Pokemon Battle Simulator Documentation

Humas Ayub

Design Documentation

Contents

High level requirements 3

Intermediate level requirements 3-4

Low level requirements 5-9

Verification 9-10

Design Justification 10

High level requirements

|  |  |
| --- | --- |
| Requirement ID | Description |
| HLR\_001 | The Pokemon Battle Simulator frontend should have an interactive terminal that supports latin characters |
| HLR\_002 | The Pokemon Battle Simulator frontend should have an interactive terminal that supports numerical characters |
| HLR\_003 | The Pokemon Battle Simulator frontend should send user interactive input data to the Pokemon Battle simulator backend |
| HLR\_004 | The Pokemon Battle Simulator backend should send battle information (Pikachu attack name, Charizard attack name, Pikachu attack damage, Charizard attack damage, round number, Pikachu health points, and Charizard health points) |
| HLR\_005 | The Pokemon Battle Simulator frontend should control the interactive terminal |
| HLR\_006 | The Pokemon Battle Simulator backend shall send an “Invalid move” exception to the Pokemon Battle Simulator frontend when the same attack is used twice in a row or a non exsistant attack name is selected |
| HLR\_007 | The Pokemon Battle Simulator backend shall send a “Battle Ended” exception to the Pokemon Battle Simulator frontend when Pikachu health points or Charizard health points are depleted |

Intermediate level requirements

|  |  |  |
| --- | --- | --- |
| Requirement ID | Description | Traceability  High-level  requirement/s |
| ILR\_001 | The Pokemon Battle Simulator frontend shall control the position of the displayed Pikachu attack damage (at the end of the first line) | HLR\_002  HLR\_004  HLR\_005 |
| ILR\_002 | The Pokemon Battle Simulator frontend shall control the position of the displayed Pikachu attack name (at the front of the first line) | HLR\_001  HLR\_004  HLR\_005 |
| ILR\_003 | The Pokemon Battle Simulator frontend shall control the position of the displayed Charizard attack damage (at the end of the second line) | HLR\_002  HLR\_004  HLR\_005 |
| ILR\_004 | The Pokemon Battle Simulator frontend shall control the position of the displayed Charizard attack name (at the front of the second line) | HLR\_001  HLR\_004  HLR\_005 |
| ILR\_005 | The Pokemon Battle Simulator frontend shall control the position of the Round number (third line) | HLR\_001  HLR\_002  HLR\_004  HLR\_005 |
| ILR\_006 | The Pokemon Battle Simulator front end shall control the position of the displayed Pikachu health points (fourth Line) | HLR\_001  HLR\_002  HLR\_004  HLR\_005 |
| ILR\_007 | The Pokemon Battle Simulator frontend shall control the position of the displayed Charizard health points (fifth line) | HLR\_001  HLR\_002  HLR\_004  HLR\_005 |
| ILR\_008 | The Pokemon Battle Simulator frontend shall control the position of the last Pikachu attack (sixth line) | HLR\_001  HLR\_004  HLR\_005 |
| ILR\_009 | The Pokemon Battle Simulator frontend shall control the position of the last Charizard attack (seventh line) | HLR\_001  HLR\_004  HLR\_005 |
| ILR\_010 | The Pokemon Battle Simulator backend should calculate Pikachu health points by subtracting the Charizard attack damage from the existing Pikachu health points | HLR\_004 |
| ILR\_011 | The Pokemon Battle Simulator backend should calculate Charizard health points by subtracting the Pokemon attack damage from the existing Charizard health points | HLR\_004 |
| ILR\_012 | The Pokemon Battle Simulator backend should select a Charizard attack based of the round number | HLR\_004 |
| ILR\_013 | The Pokemon Battle Simulator backend shall make the Pikachu health points equal to 0 if the Pikachu health points become a value of less than 0 | HLR\_004 |
| ILR\_014 | The Pokemon Battle Simulator backend shall make the Charizard health points equal to 0 of the Charizard health points become a value of less than 0 | HLR\_004 |
| ILR\_015 | The Pokemon Battle Simulator backend shall replay the round with a new user input if the same Pikachu attack is selected two times in a row | HLR\_005  HLR\_006 |
| ILR\_016 | The Pokemon Battle Simulator backend should replay the round with a new user input if a non-existent Pikachu attack is selected | HLR\_005  HLR\_006 |
| ILR\_017 | The Pokemon Battle Simulator backend should stop the simulation if Pikachu health points is equal to or less than 0 | HLR\_005  HLR\_007 |
| ILR\_018 | The Pokemon Battle Simulator backend should stop the simulation if Charizard health points is equal to or less than 0 | HLR\_005  HLR\_007 |
| ILR\_019 | The Pokemon Battle Simulator backend should determine Pokemon attack damage based of the users input data | HLR\_003 |

