Project 1: A Simple Adventure Game in MIPS

Deadline: Sunday, March 18, 2018 at 11:59 PM

Description

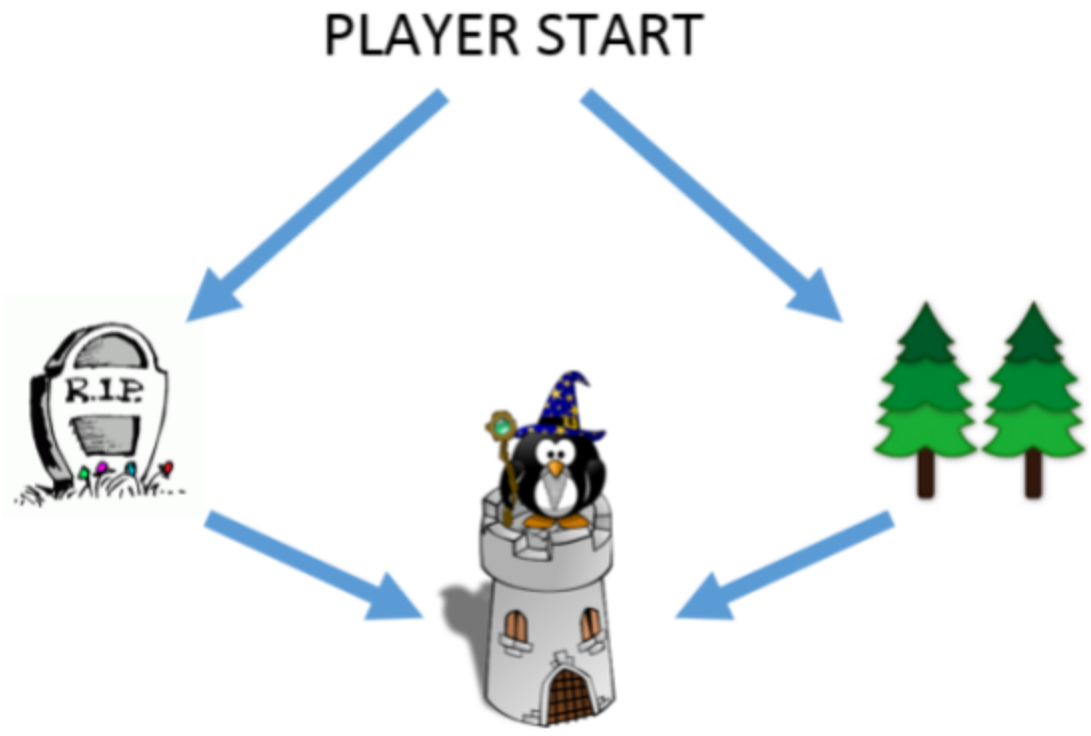
In this assignment, you will write a MIPS program for a game! The game will be a simple text-based adventure game where the player gets to do things such as choose a character and battle enemies.

The Goal of the Game

Defeat The Evil Wizard!

The Evil Wizard is located in The Castle. To reach The Castle, you must go through either The Forest or The Graveyard.

In The Forest you will encounter Goblins; in The Graveyard, Skeletons.



The Playable Characters

The player will choose a character at the start of the game. Here are the choices:

1. Rogue [HP: 55; Strength: 8; Weapon: Short Sword (1 - 4)]
2. Paladin [HP: 35; Strength: 14; Weapon: Long Sword (3 - 7)]
3. Jackie Chan [HP: 45; Strength: 10; Weapon: Jump Kick (2 - 6)]

What are the differences between the characters? The Rogue has high hit points (HP) and low Strength. The Paladin has low HP and high Strength. Jackie Chan is balanced and therefore has medium-level HP and Strength.

The Enemy Characters

There are 3 types of enemies in the game.

1. Goblin [HP: 25; Strength: 4; Weapon: Axe (2 - 6)]
2. Skeleton [HP: 25; Strength: 3; Weapon: Short Sword (1 - 4)]
3. Evil Wizard [HP: 40; Strength: 8; Weapon: Fire Blast (4 - 10)]

HP and Strength

Two attributes that every playable character and every enemy has are hit points (HP) and Strength.

