Michael Ferrer, Nathan Joslin

John David Newman & Brian Seth

Mr. Armel

CS 142

June 4, 2022

# Group Project Write Up "Zombie Highschool Adventure Game"

Zombie Highschool Adventure game is a Java game based on the Michael Ferrer's idea for a final group project. The setting is a high school in a nondescript town that has been overrun by brain eating zombies.

The game starts out by having the player choose from one of four characters: The Athlete, the Brainiac, the Bully, the Class President, and the Cheerleader. Each character has a preset value for power, speed and smarts that are different from one another based on there stereotype. The Bully has more power while the Brainiac has more smarts (intelligence). The values range from 1 to a maximum of 10.

After selecting the character, the player is given a series of two choices as they navigate through the halls and classrooms. The choices made by the player, including character selection and some random luck (dice rolling) give them the opportunity to beat a zombie goon and zombie boss.

When the player encounters a zombie or zombie boss the player rolls a die. Depending on the baseline attributes of the player and their opponent (zombie), as well as the die roll, the player either beats the zombie or suffers defeat.

The ultimate outcome of the game being either escaping with their lives or suffering an unfortunate death as zombies feast on their gray matter.

The initial version of the program consists of a player character class containing the blueprint for the characters. This includes character, power, speed, smarts, character name, diceRoll method and importing the Random java utility. A separate class was created for the zombie and zombie boss that contains similar components.

The main method does all the heavy lifting in this program. User input is key to this game, thus a scanner utility is imported and used. Input validation is required. As the player is given only two choices and a dice roll is part of the play, the code is abundant with if, nested ifs, and else statements as well as Booleans. The use of loops and break statements were limited. The game does not rely heavily on methods compared to earlier major assignments. The methods were essentially limited to a promptNumberReadline (used in the main), toString and diceRoll (in the classes) and the conventional getter and setter methods.

With the game being Michael's idea and heavily influenced by Nathan, these two tackled most of the coding and debugging. Michael and Nathan co-led the project from its concept, considering potential limitations and requirements, layout, and finally, the actual coding. Jonathan brought in a revision to include a much-needed difficulty setting and offered variants to the combatants the player would encounter, while Brian put the write up together with input from the team.

Surprisingly, there were only a few minor modifications from the original plan to the final product that was ultimately delivered. This was likely due to Michael and Nathan's understanding of the time constraints (project due date, inherent slower speed of projects with group collaboration) as well as their

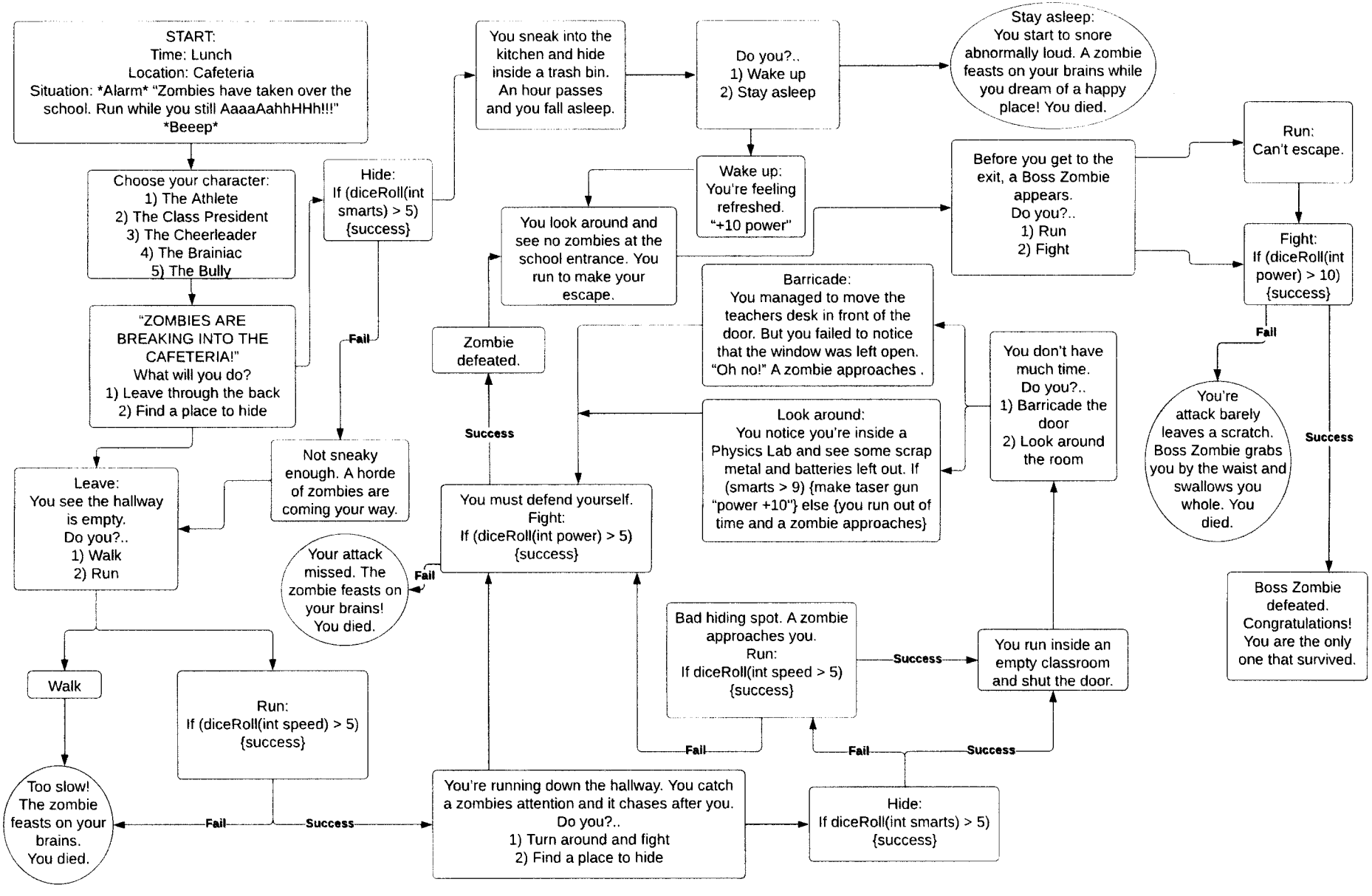
adept programming abilities. The program was not overly ambitious, and the team incorporated the concepts, methods and material covered in earlier assignments and throughout the course. While the initial stated objectives were met and there were only a few, limited deviations (moving some code from class to main, confining special abilities to increase baseline attributes, instead of their own variables and adding a difficulty setting) the group has more lofty ambitions for the game.

