




# GAME DESIGN DOCUMENT: RESILIENT KIWI MAGNIFICENT DREAM



Authors: Kadin Honeyfield, Maya Ashizumi-Munn, Robert Dumagan & Devin Grant-Miles  
RKMD AUT Game Programming Team

## Contents

Game overview .....	3
Gameplay .....	4
Mechanics .....	4
Vertical scroller .....	4
Infinite Gameplay.....	4
Downwards Gameplay .....	4
Power Ups and Collectable Currency .....	4
Enemies .....	4
Platforms.....	4
Hazard avoidance.....	4
Combat/Shooting.....	4
Player wrap around.....	4
Winning (and losing) the game .....	4
Rules.....	5
Controls.....	5
In game.....	5
In menu .....	5
Xbox 360 controller alternatives.....	5
Player view .....	5
Key game algorithms.....	6
Score keeping, levels and lives.....	6
Enemy AI Movement, States and Behaviour .....	6
Movement algorithms .....	6
State algorithms .....	6
Behaviour algorithms.....	7
Platform placement generation.....	7
Player movement physics .....	7
Horizontal movement .....	7
Vertical movement.....	7
Player states.....	7
Camera movement .....	7
Collecting power-ups .....	7
Levels.....	8
Enemies.....	8

Ground enemies.....	8
Flying enemies .....	8
Enemy data .....	8
Platforms.....	9
Static platforms.....	9
Bouncing platforms.....	9
Crumbling platforms .....	10
Moving platforms.....	10
Mock-up User Interface .....	11
Control scheme .....	13
Required assets.....	14
Sprites/Art.....	14
Audio .....	14
Particles.....	14
Animations .....	14
Data driven design .....	15
Game Sections .....	15
Aesthetic .....	16
Other notable planning sketches/brainstorming .....	17
Cheat Features .....	20
Infinite Jetpack.....	20
Never Fall .....	20
Invincibility.....	20
Infinite Money.....	20
Background chooser .....	20
Bibliography .....	21
Team Sign Off.....	22

## Game overview

### **Resilient Kiwi: Magnificent Dream** **Game Pitch by RKMD Game Programming Team**

**Resilient Kiwi: Magnificent Dream** is a vertical scrolling platformer. You play as a kiwi who dreams of flying higher than any bird has flown before. The aim of the game is to collect as many points as possible before the game is over. The player gets points from both climbing higher and through collecting kiwifruits. Once the player falls, they get another opportunity to collect as many kiwifruits as possible on the way down. High scores are placed on a leader board.

Since kiwis can't fly, so the player must use the platforms in the game's environment to get as high as possible. Screen creep slowly applies pressure on the player to move upward. Kiwi can single jump and double jump-flap to move to each platform. The typical range of platforms are implemented including bouncy platforms, crumbling platforms and moving platforms.

The game world changes as the player ascends upwards. Kiwi starts in a forest environment which changes to a city landscape, to mountains, to clouds and finally, to space. Where possible, the objects and enemies will change with each new environment such as:

- Branch platforms in trees
- Window cleaning platforms in the city
- Cloud platforms etc.

Kiwi will also encounter enemies as he traverses the world. Again, the enemies will be relevant to the current environment such as pests (rats, stoats) in the forest, birds in the clouds etc. Kiwi has the ability to shoot enemies with green laser beams that shoot from his eyes.

Kiwi can collect powerups on the way up which enhance and/or change the gameplay. Such as:

- Kiwifruit – powerups, coins
- Gumboots - spring boots for higher jump
- Hardhat + hi-vis vest - invincibility
- L&P – upside down bottle jetpack

The game art will be vibrant, cartoony and cute, with lots of fun animations and particle effects.

Sounds will have an arcade feel and the music will be upbeat, light-hearted and have an adventurous/triumphant feel to it to match the game's premise.

