**Phase Plan Document**

**GAM150S14-C**

**Spring 2014**

**Producer:**

Juli Gregg

**Tech Director**:

Dan Muller

**Game Designer**:

Luke Conlon

**Project Manager**:

Kaden Nugent

Table of Contents

[High Concept: 2](#_Toc378028337)

[Summary: 2](#_Toc378028338)

[Story: 2](#_Toc378028339)

[Game Flow: 3](#_Toc378028340)

[Game Mechanics: 3](#_Toc378028341)

[Game Characters: 3](#_Toc378028352)

[Game Resources: 4](#_Toc378028353)

[Game Environment: 5](#_Toc378028354)

[Game Controls: 5](#_Toc378028355)

[Visual Design: 5](#_Toc378028356)

[Audio Design: 5](#_Toc378028357)

[Behavior Design: 5](#_Toc378028358)

[Physics Design: 6](#_Toc378028359)

[Multiplayer Design: 6](#_Toc378028360)

[Appendices 7](#_Toc378028361)

[Appendix A: 7](#_Toc378028362)

[Appendix B: 8](#_Toc378028364)

[Appendix C: 8](#_Toc378028366)

[Appendix D: 9](#_Toc378028368)

High Concept:

A 2-D cartoon fantasy dungeon crawler side-scrolling hack and slash game.

Introduction:

The mighty princesses of the Fox Clan are sent out to rescue the clan’s prince who has been captured during an attack by their enemy, Dog Clan. Due to the king being mortally wounded from the Dog Clan’s attack the heroines are on a deadline to bring back the sole heir of the Fox Clan. The player needs to successfully help the princesses hack and slash their way through enemy clan members to reach the end of each level. Each section of levels will feature a boss fight. Players will be able to customize the princesses’ stats, weapons, and pets as they gain abilities, money and stat points as the game progresses. Players also have the option to spend money they find along the way in the game to buy new items such as weapons and pets or buffs. In the end the player must wisely power up their character in order to save the prince for if there is no heir left when the king of the Fox Clan dies their clan will disband and be open to a takeover by the Dog Clan.

Engine Proof:

|  |  |
| --- | --- |
| Code: | |
| Core Architecture: | Main mechanics and management systems are implemented |
| Graphics Code: | Provided through Alpha Engine. An additional wrapper around Alpha Engine has also been created. |
| Physics Code: | Implementation is finished and only a values need to be tweaked/ |
| Audio Code: | A wrapper around FMOD functions has been implemented. |
| Input Code: | Implemented through Alpha Engine. |
| GUI Code: | Code for HUD and basic menu functionalities has been implemented. |
| Game Logic Code | A Game State Manager and player attacking has been implemented. No enemy logic. |
| **Tools:** | |
| Art Pipeline: | Engine has the ability to import art through a wrapper around Alpha Engine’s own Open GL wrapper. |
| Content Pipeline: | Tutorial and main menu is finished. |
| Debugging Tools: | Printing to console in Visual Studio. Using a Visual Studio plug-in to detect memory leaks. Debug boxes for collision have been implemented. |
| **Content:** | |
| Art Assets: | Single player art is done including animation. Text style is finished as well as team and game logo. |
| Audio Assets: | None. |
| Written Assets: | Instructional text/pop-up text has been implemented in tutorial level. Story for the game has been plotted out. |
| Levels: | Tutorial Level is complete. Paper designs and layouts for all four boss levels and three regular levels are finished. |
| Game Data: | Sprites, Collectables, and Platforms are finished. The player and sounds still need tweaking. Weapons are in testing and the enemy currently only have placeholder data. |
| **Finishing:** | |
| Testing: | Currently testing that the Engine components and mechanics work. Engine is stable. |
| Polish: | Currently have the ability to adjust game for window resizing. |

Alpha:

|  |  |
| --- | --- |
| Code: | |
| Core Architecture: | Main mechanics management systems are implemented. |
| Graphics Code: | Provided through Alpha Engine. An additional wrapper around Alpha Engine has also been created. |
| Physics Code: | Implementation is finished and values are finalized. |
| Audio Code: | A wrapper around FMOD functions has been implemented as well as a Sound Manager. |
| Input Code: | Implemented using Windows.h API. |
| GUI Code: | Code for HUD and basic menu functionalities has been implemented. |
| Game Logic Code | A Game State Manager, basic enemy, two of four bosses, combat, and stat/upgrades logic will be implemented. |
| **Tools:** | |
| Art Pipeline: | Engine has the ability to import art through a wrapper around Alpha Engine’s own Open GL wrapper. |
| Content Pipeline: | Main menus, tutorial, and levels up to second of four bosses will be done. |
| Debugging Tools: | Printing to console in Visual Studio. Using a Visual Studio plug-in to detect memory leaks. Debug boxes for collision have been implemented. |
| **Content:** | |
| Art Assets: | Single player art is done including animation and is polished. Text style is finished as well as team and game logo. Basic enemies, three level backgrounds, and first two levels of each of the four upgradable weapons will be finished. |
| Audio Assets: | One background track will be done along with sound effects for combat. |
| Written Assets: | Instructional text/pop-up text has been implemented in tutorial level. Story for the game has been plotted out. |
| Levels: | Tutorial Level and levels up to second of four bosses will be done. Paper designs and layouts for all four boss levels and the six remaining regular levels are finished. |
| Game Data: | Sprites, collectables, weapons, sounds, and platforms are finished. The player still needs tweaking. The enemies are in testing and the first two bosses need tweaking |
| **Finishing:** | |
| Testing: | Currently testing and balancing item drops, boss AI systems, and amount of enemies per level. Engine is stable and easy to add and replace elements. |
| Polish: | Currently have the ability to adjust game for window resizing. Controls and UI should be finished. |

