AP Computer Science Final Project - README Template

Instructions:
The first step in creating an excellent APCS final project is to write up a README. At this stage, this README file acts as your **project proposal**. Once you've filled in all components, Shelby will read through it and suggest edits. Ultimately, you need a document that adequately describes your project idea and **we must agree on this plan**.

Have one member of your group **make a copy of this Google Doc**. Then, they should share it with all other members **and with Mr. Shelby** so that every group member has edit permissions, and Shelby can add comments on your ideas.

There's a lot of parts of this document that you might not have full answers for yet. Because you haven't written the program yet, it's difficult to think about the **instructions** or **which group members will do which parts**. Even though this is hard to think about, you must have something in these sections that acts as your current plan. However, during the course of the project, you'll **continuously update this document**. This means that you will not be *held* to exactly what you put here - components of this document can change (and it's pretty common!).

There is one exception: the **Features List** section. Once Shelby OKs your README, the Features List section **cannot be modified**. For this reason, it is most important that you get a solid idea of what you want to make and the primary features it will have *now*.

Talk with your group. Consider drawing some pictures of what you think your project might look like. Be precise. When you're ready, fill this out together. Each component in brackets below ([these things]) should be replaced with your ideas. Note that there are several sample READMEs posted on this assignment for you to use as guidance.

----------When README is finalized, remove everything above this line----------

Robot Death Arena

Authors: Deeptesh Day and Tyler Tamura
Revision: April 15, 2022

Introduction:
[In a few paragraphs totaling about ½ page, introduce the high-level concept of your program. What this looks like depends a lot on what type of thing you are making. An introduction for an *application* will look different than one for a *game*. In general, your introduction should address questions like these:
What does your program do?

What problem does it solve? Why did you write it?
What is the story?
What are the rules? What is the goal?
Who would want to use your program?
What are the primary features of your program?]

Welcome to Robot Death Arena, a fighting game with futuristic fights that are expected to happen 100 years later. There are two finalists(users) who are trying to compete with each other to gain the Robot Death Arena championship trophy. The main objective for each user is to knock out the other user with the weapons they choose to fight with. When the game begins, the user goes to a screen where he can choose weapons and armors, there are three different types of weapon namely: sword,spear and hammer. Each weapon has its own advantages and disadvantages, for example spears increase speed but lead to lower damage. Similarly there are three types of armor namely: Light, Medium , Heavy. The armors also have advantages and disadvantages, they affect the robots stats. Stats are numerical representations of a robot's strength, speed, health. Our program is really attractive to people who love WWE fights and fighting type games. The program uses WASD to move around and a mouse to attack. When the user clicks on the attack button in the mouse, the attack is automatically aimed at the space in front of where the user is facing. Our program specializes in real life situations of a fight through stats, for example a person could run faster with a sword than a hammer which means weapons affect speed, this real life to game transformation is our first and foremost priority when making and creating new things in the game. If the user doesn't customize his robot, his robot will have a default customization that is the first weapon and the first armor during battle. There will be a help button at the bottom left corner of the screen where there is going to be an instruction button in which the user could re-learn all the movement instructions.The game shows different types of animations for different types of weapons, for example if someone uses a hammer, the user can see hammer swinging and if there is a sword, the user can see the sword swinging. When the user hits the enemy user with their weapon, there is a bounce which means that the other character gets pushed back hence they have a chance to escape to not take more damage or to counterattack. After each slash by any weapon there is a delay of about 2 seconds before the user can attack again hence the user needs to be careful about attacking and cannot just randomly spam the attack button. The game also has special abilities that could be cast during battle, using special abilities brings up a new animation in which the special ability shows up, for example if the special ability is fireball then the game shows a fire type projectile going towards the enemy user. When a person presses the movement key, the character robot moves but it doesn't have any special animations for walking. The weapon can only attack a target in front or below the opponent, which means that a user could dodge an attack and then attack. The user could use the defend button to neglect all damages dealt if times are correct but the defend function has a cooldown of 4 seconds. The main rules for the program is to not cheat via hacking or disrupting the other person's movement and respecting the other people during gametime.

Instructions:
[Explain how to use the program. This needs to be **specific**:
Which keyboard keys will do what?

The w key lets the user jump,the s key lets the user go down if there is no obstacle, the d key lets the user to move right, the a key lets the user to move left.
Where will you need to click?
To attack the user needs to left click on the mouse.
Will you have menus that need to be navigated? What will they look like?
We have a menu that lets the user choose the attack and armor option, and they are at the top of the program with weapons and armor click buttons on them.
Do actions need to be taken in a certain order?]
The user needs to click on the battle button but after that there is no need for the user to move in a certain order.

