Jasmine Emerson
Kelly Keniston
Priya Kudva
Audrey Ramberg

Project 2 Design Document

Game Overview

1. How will the game be played?

Left arrow key - move Kirby left
Right arrow key - move Kirby right
Up arrow key - make Kirby jump, and if he's already in the air, make him flap his arms once for each press of the spacebar to gain altitude
Down arrow key - make Kirby fly down
Spacebar - suck up enemy/throw up enemy
Z key - swallow enemy - initiate special ability of his playing state if he is not in the default playing state
X key - revert to default playing state

2. What types of interactions are possible?

-Jumping off and colliding with platforms. Platforms will have their own gravity physics.
-Kirby absorbing enemies within a certain range
-Kirby holding enemies in his mouth and absorbing them for their abilities. These states include:

- Sword State
  - Stab in front
- UFO State
  - Fly and shoot bullets with beam power with limited range
- Beam State
  - Shoot a beam cone in front with limited range
- Sleep State
  - Make Kirby sleep
- Tornado State
  - Throw a tornado in front of Kirby
- Bomb State
  - Throw bombs in a parabola
- Cutter State
  - Throw a boomerang in front, if not caught it goes behind Kirby
- Fighter State
  - Punch in front of Kirby
- Hammer State
  - Swing a hammer in front and behind
- Spark State
  - Spark all around Kirby

        - Fire State
            - Breathe fire in a cone in front

-Collision with enemies or their projectiles will cause Kirby to lose health.
-Enemies colliding with Kirby's projectiles will cause them to lose health.

3. What are the visual entities in the game?

 Kirby

Enemies

-  Blade Knight (sword)
    - Uses Dijkstra's path planning to come within range of swinging its sword at Kirby
-  UFO (ufo)
    - Uses Dijkstra's path planning to collide with Kirby
-  Cappy (no power)
    - Hop around randomly on a platform
-  Waddle Dee (no power)
    - Uses Dijkstra's path planning to collide with Kirby
-  Waddle Doo (beam)
    - Approach close enough to shoot a beam in front
-  Scarfy (no power)
    - Jump around randomly
-  Noddy (sleep)
    - Walk up to Kirby

-  Bronto Burt (no power)
  - Fly up and down across the screen
-  Twister (tornado)
  - Speech towards Kirby after a few seconds from coming into the frame
-  Poppy Bros (bomb)
  - Uses Dijkstra's path planning to come within range to throw a bomb at Kirby
-  Sir Kibble (cutter)
  - Approach Kirby and slice in front of itself
-  Knuckle Joe (fighter)
  - Approach Kirby and kick in front of itself
-  Bonkers (hammer)
  - Swing around himself
-  Sparky (spark)
  - Spark around itself
-  Hot Head (fire)
  - Spit fire in a cone in front of itself

The platforms

4. What will the player do?

Players will be able to move Kirby around the screen by walking, jumping, and flying. They can also initiate Kirby's ability to absorb enemies when they come within a certain range of them. When an enemy is absorbed, Kirby enters a new playing state reflecting that enemy's abilities that the player can use to advance further into the level. Kirby resets to his default state if the player chooses to revert back, or until he reaches the end of the level.

5. What makes this idea interesting, or why do you think this will be fun?

This idea is interesting because it incorporates most of its variety in gameplay and states in the enemies, and each enemy in turn can change the state of Kirby himself. Kirby can have a variety of different power states, and each of his power states has a different animation/design for Kirby, different way of moving, different attack, etc. Kirby can also choose to abandon a power whenever the player feels like it and go back to his default state or get a new power. The enemies and levels are all eccentric and cute and are a good deviation from the usual guns and swords gameplay of a lot of games. The game is fun because it challenges the player to get through the levels while exploring the different powers that Kirby can have and to beat the bosses.

Development Strategy

Networking:

- Initial TCP networking to allow for complete packets to be sent across the server to each client. This will cause lagging, but it will make the initial processing of data simpler.
- Once the connection between the server and the multiple clients is established, we can utilize UDP networking to allow for faster data transfer and connection.
- For the layout of the network, we will use the smart server/dumb client model so that each client doesn't have to recreate the map for every frame

Map:

- Map is grid of tiles, with each storing its image, if it is an obstacle, etc
- Player and characters also represented as 1 or more tiles based on their size

- Have the art for the entirety of the map for each level but only allow a certain portion of it to be shown based on the player's position
- Smooth tile based, there are tiles in the map but Kirby can move smoothly through them and isn't restricted to standing in just 1 "tile space"
- Collisions still determined by tilemap, but allows for slopes and smooth jumping
- Make character's bounding box AABB, decompose movement into X and Y axes and determine objects that are colliding and if no collision, then character can move
- One way platformer, Kirby can jump into obstacles above him but can't fall through obstacles below him

Enemies:

- At least 10 different enemies that move and attack on their own, some traversing a set path, and some using path-planning towards the player's location using Dijkstra's algorithm.

Player:

- Kirby's movements and actions respond to the player's controls.
- At least 10 different states Kirby can switch to depending on which enemies he consumes.
- Proper physics for Kirby's movement/flying
- Proper health and damage
- Proper bounding box and collisions with walls and different types of enemies

Development Milestones

Nov 18: Basic creation of the map, Kirby and enemies, initial testing of networking
Nov 25: Interaction between all entities
Dec 2: Alpha Due Date (should work, but will likely be buggy and networking will be choppy)
Dec 9: Beta Due Date (better representation of the desired product, but could have bugs and the graphics will not be pretty)
Dec 16: Final Due Date (smooth graphics over the network and minimal to no bugs)

Technical Showpiece

This version of Kirby will have networking to allow multiplayer mode. Unlike games such as Pac-Man or Defender, this game will require a server to talk to each client in a way that there is both smooth data transfer and minimal lag. Data that will be transferred will include enemy, projectile, and player positions. Because of the nature of Kirby's abilities, there will be more player states involved than the aforementioned games. PacMan only has his two states, but Kirby will have 10 or more that includes unique animations and images, attacking abilities, and movement methods. In addition, unlike PacMan, this game will be a platform game, where there is a much larger map than just the size of the screen that is in PacMan. The map must be updated along with the position of Kirby and is much more complicated than the simple layout such as in PacMan.

See point 5 as to why it is more interesting.

High Bar

- 20 different enemies
- 20 different playing states for Kirby
- Full animation sequences for every character and action
- 3 or more levels
- Boss at the end of each level
- 3 more players for multiplayer capabilities

Low Bar Checklist

- Sidescrolling platformer
- 10 different enemies
- 10 different playing states for Kirby
- At least 2 levels
- At least 1 boss at the end of level 2
- Networking - at least 1 other player that can play Kirby