The hit points determine how much “health” or “life” a character has. A character is dead/defeated when the character’s hit points value is less than or equal to 0. When a character is hit with an attack, that attack reduces the attacked character’s HP.

The Strength of a character is one of the factors that determines the amount of damage that the character delivers when the character attacks. When a character attacks, the character’s Strength will be added to the character’s weapon damage, and that resulting sum is the attack value (ATK) for the attack, which is the magnitude of the attack (more on this below).

Weapons

Every character has a weapon. Each weapon has a damage attribute; this attribute is a range of values. For example, a short sword has a damage attribute of 1 - 4 (i.e. the set { 1,2,3,4 }).

When a character attacks with a weapon, generate **a random integer** in the range of the damage attribute to determine the damage value of the weapon for that attack (more on this below).

Attack Value (ATK)

An attack’s value (ATK) is calculated as follows: ATK = Strength + damage.

Here’s an example: assume a Skeleton has a Strength of 3 and weapon damage of 1 - 4. When that Skeleton attacks, we need to determine the damage from the weapon and add this to the

Strength value (3 in this case) to get the ATK value. Assume we generate a damage value of 2; then for this attack, ATK = 3 + 2 = 5.

The Battle System

When the player battles an enemy, each of them will take turns attacking; the player will always attack first. The enemy will attack after the player, then the player will attack again, then the enemy, etc. The battle will end when one of the participants is defeated (i.e. hit points value is less than or equal to 0).

Now, what is an "attack" in this game? An attack is a number that is determined by the attacking character, and when an attack is delivered by a character, that number is subtracted from the HP of the attacked character.

For example, if the Rogue attacks a Goblin with an ATK value of 9, and the Goblin has 25 HP, then after the attack the Goblin will have 25 - 9 = 16 HP.

Battle Example

Let's look at an example battle.

Assume we have the following two characters with the given attributes and weapons.

Skeleton	**VS**	**Paladin**
HP: 25		HP: 35
Strength: 3		Strength: 14
Weapon: Short Sword (1 - 4)		Weapon: Long Sword (3 - 7)

The Paladin attacks first with ATK = 14 + 4 = 18, and so Skeleton's HP is reduced to 25 - 18 = 7.

Now, how did we get ATK = 14 + 4 = 18? Recall that ATK = Strength + damage. The Paladin's Strength is 14, so that's where the 14 came from; now what about the 4? The 4 is the damage from the Long Sword (assume it was generated randomly from the range 3 - 7). So that is why the Paladin's ATK is 18 in this case.

Next, the Skeleton attacks. Assume his weapon does 1 damage for this attack, and so his ATK = 3 + 1 = 4, which reduces the Paladin's HP to 35 - 4 = 31.

Since both characters have HP > 0, the fight continues.

Now it's the Paladin's turn to attack. Assume his weapon damage this turn is 6; then his ATK = 14 + 6 = 20. This attack reduces the Skeleton's HP to 9 - 20 = -11.
Since the Skeleton's HP is less than 0, the Skeleton is defeated. The Paladin has won the battle!

So when a player attacks, we determine the ATK value in this way:

1. Generate a random number in the range of the weapon damage.
    a. For example, if the character's weapon has a damage attribute of 3 - 7, generate a random integer in the range 3 - 7 (inclusive).
2. Add the generated number to the character's Strength; this sum is the ATK value for the attack.

Items

The player can acquire items. We have two specific items in this game:

1. Healing Potion
    a. This item will add 10 to the player's HP.
2. Ring of Strength
    a. This item will add 5 to the player's Strength.

Program Requirements

Here is how your program should run.

a. When the program begins, prompt the player to choose a character.

```
Adventure Game - Start!

Here are the characters.
1. Rogue
2. Paladin
3. Jackie Chan

Which character do you choose?:
```

The player will then enter 1, 2 or 3. (You may assume that the player will enter a valid number.) You should then store the chosen character's name, HP, Strength, and weapon damage (HINT: store the min and max values of the range).