**Target platform:** PC

**Target audience:** Anyone over the age of 5

**Genre:** Platformer/Arcade/Casual/Action

**Unique Selling Points:**

- Kiwiana – learn about New Zealand icons and culture
- Simple and easy to play
- Cute and customisable kiwi character
- Reversible gameplay

## Gameplay

### *Mechanics*

#### Vertical scroller

The game will constantly be scrolling vertically so the player needs to stay within the camera view to not lose the game.

#### Infinite Gameplay

Along with the vertical scroller mechanic, the game will be developed in a way where the players can infinitely play the game.

#### Downwards Gameplay

A unique feature of having to glide back down vertically once the players lose will also be included to give players a second opportunity to collect as much coins as they can.

#### Power Ups and Collectable Currency

The game will feature New Zealand themed power ups that will be collectable while playing the game such as:

- Spring loaded Gumboots
- Hardhat and a Hi-Vis vest – Invincibility
- L&P Soda Jetpack
- Kiwi fruits – Coins for in-game currency

#### Enemies

The game will have enemies of different types, land-based enemies such as possums and stoats will be positioned on platforms and must be either shot by the player or landed on top of to be destroyed. Airborne enemies such as seagulls and meteors will travel across the screen and need to be shot or the player can allow them to pass, however that will cost the player time.

If an enemy collides with the player, the player enters the falling state and the run is over.

#### Platforms

For the player to survive, they will need to jump on platforms above them to climb to safety.

#### Hazard avoidance

There will be enemies in the game for the player to try and avoid to not lose the game.

#### Combat/Shooting

The player can shoot either, directly up, left or right using separate keys. The player can shoot while jumping allowing them to shoot an enemy on a platform above the player.

#### Player wrap around

The player can jump off the screen and re-appear at the other side. This mechanic will be needed for some of the levels to progress.

#### *Winning (and losing) the game*

Winning: Infinite game design allows the main goal of the game or players to beat the previous high score and collect as much coins to upgrade the players power ups. The first time the player reaches space, the game credits will roll and acknowledge the player has

completed the final level of the game. But will be able to continue upon completion of credits.

Losing: Upon enemy impact or fall off the screen

### *Rules*

- Climb as high as you by jumping on platforms and using powerups.
- When the player falls, they can fall back down collecting coins on the way down.
- No platforms are spawned during the falling phase
- Do not collide with enemy NPCs
- If the player falls beyond the screen, they will enter the falling state.
- While using the jetpack powerup the player is invincible
- Cheats will not allow the player to have their score added to the high score leader board.
- The player's falling distance will match the player's height climbed.

### *Controls*

#### In game

A – Move left  
D – Move right  
SPACE – Jump  
Left arrow key – Shoot left  
Right arrow key – Shoot right  
Up arrow key – Shoot up  
Escape – Pause/Un-pause

#### In menu

Left click – Select option  
Escape – Quit game

#### Xbox 360 controller alternatives

- Controller
  - D-Pad (left and right) – Movement
  - A – jump
  - Y – shoot up
  - X – shoot left
  - B – shoot right
  - Start – Pause/Un-pause

### *Player view*

The view will be 3<sup>rd</sup> person, where the camera will follow the player on the Y axis (whenever the player moves vertically).

## Key game algorithms

Painter's algorithm – creating a game environment using many layers of sprites, shapes and UI icons

Main game loop – Runs constantly while the game is being played, the main tasks of the game loop is to process the user's inputs, update game states when necessary, renders the graphics of the game, processes any changes needing to be made, and controls the rate of the game play using delta time. (Nystrom, 2009). This main game loop processes everything to create a game simulation to the player.

Particle Manager: 3 types of particles, landing on platform, killing enemy, collecting power-up

Menu State machine – State machine for menus will be created, so different input handling, processing and drawing can happen for each different menu.

Animated Sprites – Animated sprites will be simulated using sprite sheets, and looping over those sprites to create a moving sprite simulation.

### *Score keeping, levels and lives*

The game will have the players highest score recorded and the current score updated live while playing. The Kiwi will only have one life with each playthrough.

