{EPITECH}

APP Development - 2D Game

PrepPal

Join an exciting adventure with your tech-inspired pet, combining the nostalgia of classic virtual pets with a modern twist. Level up your coding skills and ace your interview prep in a fun, interactive experience crafted just for informatics students. Learn, play, and grow alongside your pixelated companion as you conquer coding challenges and build your expertise!

Game Overview

- **Genre:**
  - Simulation / Educational
    A blend of simulation and educational genres where players care for their virtual pets while enhancing their coding and interview skills.

- **Target Audience:**
  - Primarily informatics or computer science students, aged 16-25, who are preparing for job interviews. Gender-neutral, aimed at individuals interested in technology, coding, and gamified learning. Likely to appeal to students who enjoy interactive learning and educational games.

- **Platform(s):**
  - PC and Mobile. These platforms are accessible to students and suitable for educational and casual gaming experiences.

- **Game Inspiration:**
  - Inspired by classic pet simulation games like *Tamagotchi*, combined with elements from educational and coding games like *CodeCombat* and *Duolingo*. The game integrates interview preparation and coding challenges similar to *LeetCode* or *HackerRank*, while maintaining a casual and engaging experience.

Gameplay and Mechanics

★ Core Gameplay Loop

- **Feed and Care for the Pet:** Keep the pet's health up by feeding it to prevent it from getting too hungry.
- **Answer Interview Questions:** Respond to technical or behavioral questions to earn in-game currency (money).
- **Complete Coding Challenges:** Solve coding problems alongside your pet to gain experience points (XP).
- **Use Earned Currency:** Buy food or other items to maintain and care for the pet.
- **Level Up:** As you gain XP from coding challenges, the pet levels up until it reaches Level 10, marking full interview readiness.

★ Objectives

- Keep your pet healthy by regularly feeding it.
- Earn money by answering interview questions correctly.
- Gain XP by completing coding challenges with the pet.
- Level up the pet to Level 10 to signify interview readiness.

★ Game Progression

- The pet starts at Level 1. Players must care for the pet while answering progressively challenging interview questions and completing coding exercises.
- Progression is tied to the pet’s level, with each level requiring more XP to reach.
- Interview questions and coding challenges become more difficult as the player levels up.
- Reaching Level 10 indicates that the player is fully prepared for interviews.

- ★ **Core Mechanics**
  - ○ **Movement** - Minimal movement since the gameplay is menu-driven. Players navigate through different menus to select actions: feeding, answering questions, or starting coding challenges.
- ★ **Interaction**
  - ○ **Feeding:** Use in-game currency to buy food and feed the pet.
  - ○ **Answering Questions**: Choose from multiple-choice options for interview questions.
  - ○ **Coding Challenges:** Input answers to coding exercises in a simple coding interface.
- ★ **Controls**
  - ○ **Keyboard/Mouse**: Mouse clicks for menu navigation; keyboard inputs for typing answers to coding challenges.
- ★ **Player Abilities**
  - ○ **Feed Pet**: Keep your pet healthy by selecting the appropriate food item from your inventory.
  - ○ **Answer Interview Questions**: Choose the correct response from multiple choices to earn money.
  - ○ **Solve Coding Challenges**: Use a coding interface to solve tasks in Java or Python, gaining XP for the pet.
  - ○ **Customize Pet**: Use in-game options to change the appearance of the pet, if customization is included.

Game World Design

★ **Level Design**

- ○ The game is designed around a central hub/main screen that acts as a command center. This screen allows the player to interact with various game mechanics like feeding the pet, answering interview questions, and completing coding challenges. The design is straightforward and minimizes complex navigation, keeping the focus on the educational content.

★ **Environment**

- ○ **Style:** The game is minimalist with a tech-inspired, clean interface that's both visually appealing and functional. This style aligns with the target audience—tech students—by minimizing distractions and enhancing focus on coding and interview preparation.
- ○ **Interactive Elements:** The game interface features clickable buttons for different actions. Progress indicators (health and XP bars) are displayed prominently, updating in real-time to show the player's progress. There are no complex environmental interactions—focus is on the interface itself.

