**Project Title: "Brain up: The Learning Adventure"**

**Project Overview**

**Brain up** is an educational game designed for children aged 10 to 12 (Year 5 to Year 7). It is a turn-based, question-and-answer game that supports two players simultaneously. The game focuses on making learning fun by testing knowledge in subjects like Science, English, and Maori Vocabulary, for instance. Players will answer multiple-choice questions within a time limit, review their mistakes at the end of the game, and compete for a spot on the weekly leaderboard.

The project will be developed using **Python** , with a graphical user interface (GUI) built using either **Tkinter** or **PyQt6** . The development process will follow **Agile methodology** with **Scrum practices** , and **Jira** will be used for task management and tracking progress for two developers.

**Features Breakdown**

**Core Features**

1. **Two-Player Turn-Based Gameplay**
   * Two players take turns answering questions.
   * Each player has their own score tracked independently.
2. **Multiple-Choice Questions**
   * Questions are presented with four possible answers (A, B, C, D).
   * Only one correct answer per question.
3. **Subjects**
   * Three subjects to choose from:
     + **Science** : General science topics suitable for ages 10–12.
     + **English** : Grammar, vocabulary, and comprehension.
     + **Maori Vocabulary** : Basic words and phrases in Te Reo Maori.
     + **More subject could be added in new versions by a json file** or manually input by the application.
4. **Randomized Questions**
   * Questions are randomly selected from a predefined pool for each subject.
   * No repetition of questions within a single game session.
5. **Timer**
   * A countdown timer (15 or 20 seconds) for each question.
   * If the timer runs out, the player loses their turn, and no points are awarded. New question is generated to next player.
6. **Ranking System**
   * Weekly leaderboard based on cumulative scores.
   * Announced every Sunday evening via the app or email (optional).
7. **Graphical User Interface (GUI)**
   * Built using **Tkinter** (lightweight and beginner-friendly) or **PyQt6** (more advanced and visually appealing).
   * Clean, colourful, and child-friendly design.
8. **Review Mistakes**
   * At the end of the game, players can review their incorrect answers and see the correct ones.
   * This feature helps reinforce learning.

**Extra Features**

1. **Skip Question (Twice Per Game)**
   * Players can skip a question twice during the game.
   * Skipping transfers the question to the next player.
   * Skipping does not award any points but allows players to avoid difficult questions and transfer to next player strategically.
2. **Ask for Tips (Twice Per Game)**
   * Players can ask for a hint/tip twice during the game.
   * Hints reduce the number of answer choices (e.g., from 4 to 2).
   * Using a tip reduces the points earned for that question by 50%.
3. **Bonus Points for the Last Question**
   * The final question of the game is worth double the points of regular questions. If the player chooses the wrong answer, lose 10 points. Skip and Tips are also applied in this final question, if remaining.
   * This adds excitement and encourages players to stay engaged until the end.

**Development Plan Using Agile and Scrum**

**Sprint Structure**

Each sprint will last **1 week** , with the following activities:

1. **Sprint Planning** : Define tasks and goals for the sprint.
2. **Daily Standups** : Quick 15-minute meetings to discuss progress and blockers.
3. **Sprint Review** : Demonstrate completed features and gather feedback.
4. **Sprint Retrospective** : Reflect on what went well and what can be improved.

**Backlog Items**

**Core Features**

1. Two-player turn-based gameplay.
2. Multiple-choice questions with random selection.
3. Timer for each question (15 seconds).
4. Basic scoring system.
5. GUI using Tkinter or PyQt6.

**Advanced Features**

1. Skip Question (twice per game).
2. Ask for Tips (twice per game).
3. Bonus points for the last question.
4. Weekly leaderboard (optional if time permits).

**Nice-to-Have Features**

1. Subject selection (Science, English, Maori Vocabulary).
2. Review mistakes at the end of the game.

**Scoring Mechanism to Improve Engagement**

To make the game more engaging and rewarding, we propose the following scoring system:

1. **Base Points**
   * Correct Answer: **10 points** .
   * Incorrect Answer: **0 points** .
2. **Hints Penalty**
   * Using a hint reduces the points earned for that question by **50%** .
   * Example: A correct answer with a hint earns **5 points** instead of 10.
3. **Skipping**
   * Skipping a question does not award points but transfers a difficult question to the next player.
   * The next player can still earn points for answering correctly.
4. **Last Question Bonus**
   * The final question of the game is worth **double points** :
     + Correct Answer: **20 points** (or **10 points** if a hint was used).
     + Wrong Answer: **- 10 points.**
     + Correct Answer with hint: **10 points** (if available yet).
     + Skip: **0 points** (if available yet).
   * This creates excitement and keeps players engaged until the end. Besides, an extra element is subtly introduced, requiring players to perform calculations to choose the best strategy.
5. **Weekly Leaderboard**
   * Scores are reset weekly, and the top 10 players are announced every Sunday.
   * Players earn a badge for being in the top 3 (e.g., Gold, Silver, Bronze).