However, as the Kiwi progress they will also encounter enemies as he/she traverses the world. Again, the enemies will be relevant to the current environment such as pests (rats, stoats) in the forest, birds in the clouds etc.

- Shoot green laser beams

### Collectables/powerups

- Kiwifruit – coins that are the game currency for the player to collect and spend in the game shop
- Gumboots – higher jump
- Hardhat, hi vis vest - invincibility
- L&P – upside down jetpack
- Powerups influence – gameplay (icy tower,
- Kiwifruit items could have an upward momentum boost effect

### *Enemy AI Movement, States and Behaviour*

#### Movement algorithms

All enemies will pace back and forth to simulate idling behaviour. Aggressive enemies will attack the player by moving in a linear interpolation towards the player.

#### State algorithms

- PACING: The enemy paces back and forth waiting for the player to come into their range for attack.
- ATTACKING: The player has entered the enemy's territory, so the enemy will start moving towards the player using linear interpolation in hopes of colliding with the player and making the player lose.
- DEAD: The enemy has been shot down, or killed somehow by the player and does not exist anymore.

### *Behaviour algorithms*

When the player is in some visible range of the enemy, it will switch to the attack state and start attacking.

### *Platform placement generation*

Either pre-set sections loaded through file parser OR randomly generated platforms based on feasibility bounds. Most likely pre-set level sections which are algorithmically selected based on level/the previous section.

### *Player movement physics*

#### *Horizontal movement*

One speed moving left and right.

#### *Vertical movement*

When the jump button is pressed and the player is grounded, the Y velocity will increase for a short period of time – making the player jump and traverse vertically. While already in the jump state, the player cannot jump again while mid-air, but can move side to side.

### *Player states*

Jumping, falling, shoot left, shoot right, shoot up, moving, standing

States are determined by players velocity as well as buttons pressed.

### *Camera movement*

Initially, the camera will only move to follow the player character. The camera will move vertically to follow the player if the player moves above the bottom half of the screen.

When the player successfully completes the forest level, then the camera will start automatically slowly moving up to challenge the player. The speed of automatic camera movement will be determined by how many levels they have surpassed.

### *Collecting power-ups*

When the player collides with a power-up item on a platform, the power-up effect will apply to the player for whatever duration has been set for that power-up.



## Levels

These will be the themes of our levels, which will seamlessly integrate with each other. There will be no menu switching, but the levels will be:

- Forest (native trees)
- City (Sky Tower)
- Mountains
- Clouds
- Space

These levels will differ in the types of platforms that spawn, enemies, and background. The same type of power ups will exist in all levels.

## Enemies

There will be different enemies for each level. Most of these enemies will have some sort of AI to control its movement or attack. Here are the different types of enemies that will appear:

### *Ground enemies*

- Passive – Will not do anything when you are near, but player collides with it, the player will lose.
- Aggressive – Will attack when the player is near it, either shoot or melee attack and on successful attack of a player, the player will lose.

### *Flying enemies*

- Aggressive – Will attack and chase the player and attack in intervals.

### *Enemy data*

Enemies will have some attributes that give them different characteristics. Here are these described below:

All enemies will at least have:

- Hit points (int): The health of the enemy, when attacked by the player, this value will go down. Once it is at or below 0, it is dead.
- Movement speed Category (EnemySpeed enum): A category of how fast the player will move. Enemy Speed Enum: SLOW, MEDIUM, FAST. These will be preset numbers.
- Movement speed (int): Will be set in the constructor based on what movement speed category has been applied.

These are the unique enemies per level for each type

- Ground passive
  - Forest: Possum
  - City: Rat
  - Mountains: Sheep
  - Cloud: Unicorn
  - Space: FLEEB

- Ground aggressive:
  - Forest: Weta
  - City: Cat
  - Mountains: Goat
  - Cloud: Buzzy Bee
  - Space: Alien
- Flying aggressive:
  - Forest: Morepork
  - City: Seagull
  - Mountains: Kea
  - Cloud: Air NZ Plane
  - Space: Rocket ship

## Platforms

Platforms are the floating, elongated objects scattered across the game screen where the player will need to jump from platform to platform to survive.