Beta:

|  |  |
| --- | --- |
| Code: | |
| Core Architecture: | Main mechanics management systems are implemented. |
| Graphics Code: | Provided through Alpha Engine. An additional wrapper around Alpha Engine has also been created. |
| Physics Code: | Implementation is finished and values are finalized. |
| Audio Code: | A wrapper around FMOD functions has been implemented as well as a Sound Manager. |
| Input Code: | Implemented using Windows.h API. |
| GUI Code: | Code for HUD and basic menu functionalities has been implemented. |
| Game Logic Code | A Game State Manager, basic enemy, all four bosses, combat, and stat/upgrades logic will be implemented. |
| **Tools:** | |
| Art Pipeline: | Engine has the ability to import art through a wrapper around Alpha Engine’s own Open GL wrapper. |
| Content Pipeline: | All levels and bosses will be done. |
| Debugging Tools: | Printing to console in Visual Studio. Using a Visual Studio plug-in to detect memory leaks. Debug boxes for collision have been implemented. |
| **Content:** | |
| Art Assets: | Single player art is done including animation and is polished. Text style is finished as well as team and game logo. Basic enemies, three level backgrounds, and all (4) upgradable levels of each of the four upgradable weapons will be finished. All level backgrounds (14) will be done. |
| Audio Assets: | Three background tracks will be done along with sound effects for combat, voice overs, narration, item swapping and pickups. |
| Written Assets: | Instructional text/pop-up text has been implemented in tutorial level. Scripts for cut scenes and voice overs for character sayings will be finished. Story will be fully represented in the game. |
| Levels: | All levels will be done. |
| Game Data: | Sprites, collectables, weapons, sounds, platforms, player, enemies and bosses will be at the stage to be polished |
| **Finishing:** | |
| Testing: | Will be balancing overall gameplay through play-testing club as well as searching for any missed bugs. Engine is stable and easy to add and replace elements. |
| Polish: | Currently have the ability to adjust game for window resizing. Controls and UI should be finished. Will be creating a priority list of components to be polished in order of importance. |

Final:

|  |  |
| --- | --- |
| Code: | |
| Core Architecture: | Main mechanics management systems are implemented and polished. |
| Graphics Code: | Provided through Alpha Engine. An additional wrapper around Alpha Engine has also been created. |
| Physics Code: | Implementation is finished and values are finalized. |
| Audio Code: | A wrapper around FMOD functions has been implemented as well as a Sound Manager. |
| Input Code: | Implemented using Windows.h API. |
| GUI Code: | Code for HUD and all menu functionalities has been implemented. |
| Game Logic Code | A Game State Manager, basic enemy, all four bosses, combat, and stat/upgrades logic will be implemented and be balanced where needed. |
| **Tools:** | |
| Art Pipeline: | Engine has the ability to import art through a wrapper around Alpha Engine’s own Open GL wrapper. |
| Content Pipeline: | All levels and bosses will be done. An installer will be finished. |
| Debugging Tools: | Printing to console in Visual Studio. Using a Visual Studio plug-in to detect memory leaks. Debug boxes for collision have been implemented. |
| **Content:** | |
| Art Assets: | Single player art is done including animation and is polished. Text style is finished as well as team and game logo. Basic enemies, three level backgrounds, and all (4) upgradable levels of each of the four upgradable weapons will be finished. All level backgrounds (14) will be done. Three additional character designs will be added so player can pick which design to choose in a character select menu. Menu art will be done as well as art required for marketing tools (box art). |
| Audio Assets: | Three background tracks will be done along with sound effects for combat, voice overs, narration, item swapping and pickups. |
| Written Assets: | Instructional text/pop-up text has been implemented in tutorial level. Scripts for cut scenes and voice overs for character sayings will be finished. Story will be fully represented in the game. |
| Levels: | All levels will be done and the ability to save progress will be implemented. |
| Game Data: | Sprites, collectables, weapons, sounds, platforms, player, enemies and bosses will be polished. |
| **Finishing:** | |
| Testing: | Testing for the installer and proper save locations. |
| Polish: | Game will be finished and polished and marketing tools will be complete. |