|  |
| --- |
| **Koal-A-Duty**  Final Project Proposal  ITC 383  10-16-2013  **Professor:** Anthony Morelli  **Students:**  Zachary Alexander  Tyler Kramer  Eric Noble  Evan J. Travis |

**Team Members and Primary Roles**

The primary roles given to each team member are meant to be fluid and are not nor do they contain an exhaustive list of tasks. Any team member can contribute to the different aspects of the project. Ideally, once the mechanics and the rules governing them are created and easily tweak-able, the team will begin to heavily focus on art production, level design, and plot development.

Zachary Alexander, Senior

* Level Design Lead
  + Level order/flow
  + Art and Mechanics Requisition and Application

Tyler Kramer, Junior

* Software Development Lead
  + Core Mechanic Development
  + Modularization and Data Management

Eric Noble, Senior

* Visual and Audio Art Lead
  + Tiles
  + Pixel Art
  + Sound Effects
  + Music

Evan J. Travis, Senior

* Project Manager
  + Documents Management
  + Team Coordinator
* User Interface Design Lead
  + Screen-Flow Diagrams
  + On-Screen Controls

**Project Description**

Koal-A-Duty is a side-scrolling shooter. Players take on the role of a Homicidal Koala, whose only goal (so far) is to survive long enough in order to get to the end of the level. Players will be attacked by Australian enemies such as Dingos, Owls, Lizards, and Poachers, with each world associated with a species archetype boss battle. As a Koala Komando, players will have access to an array of weapons to aid them in their fight, while also enlisting the help of their friends, Fox and Turtle. Koal-A-Duty will be developed with the following ideas in mind:

1. Goal: Entertainment
2. Time-Interval: Real-Time
3. Player Mode: One-Player
4. Genre: Action, Platform, and Shooter
5. Aesthetics: (MDA.pdf)

* **Challenge**, *game as obstacle course*
* **Fantasy**, *game as make-believe*

**What We Expect to Complete**

Your Five:

1. Input
2. Random Elements
3. How To Play
4. End of Game that Correctly Restarts to the Beginning
5. Minutes of Unique Gameplay

Our Five:

1. One world (collection of levels, at least 1) per person
2. Allies as alternative playable characters

* Fox
* Turtle

1. Numerous art assets, combined with a unified visual and audio aesthetic
2. Limited story/plot development
3. Boss Battles, 1 / team member

* Dingo
* Owl
* Lizard
* Poacher

**Concerns**

Time and Scope:

It may be difficult to complete all of our goals given the amount of time that we have to complete them. We will have to be able to dedicate a significant amount of time to ensure that the project is completed and lives up to our expectations while remaining within the scope of our proposed design. By using the available API, it should be possible to break down the project into more manageable parts.

API Comprehension:

While the Corona SDK is very helpful and somewhat easy to use, it not possible to understand it completely with the amount of time that we have spent with it. That being said, it is possible that there will be times when project resources are wasted in order to reinvent the wheel that can be implemented much easier given the existing API. The ensuing frustration could also factor in when considering teamwork and group cohesion.

Compromise and Cooperation:

Given the inherently creative process of designing and implementing a video game, chances are that different minds will insist on or become attached to certain ideas. The team will need to understand their own responsibilities as they relate to the overall project. Additionally, team members will need to be able to communicate ideas freely without insisting that their particular vision be the only one that is brought to fruition. It will be important to maintain a consistent feel and presentation of the game, which will be achieved through a unified art aesthetic working with an accurate and robust mechanics system in order to release the full potential of an engaging play experience within individual levels and the game world as a whole.

Playability across Platforms:

Thanks to the Corona SDK, it is possible to develop a game and then have it ported to multiple systems simultaneously. However, just because a game runs on a piece of hardware, does not mean that it will run well. For instance, in the process of developing previous assignments, it became apparent that object size did not necessarily translate well from a large-screened surface to a small-screened one, resulting in a loss of direct control and perception for the player, thereby increasing frustration and decreasing (re)playability. The team will need to work hard in order to keep these kinds of problems in mind while developing for multiple systems at the same time.

Access to Platforms:

Though many students today have access to smartphones and tablets, these items can be prohibitively expensive. As a result, we may not have the preferred resources when developing a game for a preferred or possible system. Our project could be mostly complete only to find out that playing the game on the target hardware for the first time results in an unexpected gameplay feel. Granted, it is possible that situations like these can turn into happy accidents, improving the game rather than breaking it. Seeing as how everyone would lack the funds to purchase a different or new target hardware system, we are limited in what we can do to curb the snowballing effects of these types of errors. The only thing that we can definitely do is keep in mind how the game will play on the target hardware and not on how it plays with the hardware simulator that comes with Corona SDK.