Abel Hoxeng, Matt Pessa, Mitchel Pulley

Design Doc Project 2

Dr. Wallace

*Dog Warriors*

Today’s gamer wants to experience a game that opens their eyes to new experiences and interests while still evoking a sense of nostalgia. Unfortunately, many games in today’s market lack a new sense of creativity and the fortitude to push the limits. This lack of confidence is not something that is felt by Team Koala who feels that the limit is where game design should start. This is why a game like *Dog Warriors* is essential for today’s gamer. This document will explain many of the concepts required to make *Dog Warriors* a success. These key concepts are as follows: the basic idea of the game, the user interface, the entities, player goals, the low/high bar of what this game will achieve, development strategy, and a technical showpiece to catch the eye of individuals that are not normally interested in video games. Unfortunately, no game is easy to create, so the sticking points and ideas of how to overcome these issues that may come up during development will also be discussed.

**Basic Ideas of Dog Warriors**

The Basic idea of *Dog Warriors* is to create a high intensity role playing action game. The main character Spike has recently had all of his toys stolen by ninja cats. To his or her surprise, Spike is no ordinary dog. Spike was trained by Shaolin Priests and will stop at no means to get his toys back.

*Dog Warriors* will use the world coordinate system to move the main character around the screen. He will encounter enemy entities, which will switch from the world screen to a platform-based screen. The world screen will be reminiscent of classic RPG games such as the Final Fantasy series. The platform-based screen will be similar to that of classic games like Super Mario Bros or Faxanadu. In this static platform based screen, the user will control Spike in combat against the formidable felines. Spike will gain experience from fighting with cats. As he moves through the game, the hero will gain experience to advance his fighting style.

**User Interface**

The main idea for the user interface is broken up into three sections, the world map, platform map, and key controls. The world map will consist of the player-navigating Spike to locate his toys. Along the way, the hero will encounter simple enemy cats which, when they collide, will result in the platform world. The controls for the world map will use the following keys:

* The “A” key will move the player to the left.
* The “D” key will move the player to the right.
* The “W” key will move the player up.
* The “S” key will move the player down.

Once Spike has reached his toys, he will progress to the next level in the world.

If Spike collides with an enemy cat, the game will switch states to the platform world. In this world, Spike will fight and have to defeat all of the cats. If he doesn’t, he will lose a life and restart the platform level he is currently on. As Spike levels up, he will be provided with more options on the attacks he can perform. The controls for the platform work are as follows:

* The “A” key will move the player to the left.
* The “D” key will move the player to the right.
* The “W” key will make the player jump.
* The “S” key will allow the player to drop down from a platform.

The controls for the upgraded abilities will be:

* The “J” key will allow Spike to do a dropkick attack.
* The “K” key will allow Spike to shoot a water ball at enemies.
* The “L” key will allow Spike to have a bubble shield for protection.

Spike will be provided with a health, experience, and slobber bar. The health bar

will decrease as Spike takes damage from cats. As Spike defeats enemy entities, he will gain experience in order to receive new power ups and gain levels. Spike’s slobber bar is how he implements his special attacks.

**Entities**

The entities in the game will be comprised of Spike, at least three different colored enemy cat entities, dog treat entities, short-term invincibility entity, water ball entities, fireball entities, and the bubble shield.

Spike is the hero of the game who is trying to retrieve all his toys from the Evil Kitty Clan. The user will control spike with the keyboard and will attempt to keep him from dying. The player will try and keep Spike from making contact with the cats in the world map, but if they are unsuccessful they will have to fight the kittens on the platform map.

The enemy kittens are the evil villains of the game. They have decided to take Spikes toys and add them the their kitty collection. There will be three different colored cat entities. These three different classes will use different attacks in their attempt to keep Spike from his toys. These entities will only interact with Spike.

The dog treat entity is what spike will pick up to regenerate his health. These entities will be provided in the platform-based world. As Spike defeats cats, the occasional dog treat will be dropped in order to keep Spike from ending up dead. Only Spike can interact with dog treat entity.

The invincibility entity will allow for Spike to take damage without losing any health. When Spike defeats a cat, he has a small chance of an invincibility entity dropping. Only Spike can interact with the invincibility entity

The water ball entity is an attack for Spike. This entity when used correctly will repeal a cat entity by causing them to get wet. Since Spike is firing the water ball, he can't be hurt by it. The water ball can only interact with the cat entities or fireball entities.

The fireball entity is an attack for certain cat entities. This entity will be thrown from cats to inflict damage onto Spike. Their own fireball cannot hurt the cats, and the fireball can only interact with Spike or water ball entities. In the event of a collision between a fireball and a water ball, there will be a small explosion and both entities will disappear.

The bubble shield will protect Spike from cat attacks. This entity will last for one hit to keep Spike from taking damage. The only entity that can use the shield is Spike and the shield will detect collisions with the cat entities and fireballs.

**Goals**

The player will be provided with short term, medium term, and long-term goals. These goals are essential to keep the player’s interest and keep them playing. These goals will be static throughout the game, and will be the same each time the player plays.

The short-term goals are to keep Spike alive and try and reach the end of each world map. Achieving this goal requires Spike to survive battles and level up as needed.

The medium term goal is to reach the end of each of the two worlds. This can be done by avoiding all the cat enemies, but will make later battles more difficult since the player didn’t level up. As the player moves through map world the number of enemies on the world map will increase, and the number of enemies on the platform world will increase.

