Sentient Turtle

Authors: Byron, Luke, Tapish
Revision: 5/6/2

Introduction:

You are a sentient turtle hailing from a long line—the only line—of sentient turtles. However, it is the 21st century, and animal trafficking is rampant across our society. Those grubby animal traffickers caught you in your sights—not because you are sentient, but because turtles are in high demand and they must pay the bills somehow (have you seen the gas prices???). They sack you and chuck you in their turtle dungeon. Most turtles resign themselves to their fate, as they literally do not know any better, but you're different. You'll find a way out.

Thing is, once the humans chuck you in the dungeon, they themselves need to get out. However, because they're really paranoid about their product escaping, their exit consists of a labyrinth of word riddles that once solved give them the keys to the exit. A bulletproof design to prevent turtles escaping—but you're not a normal turtle. You've got the brains to get out. You will scour the room, solving the riddles one-by-one, perhaps collecting a few items necessary to escape along the way, to get out. Do you have what it takes to escape? Or will your sentience prove lacking, helping to seal your fate?

What will happen once you escape, you may ask? Well, let's worry about the escaping part first. Fire up those problem-solving parts of your brain, as your very life depends on it.

Instructions:

- Starting Menu:
    - Upon starting the game, you will be greeted with the following buttons:
        - Click "Controls" to view how to control the turtle, interact with items, etc. Press exit to close out of the pop up.
        - There will be a dropdown menu with two options: Easy or Hard
            - Easy and Hard will differ in the amount of riddles as well as the riddle difficulty. Unlike Easy mode, you will have to escape under a certain time limit.
        - After selecting the difficulty, click "Play" to start the game.
- After clicking "Play", the screen will display the story behind the turtle as well as mention the objective of the game: escape. The player can then press the "Esc" key to start playing the game itself. At this point, the player can take control of the character.
- Escape Room:
    - The character, Turtle, is placed into a room that appears to be heavily locked.
    - Use the arrow keys to move the sprite.
    - Chests that contain riddles are placed around the map.
        - To interact with chests, press the spacebar when you are on top of it.
        - You will be given a riddle which you have to enter your response.
        - Each time you answer incorrectly, 5 seconds will be added to your time in easy mode, and 5 seconds will be taken away in hard mode.

        - After guessing incorrectly 3 times, you will be given an option to see a hint, which upon accepting, will add or subtract 30 seconds from your time, depending on the difficulty.
        - Answering these riddles correctly will give you a key (the total number of keys you have can be viewed in the upper left corner).
    - At any point in the game you can press Esc to pause the game (stopwatch/timer as well) and a popup will appear. Press “Continue” to resume or “Exit to Menu” to return back to the starting menu.
- After collecting the required keys you will be able to open the main door and find your way to freedom. Your point total will be shown on the screen and there will be a button to redirect you back to the starting menu.

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

Must-have Features:

- Interactive Menu
    - When the player opens up the game, they will be greeted with the title screen along with the options to start the game after selecting a difficulty level (easy or hard) and to view controls (non-adjustable). This will be akin to a splash screen seen on other video games such as *Minecraft*, albeit with less complexity.
- Riddle sets + key collection
    - In order to win the game, you must collect a specific number of keys in order to escape the dungeon. In order to obtain said keys, you will need to complete riddles that can be found scattered throughout the dungeon, marked by either a sign or a chest (will be one of the two, haven’t decided yet). Once the riddle is solved, it will go into the player’s inventory, which will be marked with on-screen UI. Having any number of keys less than the required amount will not result in any gain for the player; the key system is akin to the Red Coin dynamics in Super Mario games: collect a certain number, and you’ll be able to unlock the door. The riddles will be ones where the player will be able to type in the answer to it, and they will be randomly chosen from a riddle bank.
- Two maps with different difficulties
    - There will be one map for easy and another for hard difficulty. The level design will not be drastically different between the two; the overall gameplay will still be the same. However, in the hard difficulty, the player will have to solve more riddles and may encounter more obstacles. Both of these maps will not be randomly generated; instead, the riddles for both difficulties will be randomly picked out of a riddle bank.
- Stopwatch/timer
    - Once the game starts, a stopwatch will start and measure how long it takes the player to complete the game (completion is marked by unlocking the exit door). After completion, a score will be displayed that will reward the player for efficiency. For example, if the player completes the game quickly, they will get a higher score. This dynamic will be reversed in the hard difficulty, i.e. the more time left on the timer, the more points they will be able to earn.
- Victory screen

    - A screen that appears after successful completion of the game. Will show a picture of you, the turtle, chilling on the beach, in addition to giving information about the game run completed, specifically score (as mentioned above) and time spent.