Features List (THE ONLY SECTION THAT CANNOT CHANGE LATER):

Must-have Features:

[These are features that we agree you will *definitely* have by the project due date. A good final project would have all of these completed. At least 5 are required. Each feature should be fully described (at least a few full sentences for each)]

- Pre Battle Screen: When the program starts, there is gonna be a robot standing and a button in left that has weapon and armor options. The screen also has a battle option that has a top down view, when pressed sends the user into a waiting zone like a lobby.
- Lobby Screen: During the lobby screen, the user can look at their robot and make last minute changes. The user can press an exit button that takes them back to the pre-battle screen.
- Custom Weapons & Armor Slots: During the Pre Battle Screen, the user can select from three different armor types: Light, Medium, and Heavy. Lighter types will reduce the hit points of the robots and increase speed while the heavier types increase hit points and reduce speed. Medium is a mix of both. The user can also select three weapons: Sword, Spear, Hammer.
- 2 Person: Both users will be able to battle each other. Multiplayer will use LAN to connect 2 computers and set them into battle.
- Movement (wasd): A for right, D for left, W for jump, S for crouch.
- Stats Bar: Just like many fighting games, both health bars of each player will be displayed on the top of the screen. The hp will be listed on the health bar. Each player will have 100 Hp as a base stat but this hp could be more for armors that give more hp.
- Animations for Attacks: The attacks have their own animation and states the damage above each attack. For example, a sword slash would have an animation of a sword slashing, the above the attack, it would state the damage done.
- Robot Stats: The robot has 4 stats: hp, damage, reload, speed. There is gonna be a button called stats at the top right of the screen. Before going to the waiting lobby, there is going to be a space at the end of the screen that specifies the robots damage and speed during battle.
- Special Abilities: Special Abilities are essentially special attacks that are obtained by storing up energy. This happens by pressing ‘c’ during the battle, which prevents the user from moving. Special abilities are kind of like an ultimate move, it could be teleporting, meteor strike or fireball basically any kind of magic. Would also have its own animation. It would be selected at the pre-battle screen.

Want-to-have Features:

[These are features that you would like to have by the project due date, but you're unsure whether you'll hit all of them. A good final project would have perhaps half of these completed. At least 5 are required. Again, fully describe each.]

- Background Music/Sound Effects: The background music would be different depending on the screen. Each screen would have its own song. Attacks would make a sound and special abilities would have their own sound.
- Different types of maps: These would purely be background images during the battle screen. The selected map would be randomly selected.
- More Stats other than Health, Speed, Attack: This could be a dodge stat, damage reduction, ect.
- More movement options like slide and kick. QA/QD for slide, EA/ED for kick. The slide would move the player, function as a duck and knock back the other player. The kick is a longer ranged attack but leaves the user "open to attacks" for a small duration after.

Stretch Features:
[These are features that we agree a *fully complete version of this program would have, but that you probably will not have time to implement*. A good final project does not necessarily need to have any of these completed at all. At least 3 are required. Again, fully describe each.]

- 3D animations: This would mean that all the robot sprites are replaced with 3d animations. It would use knockback mechanics.
- Cutscene for special abilities: When a special ability is going to activate, it would create a cutscene. The music would dampen to make room for the sound effects.
- Robot choice that have different abilities: During the pre-battle screen, the user can select between multiple robots. There would be a button to change the robots to another one. The robots would have different abilities like flying, mega-jumping, dash, ect.

<u>Class List</u>:
[This section lists the Java classes that make up the program and very briefly describes what each represents. It's totally fine to put this section in list format and not to use full sentences.

- Main
- DrawingSurface
- Robot
- Weapon
    - Sword
    - Spear
    - Hammer
- Armor
    - Light
    - Medium
    - Heavy
- Ability

        - Meteor
        - Kamehameha
- Screen
    - PreBattleScreen
    - LobbyScreen
    - BattleScreen
- PlayerData
- Player

Credits:
[Gives credit for project components. This includes both internal credit (your group members) and external credit (other people, websites, libraries). To do this:

- List the group members and describe how each member contributed to the completion of the final program. This could be classes written, art assets created, leadership/organizational skills exercises, or other tasks. Initially, this is *how you plan on splitting the work*.
  Deeptesh Dey will work on the gui part of the program whereas Tyler will work on the algorithmic part of the program.

- Give credit to all outside resources used. This includes downloaded images or sounds, external java libraries, parent/tutor/student coding help, etc.]
  Processing

Tyler:

- I thought of the initial idea
-

Deeptesh:

- I came up with a clash royale type game even though it wasn't the one we ended up doing.
-