

Project 3 - Dragon Game (Part 1)

- **Player (ACTOR)**
 - Goal And Priority
 - Goal: The goal for the player is to survive until the end of the “hall”. The player will die if they run out of any of their three given attributes: *intelligence, time, and money*.
 - Priority: The priority for the player is to survive. If manageable, it is in the player’s best interest to survive with the highest “score” possible, which is determined by the amount of the three attributes multiplied together.
 - Functionality Scope
 - Attributes of the player include *intelligence, time and money*. All attributes will start in the random range 8-25 (inclusive, I assume) at the start of each game.
 - The conditions required for the player to survive are all attributes staying above 0.
 - The player’s score is calculated by multiplying all three of the attributes; values together.
 - Identify Relationships
 - Association: The player has an association with all of the use cases in this game, which are those in a player’s turn, such as:
 - **Move (Use Case)**
 - Goal and Priority
 - Goal: The goal of the “move” use case is for the player to progress one step forward in the hall, advancing towards the goal.
 - Priority: The priority is to advance while keeping track of your attributes without falling to 0.
 - Functionality Scope
 - The player moves one step in the hall.
 - A “move” action may cause an encounter of a particular event, including a puzzle or other encounters (Encounter in detail is listed in a bullet point later on)
 - Moving consumes a unit of time.
 - Relationship
 - The “move” use case is associated with the player, as the player directly chooses to move in order to progress in the hall.

- Extension and Inclusion
 - The “move” use case includes an extension to the encounter section that creates additional steps based on the particular encounter itself.
 - The “move” use case includes decrementing the time attribute, as taking a step forward takes a unit of time from the player’s time attribute.
- Multiplicity
 - The “move” use case has a multiplicity of zero to one, because a single player can choose whether or not to move once in a turn, taking a single step.
- Notes (“Move”)
 - Ensure the time is decremented from the player’s time attribute after a step is taken
- Encounters (Use Case - Extension to “Move”)
 - Goal and Priority
 - Goal: To present the player with a specific event.
 - Priority: Successfully provide the player with an encounter, whether nothing happens or whether an event happens, after they move.
 - Functionality Scope (Given by probability)
 - 25% Chance: Nothing happens. Player moves forward one step. Time is taken away.
 - 30% Chance: Puzzle encountered. Player needs to interact further
 - Look to puzzle bullet under encounter for more information)
 - 10% Chance: Encounter professor
 - Includes Losing Time Attribute
 - Includes Gaining Intelligence Attribute
 - 10% Chance: Encounter graduate student.
 - Includes Losing Time Attribute
 - 15% Chance: Attacked by grunt work!
 - Includes Losing Time Attribute
 - Includes Losing Intelligence Attribute
 - 10% Chance: Grade papers.
 - Includes Losing Time Attribute
 - Includes Gaining Money Attribute
 - 2.68% Chance: You win the lottery.
 - Includes Gaining A LOT of Money Attribute

- 0.001% Chance: You were robbed of all your money.
 - Includes Money Attribute reaching 0.
 - Includes a message that tells you how you died:
 - “A thief has robbed you of all your money! You have gone broke!”
 - Includes the original “You Lose” screen.
- Relationships/Extension
 - Encounter is an extension of the “Move” base use case.
- Multiplicity
 - There is one encounter for one execution of the “move” use case.
- Notes (“Encounter”)
 - Make sure that the outcomes of each encounter reflect the player’s attributes by the end of the turn.
- Puzzle (Use Case - Extension to “Encounters”)
 - Goal and Priority
 - Goal: Present the player with a riddle or puzzle that requires a response
 - Priority: Enter a response that will either benefit the playing (gaining attributes) or penalize the player (losing attributes)
 - Functionality Scope
 - Includes various puzzles and riddles
 - Each puzzle has a correct and incorrect answer that will give the player a reward or a loss of attribute value.
 - Examples
 - “What gets wetter as it dries?” (Correct answer is: “towel”; Int Att +3; Int Att -3 otherwise)
 - “What is the capitol of Alabama?” (Correct answer is: “Montgomery”; Int Att +1; Int Att -5 otherwise)

- “What is the federal US minimum wage?” (Correct answer is: “\$7.25”; Money Att +7.25; Money Att -7.25 otherwise)
 - Relationship/Extension
 - Puzzle is an extension to “Encounters” use case.
 - Multiplicity
 - There is 0 to 1 puzzles for every one encounter. There may or may not be a puzzle from an encounter.
 - Notes (“Puzzle”)
 - Make sure the puzzles are challenging yet fair to the player, with the appropriate rewards or losses based on the player’s response.
- Read technical papers (Use Case)
 - Goal and Priority
 - Goal: The goal is to boost the player’s intelligence attribute
 - Priority: Increase the player’s intelligence attribute.
 - Functionality Scope
 - Player selects the option to read papers
 - Player loses fixed amount of time
 - Player gains a random amount of intelligence w/in a reasonable range
 - Relationships
 - This use case is associated with the player, since the player chooses this action
 - Inclusion
 - There is an inclusion that increases the player’s intelligence attribute.
 - Multiplicity
 - There is a multiplicity of 0 to 1 for one player, as it can choose whether or not to read technical papers in a turn.
- Search for loose change (Use Case)
 - Goal and Priority
 - Goal: The goal is to boost the player’s money attribute
 - Priority: Increase the player’s money attribute.
 - Functionality Scope

- Player selects the option to search for loose change.
 - Player loses fixed amount of time
 - Player gains a random amount of money w/in a reasonable range
- Relationships
 - This use case is associated with the player, since the player chooses this action
- Inclusion
 - There is an inclusion that increases the player's money attribute.
- Multiplicity
 - There is a multiplicity of 0 to 1 for one player, as it can choose whether or not to search for loose change in a turn.
- View character (Use Case)
 - Goal and Priority
 - Goal: To allow the player to view their character's attributes and current position in the hall.
 - Priority: To accurately display the requested information
 - Functionality Scope
 - Player selects viewing their character
 - The player is given a display of their attributes and their position in the hall
 - Relationships
 - This use case is associated with the player, since the player chooses this action
 - Multiplicity
 - There is a multiplicity of 0 to 1 for one player, as it can choose whether or not to view the character.
- Quit the game (Use Case)
 - Goal and Priority
 - Goal: The goal is to allow the player to quit the game.
 - Priority: To end the game for the player.
 - Functionality Scope
 - Player selects to quit the game
 - Game displays "You Lose" message
 - Game ends and exits.
 - Relationship
 - This use case is associated with the player, since the player chooses this action

- Inclusion and Extension
 - Includes displaying a message “You Lose”
 - Multiplicity
 - There is a multiplicity of 0 to 1 for one player, as it can choose whether or not to quit the game.
- Multiplicity
 - The player is the only role in the game at any given time.
- Actor
 - The player is an actor
- Important Notes
 - Make sure the player’s attributes are accurately tracked and updated through each turn in the game.
 - Make sure the game mechanics are in place to manage the player’s attributes and eventually calculate the score after each turn (in case a player were to die).
- **The Hall (Game Element (not an actor or use case))**
 - The hall is a linear path that is twenty (20) steps long.
 - The player will start at the beginning of the hall (which is step #0).
 - After a player moves a step, they should be told how many more steps are needed until they reach the goal.
 - If a player reaches the goal without any of their attributes falling below 0, they win, and their attributes are multiplied with each other to calculate their score.
 - The score will be displayed with a simple ASCII victory message.
 - If the player fails to reach the goal and dies beforehand, a “You Lose” message will appear with the cause of death.