Low level requirements

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | Description | Traceability  High-level  requirement/s | Traceability  Inter-level  requirement/s |
| LLR\_001 | The function that sends an “Invalid move” exception to the Pokemon Battle Simulator frontend when the same attack is used twice in a row or a non exsistant attack name is selected shall have the following characteristics  Function prototype: def attack(self, attack\_name):  if attack\_name == self.last\_attack:  raise InvalidMove("Cannot use the same attack twice in a row.")  if attack\_name not in self.attacks:  raise InvalidMove("Invalid attack name.")  damage = self.attacks[attack\_name]  self.last\_attack = attack\_name  return damage  Function attack input arguments:  attack\_name – The attack selected by the user through the interactive terminal | HLR\_006 | ILR\_015  ILR\_016 |
| LLR\_002 | The code that sends a “BattleEnded” exception to the Pokemon Battle Simulator frontend when Pikachu health points or Charizard health points are depleted shall have the following characteristics:  if self.charizard.hp <= 0:  print(f"Pikachu used {pikachu\_attack\_name} and dealt {result[0]} damage.")  self.pikachu.hp += charizard\_damage  self.last\_charizard\_attack = "None"  print(battle.battle\_info())  raise BattleEnded("Charizard has been defeated.")  if self.pikachu.hp <= 0:  print(f"Charizard used {result[1]} and dealt {result[2]} damage.")  self.charizard.hp += pikachu\_damage  self.last\_pikachu\_attack = "None"  print(battle.battle\_info())  raise BattleEnded("Pikachu has been defeated.")  Input arguments:  No input arguments | HLR\_007 | ILR\_017  ILR\_018 |
| LLR\_003 | The Charizard class shall create a method for Charizard to attack Pikachu and should have the following characteristics  Class prototype: class Charizard:  def \_\_init\_\_(self):  self.hp = 30  self.attacks = {  1: 2,  2: 1,  3: 3,  4: 2,  5: 1,  6: 3,  7: 2,  8: 1,  9: 3,  10: 2,  11: 1,  12: 3,  13: 2,  14: 1,  15: 3,  16: 2,  17: 1,  18: 3,  19: 2,  20: 1,  21: 3,  22: 2,  23: 1,  24: 3  }  def attack(self, round\_num):  if round\_num not in self.attacks:  raise InvalidMoveException("Invalid round number.")  damage = self.attacks[round\_num]  return damage  Input arguments:  round\_num – The round number at the start of the round | HLR\_005 | ILR\_012 |
| LLR\_004 | The battle\_info function shall show battle information in the correct order and should have the following characteristics  Function prototype: battle\_info(self):  return f"Round: {self.round\_num}\n"\  f"Pikachu HP: {self.pikachu.hp}\n"\  f"Charizard HP: {self.charizard.hp}\n"\  f"Last Pikachu Attack: {self.last\_pikachu\_attack}\n"\  f"Last Charizard Attack: {self.last\_charizard\_attack}"  Input Arguments:  No input arguments | HLR\_001  HLR\_002  HLR\_004  HLR\_005 | ILR\_001  ILR\_002  ILR\_003  ILR\_004  ILR\_005  ILR\_006  ILR\_007  ILR\_008  ILR\_009 |
| LLR\_005 | The code that calculates Pikachu health points by subtracting the Charizard attack damage from the exsisting Pikachu health points should have the following characteristics:  self.charizard.hp -= pikachu\_damage  Input Arguments:  No input arguments | HLR\_004 | ILR\_010 |
| LLR\_006 | The code that calculates Charizard health points by subtracting the Pokemon attack damage from the exsisting Charizard health points should have the following characteristics:  self.pikachu.hp -= charizard\_damage  Input Arguments:  No input Arguments | HLR\_004 | ILR\_011 |
| LLR\_007 | The code that makes the Pikachu health points equal to 0 if the Pikachu health points become a value of less than 0 shall have the following characteristics:  if self.pikachu.hp < 0:  self.pikachu.hp = 0  Input Arguments::  No input arguments | HLR\_004 | ILR\_013 |
| LLR\_008 | The code that makes the Charizard health points equal to 0 if the Charizard health points become a value of less than 0 shall have the following characteristics:  if self.charizard.hp < 0:  self.charizard.hp = 0  Input Arguments:  No input arguments | HLR\_004 | ILR\_014 |
| LLR\_009 | The class that creates a InvalidMove exception shall have the following characteristics  Class prototype: class InvalidMove(Exception):  pass  Input Arguments:  Exception – Pauses execution of code. A command that can be called upon to replay the round or finish the simulaton | HLR\_005  HLR\_006 | ILR\_015  ILR\_016 |
| LLR\_010 | The class that creates a BattleEnded exception shall have the following characteristics  Class prototype: class BattleEnded(Exception):  pass  Input Arguments:  Exception – Pauses execuion of code. A command that can be called upon to replay the round or finish the simulaton | HLR\_005  HLR\_007 | ILR\_017  ILR\_018 |
| LLR\_011 | The class that creates the initial statistics for Pikachu should have the following characteristics:  Class prototype: class Pikachu:  def \_\_init\_\_(self):  self.hp = 15  self.attacks = {  "Volt Tackle": 5,  "Electroweb": 4,  "Electro Ball": 3  }  self.last\_attack = None  Input Arguments  No input arguments | HLR\_004  HLR\_005  HLR\_007 | ILR\_10  ILR\_13  ILR\_17 |
| LLR\_012 | The code that selects a Charizard attack based of the round number should have the following characteristics:  charizard\_attack\_number = self.charizard.attack(self.round\_num + 1)  if charizard\_attack\_number == 2:  charizard\_attack\_name = "Air Slash"  if charizard\_attack\_number == 1:  charizard\_attack\_name = "Dragon Breath"  if charizard\_attack\_number == 3:  charizard\_attack\_name = "Ember"  charizard\_damage = self.charizard.attacks[charizard\_attack\_number]  Input Arguments:  No input arguments | HLR\_003  HLR\_004  HLR\_005  HLR\_007 | ILR\_11  ILR\_14  ILR\_18  ILR\_19 |
| LLR\_013 | The code that controls the interactive terminal should have the following characteristics:  while True:  pikachu\_attack\_name = input("Enter attack name for Pikachu: ")  try:  result = battle.advance\_round(pikachu\_attack\_name)  print(f"Pikachu used {pikachu\_attack\_name} and dealt {result[0]} damage.")  print(f"Charizard used {result[1]} and dealt {result[2]} damage.")  print(battle.battle\_info())  except InvalidMove as e:  print(str(e))  except BattleEnded as e:  print(str(e))  break  Input arguments:  No input arguments | HLR\_001  HLR\_002  HLR\_004  HLR\_005 | ILR\_001  ILR\_002  ILR\_003  ILR\_004  ILR\_005  ILR\_006  ILR\_007  ILR\_008  ILR\_009 |