After the player chooses a character, print out a message to the player with the name of the chosen character.

For example,

```
You chose: Rogue
```

b. Print out some narration text, then prompt the player to choose a path. The player will enter 1 or 2 (again, assume a valid input).

```
The Evil Wizard must be defeated!  He is in The Castle.  To get to
The Castle, you must travel through either:
1. The Forest
2. The Graveyard

Which path will you take?:
```

c. If the player goes through The Forest, she will battle 3 Goblins. If she instead travels to The Graveyard, she will battle 5 Skeletons.

After the player chooses a path, print a message to the player with the name of the chosen path.

For example,

```
You chose: The Forest
```

Then print the following narration text:

```
Once you enter The Forest, you encounter 3 Goblins!  Time for battle!
```

If the user chooses The Graveyard, the message will instead be:

```
Once you enter The Graveyard, you encounter 5 Skeletons!  Time for
battle!
```

d. Simulate a battle with either 3 Goblins or 5 Skeletons (depending on the player's path).

For each battle, print the results as illustrated by the example below.

```
***Rogue vs Goblin 1***
Rogue attacks with ATK = 8 + 4 = 12
Goblin HP is now 25 - 12 = 13

Goblin attacks with ATK = 4 + 2 = 6
Rogue HP is now 55 - 6 = 49

Rogue attacks with ATK = 8 + 3 = 11
Goblin HP is now 13 - 11 = 2

Goblin attacks with ATK = 4 + 1 = 5
Rogue HP is now 49 - 5 = 44

Rogue attacks with ATK = 8 + 3 = 11
```

```
Goblin HP is now 2 - 11 = -9
Rogue defeated Goblin 1!

***Rogue vs Goblin 2***
Rogue attacks with ATK = 8 + 1 = 9
Goblin HP is now 25 - 9 = 16

Goblin attacks with ATK = 4 + 2 = 6
Rogue HP is now 44 - 6 = 38

Rogue attacks with ATK = 8 + 1 = 9
Goblin HP is now 16 - 9 = 7

Goblin attacks with ATK = 4 + 3 = 7
Rogue HP is now 38 - 7 = 31

Rogue attacks with ATK = 8 + 2 = 10
Goblin HP is now 7 - 10 = -3

Rogue defeated Goblin 2!

***Rogue vs Goblin 3***
Rogue attacks with ATK = 8 + 4 = 12
Goblin HP is now 25 - 12 = 13

Goblin attacks with ATK = 4 + 2 = 6
Rogue HP is now 31 - 6 = 25

Rogue attacks with ATK = 8 + 3 = 11
Goblin HP is now 13 - 11 = 2

Goblin attacks with ATK = 4 + 3 = 7
Rogue HP is now 25 - 7 = 18

Rogue attacks with ATK = 8 + 1 = 9
Goblin HP is now 2 - 9 = -7

Rogue defeated Goblin 3!

--Rogue wins the battle!--
```

Battling Skeletons will be similar to the above except there are 5 Skeletons, and Skeletons have a different Strength and weapon damage.

e. In the example above, the player won the battle; but what if the player was defeated instead?

If the player is defeated in battle, replace the last line of text in the example above with the following:

```
--Rogue is defeated in battle!--

GAME OVER
```

Your program should then terminate.

f. If the player wins the battle by either defeating the 3 Goblins or the 5 Skeletons, then display the player's current HP value and then present a reward to the player as follows.

```
Your HP is: 18

Please choose a reward.
1. Healing Potion
2. Ring of Strength

Which item do you choose?:
```

(You may assume that the player will enter a valid input.)

If the player enters 1 - i.e. selects the Healing Potion - then add 10 to the player's HP and print a message that is similar to the following:

```
You chose: Healing Potion

Your HP has increased to 18 + 10 = 28!
```

If the player enters 2 - i.e. selects the Ring of Strength - then add 5 to the player's Strength and print a message that is similar to the following:

```
You chose: Ring of Strength

Your Strength has increased to 8 + 5 = 13!
```

g. Tell the player that he has reached The Castle and will now begin battling The Evil Wizard.

```
You have now reached The Castle!  Time to battle The Evil Wizard!
```

Now, battle with The Evil Wizard is different from battle with Goblins and Skeletons.