Want-to-have Features:

- Multiple levels for each difficulty

  There will be multiple levels of difficulty involved in the game. The difficulties are picked by the player, and each level would have its own unique set of problems to solve that correspond with the difficulty level. The opening section should be clear and very easy to identify, and by clicking on the level it should take you directly into the corresponding game.
- Leaderboard (stores previous attempts to beat the game)

  There will be a leaderboard up for display, showing each of the fastest times/attempts at the game for each level that is previously achieved. This will serve as a measurement of skill for comparison by players themselves.
- Live Scoreboard (rewards the character points for certain actions)

  A simple score tracker that can show how many points a player has earned during the duration of the game. Committing certain actions would reward you points, and a player can keep track of how they themselves are doing during the progress of the game.
- Pop riddles (riddle has to be answered on the spot)

  While walking around the map, at random, the player may encounter pop riddles, which are riddles that pop up and must be answered correctly on the spot. If they fail to do so,
- Enemies that kill the character on contact (will move in a certain path)

  The enemy will be placed on different points of the map. When a player runs directly into one, the game will end, as the turtle's dead and therefore unable to escape. Enemies will be programmed so that they move in a certain direction and pattern repeatedly, and the player must avoid them or maneuver to evade them.
- Coding challenges (quick prompts similar to pop riddles)

  In this world where STEM (or perhaps computer science) is getting ever more and more popular, it wouldn't hurt to get some coding proficiency before you escape. These will be similar to riddle boxes and pop riddles, but they will have to do with coding. They will spawn in regular riddle markers and the turtle must respond with a CS-related answer.

Stretch Features:

- Randomly generated levels
    - A randomly generated level with a direct entrance/start and exit/end with chests placed randomly throughout the map would make the game much more enjoyable, and gives the sense of freshness everytime the game is replayed. The randomizer will operate in such a way that the levels will always be able to be completed, meaning that the enemies and obstacles will not make game completion impossible.
- Character customization

    - Would allow the player to “dress up” the turtle and give them a unique look. Can improve immersiveness of the game and give more power of expression to the player. Possible monetization schemes can also be implemented.
- Infinite map/Sandbox
    - Instead of having the objective of completing the level, there could be an infinite map/sandbox mode where the player is able to solve as many riddles as they possibly can, all while evading enemies. This feature is most likely the hardest to implement out of all the stretch features, due to the immense number of riddles that would need to be coded into RiddleBank.

Class List:

- GameScreen
    - Screen that has the map of the game itself
- Sprite
    - Super class for all sprites present on map (obstacles, riddle markers, turtle character). Will hold basic information common to all sprites (such as x and y coordinates).
- Obstacle
    - Obstacles that will be spawned on the map that the player cannot pass. Will be randomly spawned.
- RiddleMarker (will be renamed to Chest or Sign once we decide on what will mark riddles on the map)
    - Class for the sprite that the player can interact with to solve riddles. Will *not* be randomly spawned on the map, instead will be hard-coded based on the map difficulty configuration.
- Turtle
    - The turtle avatar that the player controls. Will also handle player inventory keeping track of the keys the player has.
- RiddleBank
    - Class that serves as a “riddle bank”, where a random number generator will pick riddles from (without repetition)
- Screen
    - Class that handles all of the screens needed for the game (home screen, game itself, victory pop-up)
- SplashScreen
    - Main home screen
- FinishScreen
    - Screen upon which the game concludes
- Main
    - Main class that runs the program
- DrawingSurface
    - Class that handles all inputs, displaying, etc.

Credits:

Tapish: Project lead, coded skeleton code, set up packages/classes in project folder, wrote Introduction and Class list, in addition to must-have features
Byron: wrote Instructions, in addition to writing JavaDocs
Luke: N/A