### Static platforms

Most of the platforms in this game will be static. They will not move, break, or make the player bounce. Initially, only one type of static platform per level but could be open to adding more types in the future

*Static platforms for each level:*

- Forest: Branches with green leaves
- City: Building ledges
- Mountains: Rocky cliff edges
- Cloud: Clouds
- Space: Asteroid

### Bouncing platforms

Bouncing platforms will allow the player to bounce up in the air once landed on the platform. Bouncing platforms will be static platforms with a static entity/object sitting on the platform to jump on and give the jump boost.

*Bouncing platforms for each level:*

- Forest: Sheep (regular)
- City: Sheep with suit
- Mountains: Sheep with horns to look like a horn
- Cloud: Cute cloud sheep that can be rainbow or pink
- Space: Space sheep with helmet

### Crumbling platforms

Crumbling platforms will break after 3 seconds of the player being on it. It will shake while the player is landed on it.

#### *Crumbling platforms for each level:*

- Forest: Dying branches with yellow/brown leaves
- City: Cracking building ledges
- Mountains: Cracking rocky cliff edges
- Cloud: Rain cloud
- Space: Broken asteroid

### Moving platforms

Moving platforms will move side-to-side or up and down, despite the player being on top of it or not.

#### *Moving platforms for each level:*

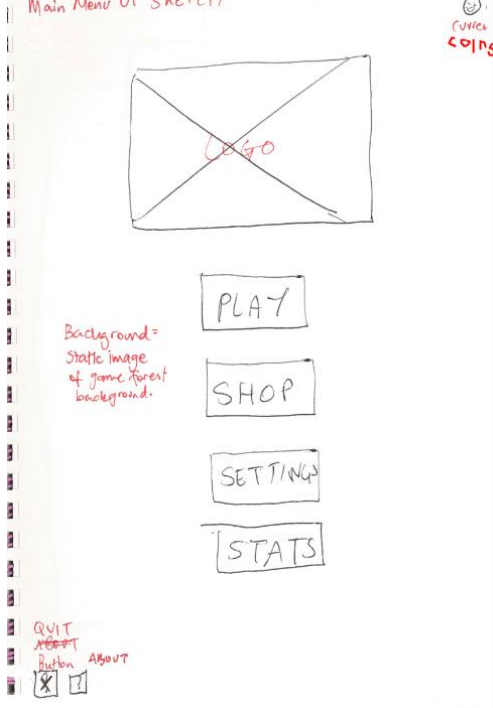
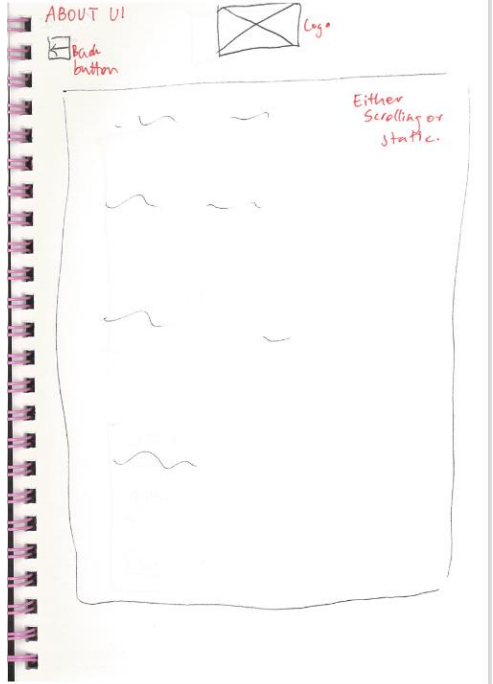
- Forest: Buzzy bee
- City: Window washing platform
- Mountains: Ski lift
- Cloud: Westpac helicopter (up and down) Air NZ plane (left right)
- Space: Rocket powered metal platform

## Mock-up User Interface

The window sizing will be: 854 px height by 480px width.