The player may attack The Evil Wizard with his weapon, just like when battling Goblins or Skeletons, but the player may also choose to attempt to cast a spell that will reduce The Evil Wizard's HP to 0.

To cast the spell, the player must correctly guess a randomly-generated integer that is from the range 1 - 5.

So what your program needs to do before the battle with The Evil Wizard begins is this: generate a random integer in the range 1 - 5 and store this value. During the battle, if the player chooses to attempt a spell cast, the player will enter a number as a guess; if the two numbers match, then the spell cast is successful, and that means The Evil Wizard's HP goes down to 0.

If the spell-cast attempt is unsuccessful, display a message such as:
`The spell cast failed!  No damage is dealt to The Wizard!`
and then end the player's turn.

During this battle, the player will get to choose during each turn if she wants to attack or attempt a spell cast, so your program needs to prompt for this choice whenever it is the player's turn in the battle.

Here is an example battle. Assume the Rogue's HP is 23 and that the random number for the spell cast is 3:

```
***Rogue vs The Evil Wizard***
Choose your action:
1. Attack
2. Attempt Spell Cast

What would you like to do?: 1

Rogue attacks with ATK = 8 + 3 = 11
Wizard HP is now 40 - 11 = 29

Wizard attacks with ATK = 8 + 7 = 15
Rogue HP is now 28 - 15 = 13

Choose your action:
```

```
1. Attack
2. Attempt Spell Cast

What would you like to do?: 2

Enter your guess: 3
Correct!

The Rogue's spell is cast successfully!  The Wizard's HP is now 0!

--Rogue wins the battle!--

You win!  Congratulations!
```

Your program should then terminate.

If the player is defeated in battle, replace the last 2 lines of text in the example above with the following:

```
--Rogue is defeated in battle!--

GAME OVER
```

Your program should then terminate.

Project Requirements

1. The game must run with the MARS simulator and be implemented as a MIPS assembly language program.

2. Your output messages should match the ones described here and in the sample output.

3. The project is an individual effort. You may ask other students how they are approaching the problem, but you must **not** work together on the coding.

Project Submission

You must submit the following two files:

- **adventure-game.asm**
  This file should be your main project file.

- **README.txt**
  This file should describe any procedures, macros, and constants that you use. Also, it should describe your method for storing the player and enemy attributes. And finally, describe any known bugs in your code.

If you break up your source code into multiple files, be sure to include all files in your submission.

Submit your file(s) on CourseWeb using the project link.

Hints and Suggestions

1. It is strongly recommended that you map out your approach in pseudo-code before you begin writing your MIPS solution.

2. Another strong suggestion is to use **constants**, **procedures**, and **macros**.

3. How to generate random numbers in MIPS?

Use syscall 40 to seed a random number generator, and then use syscall 42 to generate a random number in a specified range.

Below is some MIPS code to seed the random number generator with the current timestamp. Note that if you want to consistently generate the same random sequences, you can do this by setting the seed value, stored in $a1, to a constant value (such as 0).

```
# get the time
li   $v0, 30         # get time in milliseconds (as a 64-bit value)
syscall

move $t0, $a0        # save the lower 32-bits of time

# seed the random generator (just once)
li   $a0, 1          # random generator id (will be used later)
move $a1, $t0        # seed from time
li   $v0, 40         # seed random number generator syscall
syscall
```

Once we seed the generator, we can generate random numbers with syscall 42, which returns a random number in a specified range. Note that the random generator id that we set above is also set below to the same value. Since you will be making this syscall MANY times in your program, I recommend creating a macro, such as:

```
.macro get_random_int %upper_bound
     li $v0, 42
     li $a0, 1
     li $a1, %upper_bound
     syscall
.end_macro
```

Note that this macro has a parameter for the upper bound of the range, where the range starts at 0. But often times we want a range such as (1 - 4) or (2 - 5). To achieve this, you can create a macro that has two parameters, one for the lower bound and one for the upper bound.