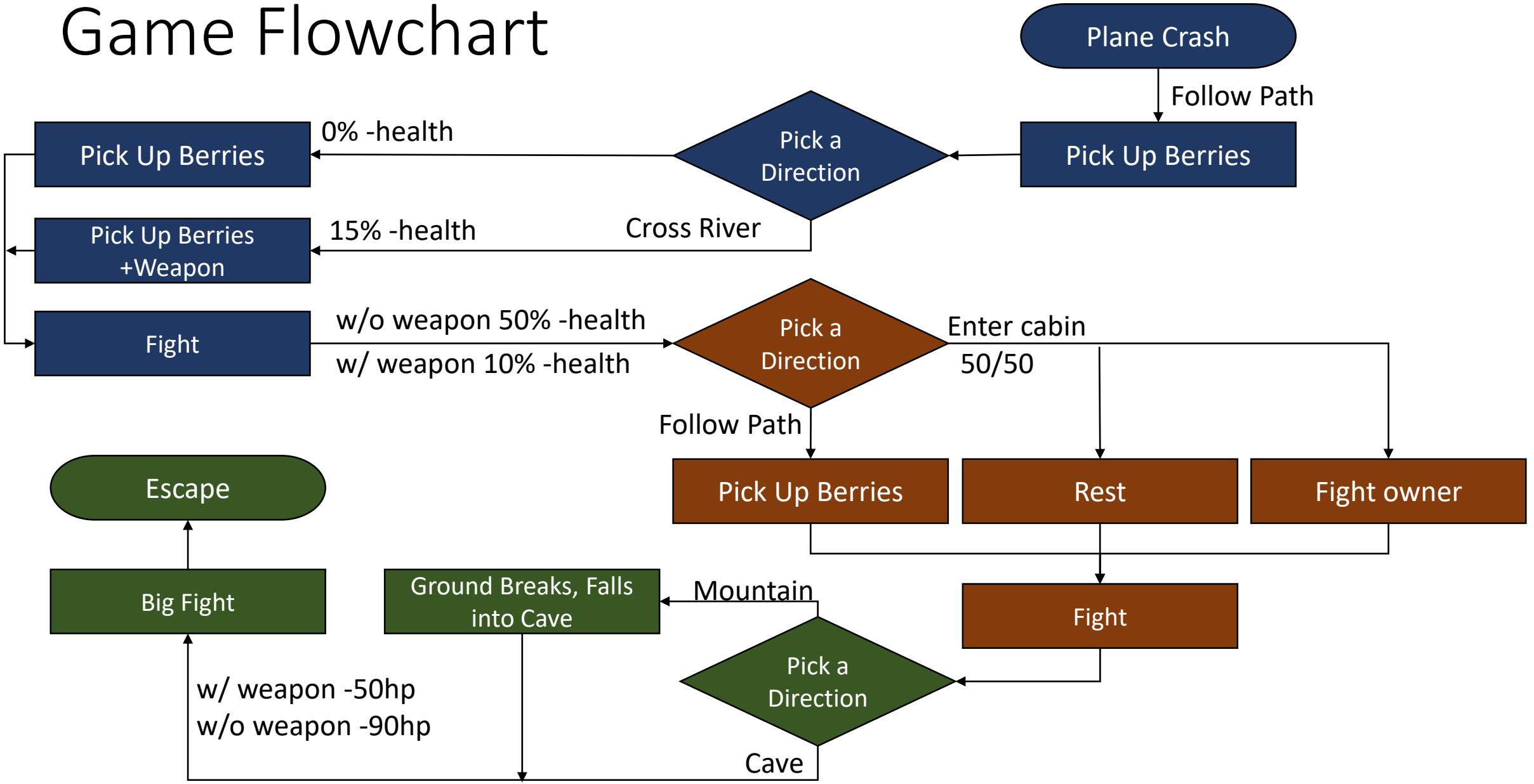
Designing the game

First step we took was to draw out the path and add notes like putting in where each fight will happen, where berries and weapons could be picked up, any environmental features we want to include (like rivers or mountains) and the percentage chance of an event occurring.

Structing this like a flowchart allowed us to visualise how the game will run and give us a clear plan, that we both agreed on, to refer back to when coding. This allowed us to code independently then bring our work together to create the game.

Level 1 | Level 2 | Level 3

Game Flowchart



Features

- Health reaching 0 means you die and game ends
- Player can pick up berries
- Player has health
- Berries and resting increases health
- Fighting animals or people decreases heath

Coding Decisions

Using the original game flowchart, we identified we would need a class to run possibility calculations as they would be called by multiple methods.

We'd use a random number generator to pick a random number between 0 and 99 and then using the parameters to decide whether the output should be true or false.

Example: The number generated is 8.

The possibility entered in the parameters is 10, that means the class will output true if the random number is between 0 and 9

8 is between 0 and 9 so the output is true.

Game play demo

evaluation

The game runs and you can make it to the end of the first level. The final game has all intended features. However, there's no checking system in place to make sure the player's inputs are valid. Next time, I would add a way to check the player's inputs and, if they aren't valid, ask for them again.

There is also no GUI, if I were to make the game again I would add that so the game feels more immersive and professional. There is also no checking to see if the HP has reached zero to end the game as the player has died.