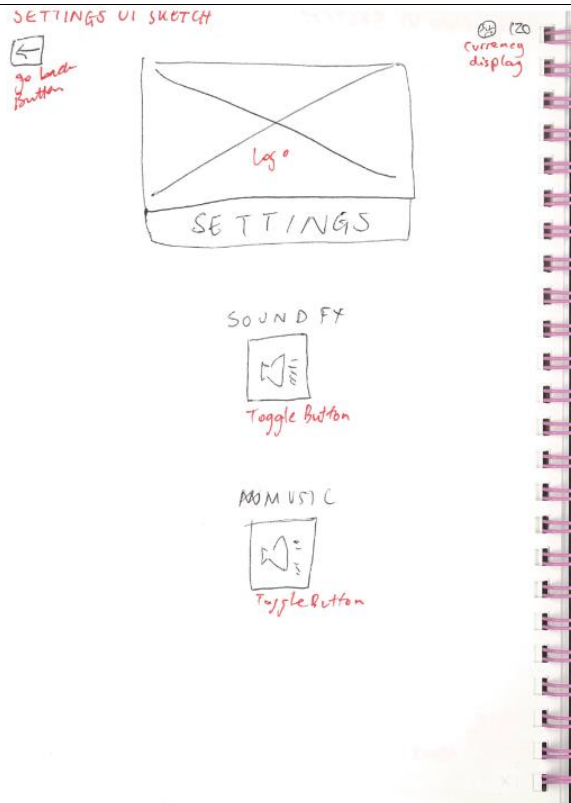
Things to sketch

- Player on the most ground level
- Player in between 2 levels (forest and city) to demonstrate how they are integrated
- RKMD splash screen logo?
- Game over screen

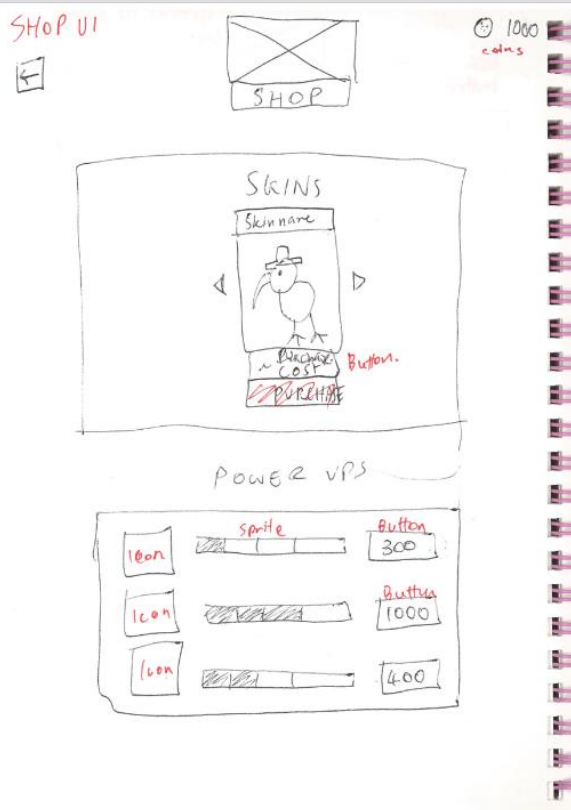
Description	UI Sketch
<p><b>Main menu interface</b></p> <p>This is the first menu the user will see after launching the game. It will be displayed after the splash screens have displayed.</p>	 <p>Main Menu UI SKETCH</p> <p>Background = Static image of game forest background.</p> <p>QUIT ABOUT RUSH ABOUT</p> <p>COINS</p>
<p><b>About page interface</b></p> <p>There will be an 'About' page accessible from the main menu where the player will be able to see game credits</p>	 <p>ABOUT UI</p> <p>Back button</p> <p>Either Scrolling or static.</p>