**User Stories**

1. As a player, I want to register so that I can track my progress.
2. As a player, I want to answer multiple-choice questions within a time limit so that the game is challenging and engaging.
3. As a player, I want to skip a question twice per game so that I can avoid difficult questions strategically.
4. As a player, I want to ask for hints twice per game so that I can get help when stuck.
5. As a player, I want to earn bonus points for the last question so that the game stays exciting until the end.
6. As a player, I want to see my score after each round so that I know how well I am performing.

**Timeline**

Given the new constraint of **5 weeks** , we need to condense the project into **5 sprints** , each lasting **1 week** . This requires prioritizing core features, simplifying advanced features, and focusing on delivering a minimum viable product (MVP) that can be expanded later.

**Sprint Structure**

Each sprint will last **1 week** , with the following activities:

1. **Sprint Planning** : Define tasks and goals for the sprint.
2. **Daily Standups** : Quick 15-minute meetings to discuss progress and blockers.
3. **Sprint Review** : Demonstrate completed features and gather feedback.
4. **Sprint Retrospective** : Reflect on what went well and what can be improved.

**Revised Backlog Items**

**Revised Sprints**

**Week 1: Sprint 1 - Foundation**

* Set up the project repository and Jira board.
* Create basic GUI layout using **Tkinter** (lightweight and quick to implement).
* Implement player registration/login system (basic functionality).
* Design database schema for storing questions, answers, and scores.
* Populate the database with a small set of questions for one subject (e.g., Science).

**Deliverables** :

* Basic GUI framework.
* Functional login/registration.
* Database with sample questions.

**Week 2: Sprint 2 - Core Gameplay**

* Add functionality for two-player turn-based gameplay.
* Implement multiple-choice question logic.
* Integrate a timer for each question (15 seconds).
* Randomize question selection from the predefined pool.
* Develop a basic scoring system (10 points for correct answers, 0 for incorrect).

**Deliverables** :

* Turn-based gameplay.
* Multiple-choice questions with randomization.
* Timer functionality.
* Basic scoring system.

**Week 3: Sprint 3 - Advanced Features**

* Implement "Skip Question" functionality (twice per game).
* Implement "Ask for Tip" functionality (twice per game):
  + Hints reduce answer choices from 4 to 2.
  + Using a hint reduces points earned by 50%.
* Add bonus points for the last question (double points).

**Deliverables** :

* Skip Question feature.
* Ask for Tip feature.
* Bonus points for the last question.

**Week 4: Sprint 4 - Ranking System**

* Develop a scoring mechanism (refined based on earlier feedback).
* Create a weekly leaderboard (if time permits).
* Add functionality to announce winners every Sunday (optional).

**Deliverables** :

* Refined scoring system.
* Weekly leaderboard (if feasible).

**Week 5: Sprint 5 - Final Testing and Deployment**

* Test the app thoroughly for bugs and usability issues.
* Conduct user acceptance testing (UAT) with children in the target age group.
* Deploy the app to a desktop platform.
* Document the code and create a user manual.

**Deliverables** :

* Fully functional MVP.
* Deployed application.
* User documentation.

**Prioritization of Features**

To meet the 5-week deadline, we prioritize the following:

**Must-Have Features**

1. Two-player turn-based gameplay.
2. Multiple-choice questions with random selection.
3. Timer for each question.
4. Basic scoring system.
5. GUI using Tkinter.

**Should-Have Features**

1. Skip Question (twice per game).
2. Ask for Tips (twice per game).
3. Bonus points for the last question.

**Could-Have Features**

1. Weekly leaderboard (optional if time permits).

**Won't-Have Features**

1. Subject selection (focus on one subject initially).
2. Review mistakes at the end of the game (can be added later).

**Scoring Mechanism**

The scoring system remains the same as previously defined but is simplified for the MVP:

1. **Base Points** : Correct Answer = **10 points** .
2. **Hints Penalty** : Using a hint reduces points by **50%** (5 points instead of 10).
3. **Last Question Bonus** : Double points for the final question (**20 points** or **10 points** with a hint).