

# Introduction to C#



In an ancient land, where legends haunt the air and mysteries lurk around every corner, there sits a formidable castle, perched upon a mountainous precipice. There was a time when this castle's towering wall acted as its kingdom's proud defence, but its once-merry halls now echo with whispers of fear and desolation.

Unleashed from the depths of darkness, a malevolent dragon known as Morgoth the Devourer, has focussed its wrath upon the castle walls with relentless ferocity. Each passing day brings a new assault, threatening to breach the once-impenetrable defences. The very existence of the kingdom hangs in the balance, for if those walls should crumble, all will be lost.

## The game

Your task is to program a turn-based game based on the defence of the castle wall.

On each turn the dragon will decide whether or not to attack the castle. As the head of the castle guard you need to decide whether to spend the turn defending against a possible attack or rebuilding the castle wall.

At the start of the game the castle wall starts with 100 HP.

- If you choose to defend the castle and the dragon decides to attack, the castle wall HP is reduced by a random value between 1 and 20.
- If you choose to rebuild the castle wall and the dragon attacks, the castle wall HP is reduced by a random number between 10 and 35.
- If you choose to rebuild the castle wall and the dragon chooses not to attack, the castle wall HP is increased by 15.

The player must prevent the castle wall from being destroyed completely over the course of 10 turns.

## Instructions

1. Create two integer variables: `castleWallHP` and `turn`. Assign them the values 100 and 1, respectively.
2. Create a While loop. This should iterate while `castleWallHP` is greater than 0 and `turn` is less than 10.
3. Inside the while loop:
  - a. Let the player know what turn it is.
  - b. Ask them whether they want to rebuild the castle or defend against the next attack.
  - c. Get the player's choice and store it in a string variable called `action`.

The output might look like the following:

```
Turn 1
-----
Morgoth is sleeping, but will attack again soon.

What do you want to do?
Enter R to rebuild the castle wall.
Enter D and prepare to defend Morgoth's next attack.

>>
```

4. Inside the while loop, generate a random number, 1 or 2, and store it in a variable called `dragonAction`. 1 will represent the dragon deciding to attack. 2 will represent the dragon resting.
5. Still inside the while loop, create a selection statement, as follows:
  - a. If the player chose to defend the castle and the dragon decides to attack, the castle wall HP is reduced by a random value between 1 and 20.
  - b. If the player chose to rebuild the castle wall and the dragon attacks, the castle wall HP is reduced by a random number between 10 and 35.
  - c. If the player chose to rebuild the castle wall and the dragon chooses not to attack, the castle wall HP is increased by 15.
6. Create a selection statement outside of the loop to display the outcome of the game, as follows:
  - a. If the castle wall was destroyed, the player loses the game.
  - b. If the castle wall is still in tact after the tenth turn, the player wins.

## Review

Analyse the mechanics of the game.

1. Calculate the likelihood of the player winning and losing the game.
2. Redesign the mechanics so that there is a 50/50 chance of winning/losing.
3. Decide on some ways that you could further develop the game.