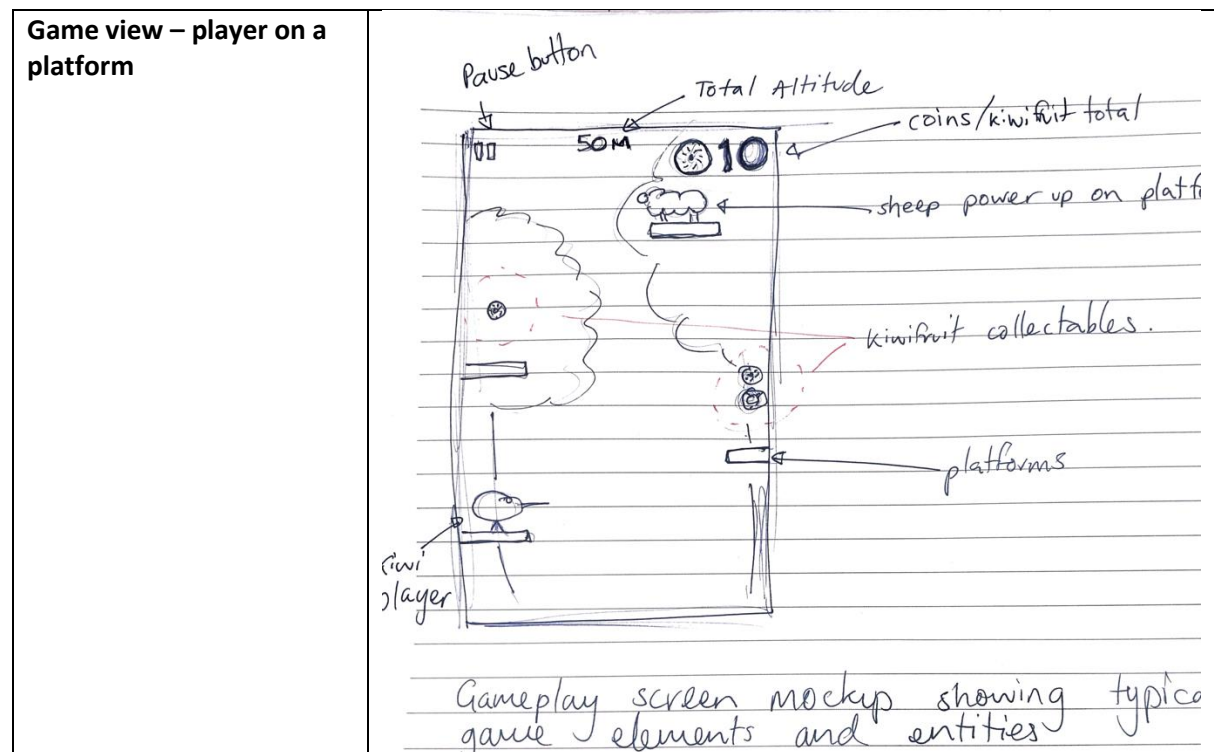
**Settings user interface**

This will be the settings menu for the user to toggle sound on and off

**Shop interface**

This will be the shop where players can buy cosmetic upgrades to their kiwi or improvement of powerups





## Control scheme

These are the mappings for controls for keyboard and Xbox controller.

ACTION	Keyboard	Controller
Move left	A	Left joystick move left or LEFT DPAD
Move right	D	Left joystick move right or RIGHT DPAD
Jump	SPACE	A
Shoot Up	W	Y
Shoot Left	Q	X
Shoot Right	E	B
Open settings menu	ESC	START

## Required assets

### *Sprites/Art*

- Platforms
  - Tree branch
  - Mountain
  - Clouds
  - Moving
  - Space
- Interface - Sprite sheet
  - Buttons: static, pressed, toggled, disabled
  - Score Container
  - Coins Container
  - Content Container

### *Audio*

- Sound effects
  - Coin collection
  - Bullet fire
  - Enemy damage
  - Kiwi damage
  - Kiwi jump
  - Power up collection
  - Platform crumble
  - Platform bounce
- Background music