Overall, for our first group programming assignment ever, things went rather well but there are certainly components of the project that could be improved. Practical ideas on how future work could proceed are broken out into two key areas. The game coding itself and the workflow development.

John's idea for zombie variants and more encounters with them, being one already in the works. Implementing a weapon system and inventory (arrays) would be an improvement. Game mechanics can be altered, where instead of rolling higher than the stat value, we can add hit points. In addition to this, adding animation, and extending the game play and a more comprehensive plot or storyline would afford the end user a more enjoyable gaming experience.

With respect to project/workflow enhancements, the use of a centralized repository for the program would be beneficial, affording the team members the ability to update/edit and debug the program in real time. This would eliminate the need to send files via email and meet in person. The use of a communication platform (Discord was the intent, but most communication was done via Canvass) to post questions, comments and directives would allow for open and direct discussion.

All in all, the project was conceived, the work was done, and the program delivered on time and as required. Every member of the team did their share of the work needed from start to finish. We hope you enjoy Zombie Highschool Adventure Game".


START:
Time: Lunch
Location: Cafeteria
Situation: *Alarm* "Zombies have taken over the school. Run while you still AaaaAahhHHh!!!" *Beeep*
Choose your character:
1) The Athlete
2) The Class President
3) The Cheerleader
4) The Brainiac
5) The Bully
"ZOMBIES ARE BREAKING INTO THE CAFETERIA!"
What will you do?
1) Leave through the back
2) Find a place to hide
Hide:
If (diceRoll(int smarts) > 5) {success}
Fail
Not sneaky enough. A horde of zombies are coming your way.
Leave:
You see the hallway is empty.
Do you?..
1) Walk
2) Run
Walk
Too slow! The zombie feasts on your brains. You died.
Run:
If (diceRoll(int speed) > 5) {success}
Fail
Success
You're running down the hallway. You catch a zombies attention and it chases after you.
Do you?..
1) Turn around and fight
2) Find a place to hide
You must defend yourself.
Fight:
If (diceRoll(int power) > 5) {success}
Fail
Your attack missed. The zombie feasts on your brains! You died.
Success
Zombie defeated.
You look around and see no zombies at the school entrance. You run to make your escape.
You sneak into the kitchen and hide inside a trash bin. An hour passes and you fall asleep.
Do you?..
1) Wake up
2) Stay asleep
Stay asleep:
You start to snore abnormally loud. A zombie feasts on your brains while you dream of a happy place! You died.
Wake up:
You're feeling refreshed.
"+10 power"
Barricade:
You managed to move the teachers desk in front of the door. But you failed to notice that the window was left open. "Oh no!" A zombie approaches .
Look around:
You notice you're inside a Physics Lab and see some scrap metal and batteries left out. If (smarts > 9) {make taser gun "power +10"} else {you run out of time and a zombie approaches}
You don't have much time.
Do you?..
1) Barricade the door
2) Look around the room
Bad hiding spot. A zombie approaches you.
Run:
If diceRoll(int speed > 5) {success}
Fail
Success
You run inside an empty classroom and shut the door.
Fail
Success
Hide:
If diceRoll(int smarts) > 5) {success}
Before you get to the exit, a Boss Zombie appears.
Do you?..
1) Run
2) Fight
Run:
Can't escape.
Fight:
If (diceRoll(int power) > 10) {success}
Fail
You're attack barely leaves a scratch. Boss Zombie grabs you by the waist and swallows you whole. You died.
Success
Boss Zombie defeated. Congratulations! You are the only one that survived.