The long-term goal is to complete all the levels and defeat the last boss at the end of world map 2. By accomplishing this, Spike will retrieve all his toys and show the cat population he is not to be taken advantage of. Once Spike completes all the levels the game will end and the player will be notified of his or her completion of the game.

**Sticking points**

No game is ever as easy to implement, as it seems it will during the planning stage. Many problems can present themselves and they must be addressed to have a successful game.

The first issue is not having a smooth transition from the world map to the platform map. This issue must be solved or the whole game will have to change. The player doesn’t want to waste time while the program loads from one map to another. This should be overcome with states for both the world and platform maps.

Another issue for *Dog Warriors* is the physics for the platform world. It is essential to ensure that Spike can jump realistically. Using the proper mathematical formulas to come up with a simulated gravity for the game will do this.

Another issue will be the firing of water balls from Spike and fireballs from Cats. The issue with this is that the attacks must only affect certain entities. Also these attacks must look the same every time. This sticking point must be solved or the game will lose one of its main wow factors.

The final, and most complex, issue is creating the randomized maps for each level. There will be one entrance and one exit for each level and a path must connect them. There must also be multiple paths that lead to dead ends or are loops to increase the enjoyment of game and provide some challenge. The path to the exit must also be long enough that the player has time to move around the map and interact with the cat entities.

**Low Bar/High Bar**

The low bar is as follows:

* Initial low-bar will contain 2 world levels. These two levels will be randomly generated.
* Enemies will be harder on each progressive map and some enemies will have similar abilities to the learned abilities of the player. These abilities include fire balls, powerful kicks, and shields
* There will be a final boss at the end of the last level to earn the final chew toy. This boss will be stronger than all the other cats and will require more than one successful attack to be defeated
* There will be at least three different platform worlds for the player to interact with. These worlds will have a random amount of cats in them for the player to fight with.
* There will be at least three power ups obtained by the player for leveling up. These have been referenced earlier in the paper.

The high bar will be an expansion of the low bar but more complex. We plan to create additional levels, hopefully a total of five to six levels, as well as adding random treasure chests around the map that will increase the stats of a player. These increases could be as simple as a permanent addition of twenty point of health or something more along the lines of decrease the slobber required for special abilities by five percent. The increases would be cumulative also. So finding two chests that give the five percent decrease in slobber required will result in the player not requiring ten percent less slobber to implement abilities. Another idea that we are considering for our high bar implementation would be to have moving vehicles on the roads. The player would be damaged when hit by a car. Our last little bit for the high bar is including a save game option. So a player can enter back into their story right where they left off. This is an essential part of most games, especially RPG style games.

**Development Strategy**

This game will be created with very little code from previous games. Some of the code that can be reused are sprites for fireballs and for water balls, code for creating tile-based maps, and code for how to implement the built in A star algorithm. One of the hardest parts of this game will be creating the platform-based world. This is because basic physics will have to be taken into account. The dog jumping needs to look as life like as possible. It is important to ensure that the dog doesn’t look like he is jumping on the moon with how the movement looks. The technical showpiece will also be a challenge. This is because a lot of key factors go into creating a randomized map. Some of these factors include the map looking like a maze and not just a wide-open space with a few obstacles in the way. Another consideration in ensuring the player isn’t blocked in on all sides by walls when he or she spawns into the game. The last challenge is coming up with a great story that will keep the player interested. There are already many games like *Mario Bros.* on the market so it is imperative to create a story that people like and want to follow.

Many goals need to be achieved in a timely manner for a game like this to be a success. The planning stage for *Dog Warriors* has already began and will continue till Monday November eleventh. This is to ensure that everything has been properly planned out and no big issues have been left open that can cause issues later on in development. From there a week will be dedicated to creating the world map and platform map. The team will split up to work on these two areas with Abel Hoxeng working on the world map and Matt Pessa working the platform map. While Mitchel Pulley works on the algorithm to randomly generate the map world. Once the world has been created, time will be used to create all of the entities for both worlds. This will start of simple with just being able to have entities move around on the world map and also on the platform map this will require time being spent on physics for the platform world. This should be able to be accomplished by Monday November twenty-fifth. Over the Thanksgiving break, the team will work to install all the fighting and leveling up for the game. Matt Pessa and Mitch Pulley will work on the fighting while Abel Hoxeng works on the leveling up portion of the game. Monday December second the alpha release will be presented to demo. This will allows users to test the game and to give constructive criticism on how to make the game better. After this event, the team will work on fine-tuning any issues that hasn’t been solved, and fix any major issues that the alpha testers find.

**Technical Showpiece**

Not everyone who will be attending the launch of *Dog Warriors* is a video game fan and it is because of this that *Dog Warriors* will be creating randomized maps to catch their technical interest. As discussed in sticking points, these randomized maps provide a goal to find on the map that will be different every time the player plays the game. The randomized map will consist of road, houses, trees, grass, and fences. This will be difficult because the program will have to ensure it puts tiles in places that make sense.

The following constraints will have to be taken into account when creating a map. It is important that the program can’t create a piece of road in the middle of a house. The program will have to know what are the maximum amounts of certain tiles that the game can use. This is important because the game will not be realistic if the entire map is a road. It will be a challenge to ensure the town after it has been generated actually looks like a town. The generator will also place the cats into certain locations for interactions with the player. This is essential to ensure that cats aren’t place in areas where they will never interact with the player. The player will still be located in a maze where they are trying to find the chew toy, but it won’t look like a maze it will look like a town. It is also important to ensure that there is a path to reach the chew toy. If a map is generated where the player never is able to reach his or her goal they will not play after too long.