**Contents**

1. Summary...............................................................2

2. User Discussion.......................................................2

2.1 Splash Page..........................................................2

2.2 Play.................................................................3

2.3 How To Play Page.....................................................4

2.4 Resume Page..........................................................5

3. Developer Discussion..................................................6

3.1 Splash Page..........................................................6

3.2 Play.................................................................6

3.3 How To Play Page.....................................................6

3.4 Quit.................................................................6

4. Feasibility...........................................................7

4.1 Constraints..........................................................7

4.2 Cost.................................................................7

4.2.1 Analysis...........................................................7

4.2.2 Design.............................................................7

4.2.3 Coding.............................................................8

4.2.4 Testing............................................................8

5. System................................................................9

5.1 Kernel...............................................................9

5.2 Standard.............................................................9

5.3 Super................................................................9

6. Schedule..............................................................9

**1. Summary**

C.R.A. (working title) is a side-scrolling game about a robot in formalwear escaping from a toy store. He is constantly moving in relation to his surroundings. He progresses through a world filled with obstacles that he must jump over or otherwise manipulate himself to the environment. It is to be developed for the Android “JellyBean” 4.0 environment using the Java programming language. The game launch will be free-to-play, and available on the Android platform via digital sources such as Google Play.

C.R.A

**2. User Discussion**

Cybernetic Running Android

**PLAY**

2.1 Splash Page

This screen is used to give the user  
three options including play, how to  
play, and quit.

**HOW TO PLAY**

**QUIT**

2.2 Play

This is just a sample gameplay screen.

Switch Color:

2.3 How To Play Page

This screen describes to the user how to play the game.

Your Robot is in a hurry! He will be running at a constant speed because his maker designed him to do so. Your job is to help him complete the level because he is incapable of performing actions, besides running, by himself. He can color-shift, meaning he can change between two colors at any given time. The background and the objects will be two different colors. The Robot cannot go through an object that is not the same color as him. By pressing the button on the bottom of the screen, you can invert the colors of the objects. Good Luck!

Back

**How To Play**

2.4 Resume Page

This screen is displayed when a user opts out of the app altogether by going back to the home screen on the phone. There will not be a pause option in the app. This will be the only way to essentially pause the app. ( not to scale)

RESUME

**3. Developer Discussion**

3.1 Splash Page

The splash screen will contain three options to proceed from the game. There will be a "Play" option which will start the game by calling the play class. The next option will be the "How To Play" option, which will take the user to a tutorial of how to play the game and some simple settings. The last option for user will be the "Quit", which will end the game.

3.2 Play

The play option will start the game immediately and starts keeping track the player's score. The score will be based on scale of difficulty. The playing screen will be a rolling screen in which the character will have to avoid obstacles to progress in the game. The rolling screen would be a long graphic design the entire level. There will be several obstacles for the character to avoid by jumping over them. Some of methods needed for the play class will be a collision method to determine if the character collides with one of object and end the game. If necessary, there will be a score method to keep track of the player's score throughout each level.

3.3 How To Play Page

The how to play page will give the user a small tutorial of how to play the game and perform each action correctly to avoid obstacles. The tutorial will show the user the controls which will be a button on screen to command the character to jump. In addition to the tutorial there will be settings and controls for the user to adjust the volume.

3.4 Quit

The quit page will shortly appear with salutation before exiting the game.

**4. Feasibility**

4.1 Constraints

This application is designed for Android Phones. It will support all Android operating system versions from 4.0 to the current version, namely Ice Cream Sandwich, Jelly Bean, and Kit Kat. It will also only support Android’s generalized “normal” screen size. The “normal” screen size ranges from approximately three to five inches. The minimum processor and memory requirements have not yet been determined. The application will only support the English language.

4.2 Cost

The estimated total time required for the project is 1000 hours. Each of the four team members will need to contribute approximately 17 hours per week to complete the project on schedule. After making allowance for unanticipated problems, the total cost of the project is $210,000. The following is a breakdown of each phase of the project and the estimated time required for the phase.