Verification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case no. | Description | Test step/Evidence used | Reference/Evidence  (video id)  (wiring diagram) | Requirements fulfilled |
| 001 | The Pokemon Battle Simulator frontend should have an interactive terminal that supports latin characters and numerical characters | video ID | [Insert video here] | LLR\_004 |
| 002 | The Pokemon Battle Simulator frontend should send user interactive input data to the Pokemon Battle simulator backend | video ID | [Insert video here] | LLR\_012 |
| 003 | The Pokemon Battle Simulator backend should send battle information (Pikachu attack name, Charizard attack name, Pikachu attack damage, Charizard attack damage, round number, Pikachu health points, and Charizard health points) | Video ID | [Insert video here] | LLR\_004  LLR\_005  LLR\_006  LLR\_007  LLR\_008  LLR\_011  LLR\_012  LLR\_013 |
| 004 | The Pokemon Battle Simulator backend shall send an “Invalid move” exception to the Pokemon Battle Simulator frontend when the same attack is used twice in a row or a non exsistant attack name is selected | Video ID | [Insert video here] | LLR\_001  LLR\_009 |
| 005 | The Pokemon Battle Simulator backend shall send a “Battle Ended” exception to the Pokemon Battle Simulator frontend when Pikachu health points or Charizard health points are depleted | Video ID | [Insert video here] | LLR\_002  LLR\_010  LLR\_011  LLR\_012 |

Design justification

*Use of regular classes instead of dataclasses*

Regular classes were used as opposed to dataclasses, because their was a stronger familiarity with regular classes. This meant that the time taken to write the code was quicker, as their was no new coding concept to learn.

*Method used to advance the round*

A potential method of advancing the round would have been to use the “advance\_round(attack: str) -> None” method, however it was decided that a “advance\_round(self, pikachu\_attack\_name)” method would be used as this reduced the complexity of the function. This reduction in complexity allowed for the code to be written quicker, as it is easier to understand the code.

*Interactivity*

One of the objectives of this project was to entertain the user running the program. This was done by allowing the user to choose the attacks of a Pokemon that the user would be controlling. Since the objective was to entertain the user dialogue was added such as “Pikachu used (pikachu\_attack\_name) and dealt (attack\_damage) damage” and “Charizard used (charizard\_attack\_name) and dealt (attack damage) damage”