### *Particles*

- Leaves
- Wool
- Feathers/fur
- Collectable particles (kiwifruit, L&P fizz)

### *Animations*

- Flapping kiwi
- Jumping kiwi
- Falling kiwi
- Kiwi look right
- Kiwi look left
- Shooting kiwi
- Enemy deaths
- Enemy moving
- Enemy attacks
- Platform crumbles

## Data driven design

### *Game Sections*

Predefined platform positions rendered by the game in random order providing randomness. It will also specify for each section, how the following will spawn:

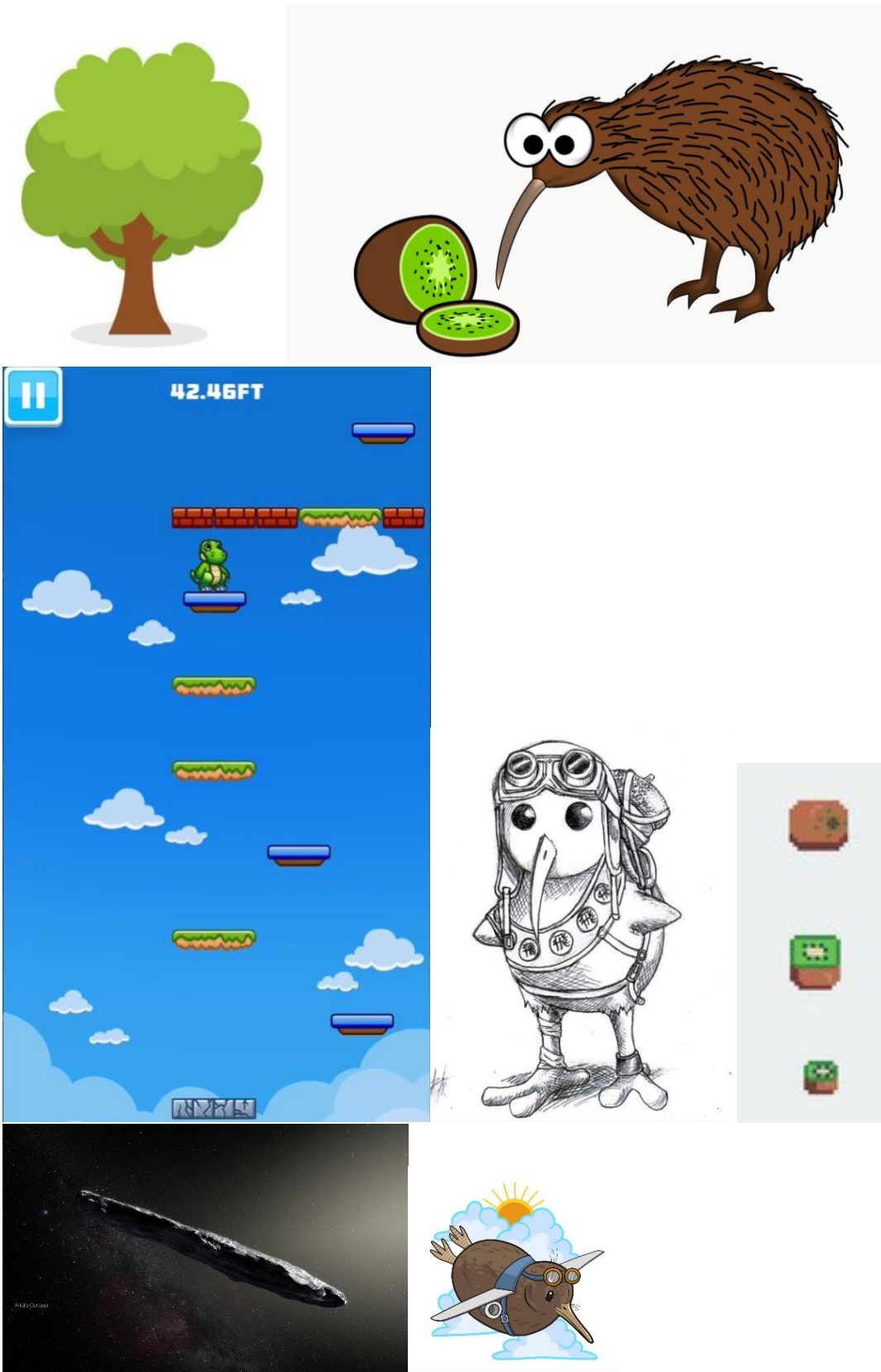
- Item placement
- Platform placement
- Enemy placement