4.2.1 Analysis

The analysis phase of the project will require 250 hours to complete. It is estimated to require twenty-five percent of the total time necessary for completing the project. The analysis phase was given twenty-five percent of the total time allowed for the project because it is an essential phase for developing the storyline and specifying the basic functionality and features of the game. All of these elements are crucial to developing a successful game.

4.2.2 Design

The design phase of the project will require 450 hours to complete. Forty-five percent of the total project time was allowed for the design phase. The design phase requires the largest portion of the total project time because effective design is essential to avoiding bugs and major design flaws that would make the coding and testing phases take longer than expected and increase the total project time.

4.2.3 Coding

The coding phase will require 150 hours to complete. This is calculated at fifteen percent of the total time for the project. If the analysis and design phases are completed properly, the coding phase should not require a significant amount of time.

4.2.4 Testing

The testing phase will require 50 hours to complete. This is calculated as five percent of the total project time. Some of this time will be interspersed into the coding phase as additional features are added. The rest of the time will be used during an intensive testing phase near the completion of the project to ensure the project meets all specifications.

|  |  |  |
| --- | --- | --- |
| Time Requirements Summary | | |
|  | Hours | Percentage |
| Analysis | 250 | 25% |
| Design | 450 | 45% |
| Coding | 150 | 15% |
| Testing | 50 | 5% |
|  | 1,000 |  |

**5. System**

5.1 Kernel

The 10-week kernel should include the necessary baseline to function, essentially, a single testing level, complete with obstacles, jumping functionality, a constantly moving screen, unit collision for end-of-game purposes, and some basic sprites.

5.2 Standard

The 14-week standard should include five total levels, conveying progression; furthermore, it will include a color-swapping functionality as another gameplay-driven mechanic; therefore, at this point, the art for the game will include a variety of colors. At this point, there will be pause functionality, as well, so the game can be stored in memory in the background while the user does more important things.

5.3 Super

The projected 18-week super will include a final, randomly generated, infinite level. Based upon how far the player goes, there will also be a local high score functionality. Provided we can acquire a server to offload data to, we may be able to include into the project a global high score function, as well. Other functionality will depend on time constraints, but additional art would be in order for this infinite level.

**6. Schedule**

**Week 1:** Initial meeting of group. Decisions on what kind of project was to be worked on.

**Week 2:** Initial Analysis. Platform decided upon; group settled on game development for Android. Game features brainstormed.

**Week 3:** Continued Analysis. Game features ironed out further, requirements document started.

**Week 4:** Continued Analysis. Game features finalized, requirement document near completion.

**Week 5:** Final Analysis. Finish requirements document, begin work on design document.

**Week 6:** Initial Design. Flesh out art/level/character design. Begin work on various UML diagrams.

**Week 7:** Continued Design. Continue art design. Address any valid concerns about user perception, i.e. colorblindness, etc.

**Week 8:** Continued Design. Finalize art design. Begin simple level design.

**Week 9:** Continued Design. Finalize UML diagrams. Continue level design. Address any valid concerns about usability on the user’s end, i.e. playability, realistically.

**Week 10:** Final Design. Finish design document. Have all class and object relationships established. Finish level design.

**Week 11:** Initial Coding. Begin coding of program shell; ensuring game is compatible on the targeted Android platforms.

**Week 12:** Continued Coding. Code in unit collision, movement, align art properly in-game, start coding the levels.

**Week 13:** Final Coding; Initial Testing. Finish coding the levels. Begin error-checking software for any bugs that may crop up.

**Week 14:** Final Testing. Intentionally try to break the software in testing, checking if malicious intent could have an adverse effect on the hardware or software.

**Week 15:** Finalized Product. The game will be ready to be authorized on various application dealers, e.g. Google’s Play store.