★ **Dynamic Elements**

- ○ The game features a static, timeless world. There is no progression tied to real-world time—the health bar stays consistent unless the game is closed, in which case it resets. The pet is represented by a single, unchanging gif with no animations or complex reactions, emphasizing simplicity. The focus remains on the gameplay and coding challenges, with visual updates limited to interactions with buttons and UI elements.

Characters

★ Player Characters

These pets are designed to not only represent different technologies but also to serve as motivational avatars, growing alongside the player's journey to interview success.

- **Gradly** (Gradle Elephant): A cute elephant mascot that excels in staying healthy. Gradly gains more health when fed, making it a reliable companion that can sustain longer sessions.
- **Docky** (Docker Whale): This pixelated whale is a coding expert, earning more XP from coding challenges. Docky helps players level up faster with its natural coding prowess.
- **Nux** (Linux Penguin): A familiar and lovable penguin that’s all about endurance. Nux's slower health depletion makes it easier to keep healthy, perfect for longer, more relaxed play sessions.

Art and Visual Design

★ Concept Art

The concept revolves around blending technology themes with playful, pixel art visuals. Each character is a simplified, colorful 8-bit version of its tech-inspired counterpart, aiming for a lighthearted and engaging look. This playful style is carried throughout the game's user interface, which is simple and intuitive to keep players focused on the content. Concept sketches include:

- 8-bit character designs for Gradly, Docky, and Nux.
- A clear, clean hub screen showing the pet's stats and progress.
- Large, easy-to-recognize buttons with pixel-style animations.

★ UI/UX Design

The UI is tailored for clarity and simplicity, using pixel art aesthetics to enhance the playful theme:

- ○ **HUD Elements:** The heads-up display features bright, bold bars and meters that showcase the pet’s health and XP. Every detail is designed to be visually clear, with contrasting colors that are easy to read and visually appealing.
- ○ **Menus:** Navigation is effortless, featuring large, clickable buttons with pixel animations that add a satisfying sense of interaction. Accessing different sections—like feeding the pet, coding games, or checking interview questions—is seamless, keeping the focus on the educational and entertaining elements

★ Animation

The game’s animations are minimal to maintain the nostalgic 8-bit style:

- ○ **Buttons:** Buttons animate with subtle, satisfying visual feedback when clicked, bringing a sense of responsiveness without overwhelming the screen.
- ○ **Pet Display**: Pets are represented with static gif images, preserving the simplicity of the pixel style. There are no animations for pet reactions, allowing players to focus on gameplay instead of distractions.

Music

- ★ **Music**
  - ○ A single, cheerful, and cute music track plays throughout the game, adding a lighthearted atmosphere that keeps the player engaged while focusing on coding challenges.
  - ○ The volume is fully customizable, allowing players to adjust it to their preference or mute it entirely if they prefer silence while playing.
  - ○ There are no dynamic music transitions, ensuring a consistent and relaxing audio experience from start to finish.

Technical Specifications

- ★ **Engine:**
  - ○ **JavaFX:** The game is developed using JavaFX, a powerful framework for building rich client applications. It is used for the user interface (UI) and to handle the graphics, allowing smooth interaction with the pet and easy navigation through the menus. JavaFX provides a responsive environment that supports both 2D graphics and UI elements.
- ★ **Build & Dependency Management**
  - ○ **Gradle:** The project uses Gradle as the build automation system. Gradle streamlines dependency management, building the game’s structure, and handling various tasks like packaging and testing. It makes the development process more efficient by automating many of the repetitive tasks.

★ Data Handling

- **JSON (via GZone):** Game data such as settings, scores, and progress are stored in JSON format. GZone is used to read and parse these JSON files, allowing the game to load and store information dynamically. This simplifies the management of data and ensures that the game state persists across sessions.

★ Testing & Quality Assurance

- **GUnit:** GUnit is used for unit testing the game logic. It ensures that all components of the game, such as pet health and experience calculations, work as expected. Automated tests help maintain code quality and ensure that new features do not break existing functionality.
- **JaCoCo:** JaCoCo is employed for code coverage analysis, ensuring that the codebase is thoroughly tested. It tracks which parts of the code are being tested and provides metrics for improving test coverage.
- **Mockito:** Mockito is used for mocking objects and behaviors in tests. It allows the developers to simulate different scenarios in the game (such as feeding the pet or answering a coding question) without having to rely on external dependencies or complex setups.

