Chameleo Car: Proof of Concept and Software Demonstration

Presentation Script

I. Slide 1: Title Slide
   a. Our toy concept is the "Chameleo Car"
   b. This is a smart-phone controlled toy race car capable of changing colour and customising its appearance with the goal of encouraging diversity in racing

II. Slide 2: Concept Introduction
   a. Car racing participants do not represent the diversity of our population; for instance, the Hamilton Commission found that only 1% of the people come from Black background (Mitchell-Malm, 2021).
   b. Besides racial and ethnic minorities, multiple studies have revealed barriers to entry for women too, stating that it has been traditionally confined to male drivers (Reid & Lightfoot, 2019; Howe, 2022).
   c. Because of this, our mission is to create a toy that encourages diversity and inclusion in car racing, with a vision of increasing diverse representation in Formula 1.
   d. By creating a toy that 1) appeals to audiences of broader racial, ethnic, and gender representations, and 2) shows that drivers can look like anyone, we hope to bring a wider societal participation into car racing.
   e. The output of this development phase is software that enables functionality of the car, and a simulation environment to demonstrate and test these functions.

III. Slide 3: Requirements: 'Child' Persona
   a. Before we began any design or development activities, we first generated requirements for the toy's software.
   b. To do so, we used a persona-based framework as per best practices in Software Engineering-related Project Management (Nielsen & Nielsen, 2019).
   c. After considering the many potential use cases, scenarios, and relationships with this toy, we settled on three personas:
   d. 'Child,' the primary user of the toy, likely between 6 and 12 years of age
   e. 'Guardian,' the parent, caretaker, or responsible party monitoring the child's interaction with the toy
   f. 'Producer,' the manufacturer of the final physical toy and developer/distributer of the accompanying smart phone app
   g. Unfortunately, we did not have access to child users, so we researched features of comparable toys on the market
   h. Due to not having access to these users, there is inherent risk that these requirements do not reflect user wants and needs
   i. Requirements are prioritized in order of 1) functional dependencies as aligned with our mission and 2) predicted user enjoyment

j. Requirements 1 and 2 focus on the child user being able to control the car. Since these requirements form the basis for the core race car concept, these requirements are high priority.
k. Requirement 3 – honking the horn – is not critical to this toy concept. Rather, it's an extra feature that a child user may enjoy.
l. Requirements 4 and 5, changing the appearance of the car and the driver, directly support our mission to create a toy that is inclusive and representative. Accordingly, these are high priority requirements.
m. Requirement 6, displaying the battery status, is an important requirement for practical use and to inform users when the car needs to charge. This is a medium priority.
n. Requirement 7, sending a low battery alert, is another "nice to have" feature – but not necessary for use or enjoyment of the toy.

IV. Slide 4: Requirements: 'Guardian' and 'Producer' personas
a. As with the 'child' persona, we did not have access to parent or producer users, so we researched features of comparable toys on the market
b. Requirement 1 for the Guardian persona is for password-protected parental controls to limit features of the toy. This is important, but not critical to the operation of the toy – so this is a medium priority.
c. Requirement 2 is a setting in the parental control menu to set the maximum speed of the race car. We believe this is a useful feature, but not vital to toy operation, so this is also a medium priority.
d. Requirement 3, allowing the parent to set the driver's appearance, is a low priority since this feature is enabled through the child persona.
e. Requirement 1 for the Producer persona is to gather toy usage statistics. This data could be used to improve future toys but is not critical to toy operation so it's a low priority.
f. Requirement 2 is for diagnostic reports. This is of medium importance since it doesn't impact toy operation, but it may help identify problems with the toy.
g. Future requirements will focus on enhanced security, such as multi-factor authentication and integrating analytics

V. Slide 5: Design and Plan
a. We applied object-oriented design principles to this application in terms of classes, attributes, methods, and relationships (XX CITE OOIS REFERENCE)
b. After researching similar applications, we decided to leverage common game functions from the widely-adopted pygame library (Pygame, 2022). Alexander Svilarov's "Race It!" app inspired our vehicle control, physics, and background implementation (2019).

VI. Slide 6: Sprint Progress and Project Status
a. This slide shows a sprint-level view of the entire release of the race car software and demo environment.
b. Using Agile principles, we allocated requirements across three fortnightly sprints for this initial release.
c. This achievement was possible via adequate team and project management, understanding and managing risks and dependencies from the start, and ensuring code reviews and the correct sequential development and testing of the components in the application.

VII. Slide 7: Budgets and Summary
   a. As a team of seasoned IT professionals, we used a combination of methods to estimate our labour allocation. These methods included top-down, bottom-up, and expert judgment estimation techniques (Nasir, 2006). We considered our total time and resources available, the … (XX COMPLETE AND CITE XX)
   b. Our team did not limit a team member to a specific role, as each person contributed to multiple functions. Thus, we made an effort to track our time according to the tasks performed.

VIII. Slide 8: Development of Demo/Simulation
   a. Accepts input and produce desired actions in the simulated car: movement, parental controls, and customisation of car colour and driver appearance
   b. Uses pygame module for implementation, Gherkin for requirements testing, pytest for testing, bandit and safety for security checks
   c. Developed with Agile Scrum, using GitHub as a VCS and requirement tracker, and GitHub Actions for CI/CD
   d. Next step is to port these functions to hardware: smart phone app and physical race car toy

IX. Slide 9: Challenges and Solutions
   a. Developing software remotely, using different operating systems and environments, required additional testing and setup configuration
   b. Establishing a modular Python package-like structure and a common environment.yml file enabled to standardise the setup and minimised the room for any configuration errors
   c. Initially, linting checks were run manually, but automating them via GitHub Actions enabled early visibility and correction of errors to increase code quality.