**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](#_Toc380102864)

[Introduction: 2](#_Toc380102865)

[Engine Proof: 2](#_Toc380102866)

[Code: 3](#_Toc380102867)

[Tools: 3](#_Toc380102868)

[Content: 3](#_Toc380102869)

[Finishing: 3](#_Toc380102870)

[Alpha: 4](#_Toc380102871)

[Code: 4](#_Toc380102872)

[Tools: 4](#_Toc380102873)

[Content: 5](#_Toc380102874)

[Finishing: 5](#_Toc380102875)

[Beta: 5](#_Toc380102876)

[Code: 6](#_Toc380102877)

[Tools: 6](#_Toc380102878)

[Content: 6](#_Toc380102879)

[Finishing: 7](#_Toc380102880)

[Final: 7](#_Toc380102881)

[Code: 7](#_Toc380102882)

[Tools: 7](#_Toc380102883)

[Content: 8](#_Toc380102884)

[Finishing: 8](#_Toc380102885)

High Concept:

A 2-D fast-paced, side-scrolling hack and slash game set in a cartoon fantasy world with character ability progression.

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 and the king of the Fox Clan dies their clan will disband and be open to a takeover by the Dog Clan.

Engine Proof:

By this milestone the player should be able to navigate the Main Menu and play through the Tutorial Level. This Tutorial Level will convey to the player the art style that will be displayed throughout the game as well as the main mechanics. Full character animation including attacking animation will also be displayed. The Tutorial Level instructs the player via pop-up text how to use their weapon, switch weapons, and pickup and use buffs. Platforms will also be featured in this level along with a bouncy pad. Players can also pause the game and adjust the volume. In order to complete the Tutorial Level the player will have to hack their way through a dummy blocking the doorway to the level’s exit. Overall the player should get a feel for how to use their character to interact with the in-game environment.

|  |  |
| --- | --- |
| Code: | |
| Core Architecture: | Main mechanics and management systems for objects and memory allocation are implemented as well as a Game State Manager. |
| 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 | Player attacking has been implemented. No enemy logic. |

|  |  |
| --- | --- |
| Tools: | |
| Art Pipeline: | Engine has a simple and robust way to pass art assets into Alpha Engine to draw it on the screen. |
| Content Pipeline: | Tutorial and main menu is finished. |
| Debugging Tools: | Printing to console. Using break points in Visual Studio. Using a Visual Studio plug-in to detect memory leaks. Debug boxes for collision have been implemented. |

|  |  |
| --- | --- |
| Content: | |
| Art Assets: | Player art is done including animation. Text style is finished as well as team and game logo. Tutorial art is also finished. |
| 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 (food), and Platforms are finished. The player and sounds still need tweaking. Weapons are in testing and the enemy object structure has been created. |

|  |  |
| --- | --- |
| Finishing: | |
| Testing: | Currently testing that the Engine components and mechanics work. Engine is stable and easy to add and replace elements. |
| Polish: | Currently have the ability to adjust game for window resizing. |

Alpha:

By this milestone the player should be able to play a more polished tutorial as well as seven other and two of the four planned boss levels. They will be able to navigate through the game defeating basic enemies as well as a few more powerful enemy types. The player will also be able to upgrade their stats and weapons based on the experience points they gain throughout each level. Each boss will feature different abilities making them unique from other enemies in the game. A background track and some sound effects will also be featured. Additional menus that allow the player to change the volume for background music and sound effects individually will be accessible from the main menu and the pause menu.

|  |  |
| --- | --- |
| Code: | |
| Core Architecture: | Main mechanics and management systems for objects and memory allocation are implemented as well as a Game State Manager. |
| 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 API. |
| GUI Code: | Code for HUD and menu functionalities has been implemented. |
| Game Logic Code | Player attacking has been implemented. Basic enemy, two of four bosses, combat, and stat/upgrades logic will be implemented. |

|  |  |
| --- | --- |
| Tools: | |
| Art Pipeline: | Engine has a simple and robust way to pass art assets into Alpha Engine to draw it on the screen. |
| Content Pipeline: | Main menus, tutorial, and levels up to second of four bosses will be done. |
| Debugging Tools: | Printing to console. Using break points 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 including animation is done and 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 of fourteen remaining regular levels are finished. |
| Game Data: | Sprites, Collectables (food), Weapons, Sounds, and Platforms are finished. The player is being polished. 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 will be finished. |

Beta:

For this milestone the game should be level complete and playable including each of the four boss levels. The cut-scenes explaining the game’s story will also be in featured. The plan is to stay ahead and therefore allow for the game to be content complete and only in need of minor tweaking by this point in development. Therefore the player will be able to play the game from start to finish, witness the story being told, and be able to customize the upgrading of their character in order to beat the game. There will also be more sound effects, three background tracks that rotate during gameplay, and voice overs to help explain the story and express the character’s personality.

|  |  |
| --- | --- |
| Code: | |
| Core Architecture: | Main mechanics and management systems for objects and memory allocation are implemented as well as a Game State Manager. |
| 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 API. |
| GUI Code: | Code for HUD and menu functionalities have been implemented. |
| Game Logic Code | Player attacking, basic enemy, all four bosses, combat, and stat/upgrades logic will be implemented. |

|  |  |
| --- | --- |
| Tools: | |
| Art Pipeline: | Engine has a simple and robust way to pass art assets into Alpha Engine to draw it on the screen. |
| Content Pipeline: | All levels and bosses will be done. |
| Debugging Tools: | Printing to console. Using break points 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 (food), 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:

At this point the game will be complete, polished, and fully playable. The ability to save game progress will also be available to the player. There will be marketing materials such as a case for the game and an installer. Gameplay will feel balanced while at the same time somewhat customizable due to the ability to swap out and upgrade weapons and stats based on the players’ choices. If time allows co-op/multiplayer will also be featured in the game.

|  |  |
| --- | --- |
| Code: | |
| Core Architecture: | Main mechanics and management systems for objects and memory allocation are implemented and polished as well as the Game State Manager. |
| 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 API. |
| GUI Code: | Code for HUD and all menu functionalities have been implemented. |
| Game Logic Code | Player attacking, basic enemy, all four bosses, combat, and stat/upgrades logic will be implemented and be balanced where needed. |

|  |  |
| --- | --- |
| Tools: | |
| Art Pipeline: | Engine has a simple and robust way to pass art assets into Alpha Engine to draw it on the screen. |
| Content Pipeline: | All levels and bosses will be done. An installer will be created. |
| Debugging Tools: | Printing to console. Using break points 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). Trailer video to be finished. |
| 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 (food), 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. |