Development Plan

★ Team Roles

- **Yana**: Responsible for building the game's core structure, implementing gameplay mechanics, and conducting thorough testing to ensure functionality. Yana also managed the technical setup, including integrating game features and ensuring the game runs smoothly.
- **Melody**: Focused on the visual and aesthetic aspects of the game designing the UI, and managing the overall visual effects, ensuring that the styling matched the game's playful, tech-inspired theme.

★ Project Milestones

- **Prototype**: Early development phase, focusing on basic functionality—creating the pet selection, setting up health and XP bars, and implementing the core gameplay loop.
- **Alpha**: Introduced the initial user interface, connected core mechanics like feeding and coding challenges, and tested basic features.
- **Beta**: Expanded the game with refined UI, added the final visual elements, polished animations, and made final tweaks to gameplay mechanics. Completed integration of sound and visual assets.
- **Launch**: Final version with all features implemented, bug fixes completed, and polished user experience. Includes the ability to customize the game's sound settings.

★ Schedule and Timeline

- **Phase 1** - Prototype (2 weeks): Build the basic structure, including the pet selection system and core mechanics.
- **Phase 2** - Alpha (3 weeks): Develop core features, integrate initial UI, and connect pet mechanics to the menu-driven gameplay.

    - **Phase 3** - Beta (3 weeks): Focus on visuals and aesthetic polish, finalize sound, and conduct user feedback sessions to improve gameplay.
    - **Phase 4** - Launch (2 weeks): Final round of testing, bug fixing, and polishing the user experience.

- **Testing**

    - **Playtesting Processes:** Each milestone included playtesting to identify gameplay issues, refine mechanics, and adjust the difficulty curve. Testing focused on user experience, ensuring intuitive interactions and balanced gameplay.
    - **QA Plans for Bug Fixes and Polish:** Used JUnit for automated tests to verify gameplay functions like feeding mechanics, XP gains, and pet stats. QA sessions included manual testing to find edge cases, graphical inconsistencies, and any gameplay bugs. A final round of polish was dedicated to refining the UI and ensuring the game operated smoothly across different devices and screen resolutions.

Risk Assessment

- **Potential Challenges**
  - Maintaining player engagement through educational content is the primary challenge—coding and interview scenarios must be accessible and rewarding. Balancing difficulty and pacing is key, ensuring the game remains appealing to a wide range of player skills.
- **Backup Plan**
  - Player feedback during early testing phases will guide adjustments. If a particular challenge or feature proves too difficult, it will be refined based on player input, making the game more user-friendly without compromising educational value.

Marketing and Distribution

- **Target Market Strategy**
  - The game is designed for computer science students, coding enthusiasts, and those preparing for technical interviews. Marketing will focus on student forums, educational coding websites, and social media groups, emphasizing the fun and friendly approach to learning.
- **Platforms for Launch**
  - The game will initially launch on PC through platforms like Itch.io, targeting accessible, downloadable content for students.

Appendices

References:

- **Inspirations:**
  - The concept is inspired by the classic Tamagotchi, blending nostalgic pet care mechanics with modern tech themes tailored for coding students.
  - Pixel art style inspired by retro 8-bit games, which are known for their simplicity and charm.
- **Research Materials:**
  - Documentation and guides for **Java 17**, **JavaFX**, and **Gradle** were used extensively to understand and implement the core gameplay features.
  - Coding challenges and interview preparation guides inspired the educational content integrated into the game.

Glossary:

- **Health Points (HP):** A measure of the pet’s well-being. Decreases if the pet isn't fed or if the game is closed. Starts at zero if the game is relaunched.
- **Experience Points (XP):** Points earned by completing coding challenges. Accumulating XP allows the pet to level up.
- **Coding Challenges:** A game mode that tests the player’s coding knowledge. Successfully completing a challenge rewards XP.
- **Feed:** An option in the game that allows players to increase the pet’s HP. Different pets benefit from feeding in unique ways.
- **Hub/Main Screen:** The central interface where players can monitor their pet’s stats and access game options.
- **8-bit Art Style:** A pixelated visual style reminiscent of early video games, characterized by bright colors and simple animations.
- **UI (User Interface):** The elements that make up the game’s layout, including buttons, health bars, and menus.
- **JSON Files:** A lightweight data format used to store and retrieve information in the game, such as pet data and player progress.