The sections will be created in some sort of text file that each character will represent an item in a section, then parsed through an INI parser to load into our game.



## Aesthetic

Game visuals should be cartoony and vibrant. Kiwi and other characters and enemies should be cute.

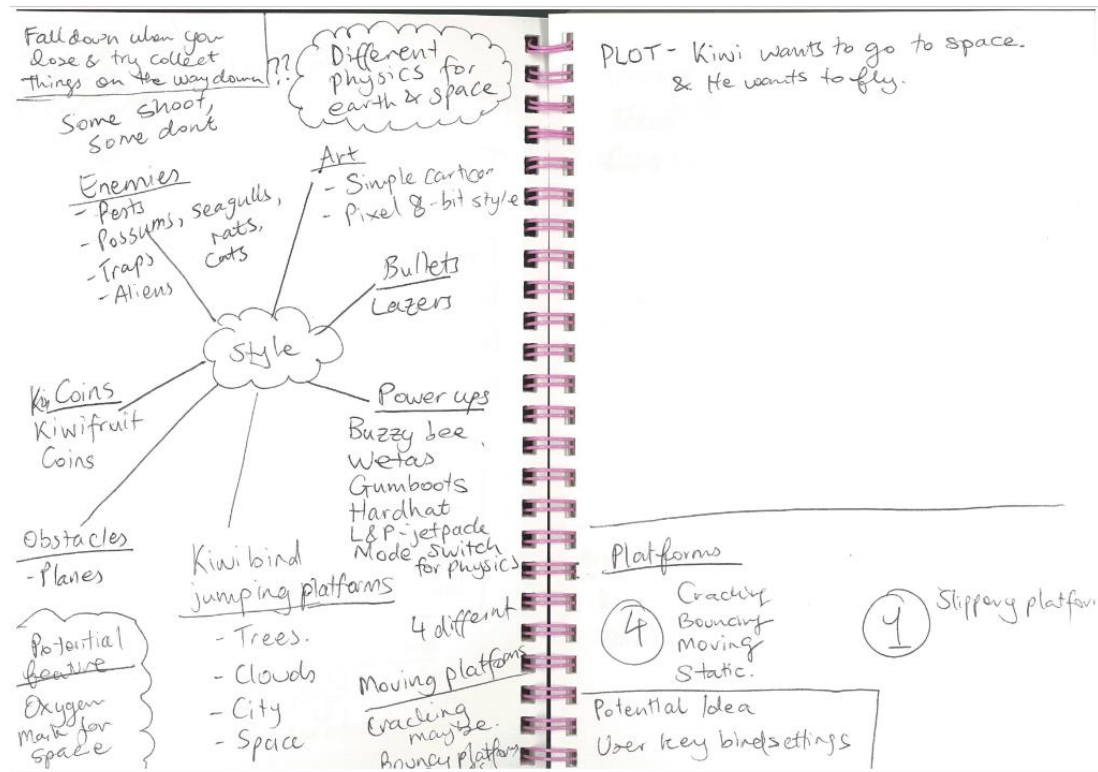


<https://www.google.com/url?sa=i&url=https%3A%2F%2Fexploringkiwis.com%2F&psig=AOvVaw0uYNk1wjs2X0clw14C7sqy&ust=1602451058678000&source=images&cd=vfe&ved=0CAIQjRxqFwoTCOiRo8f5quwCFQAAAAAdAAAAABAE>

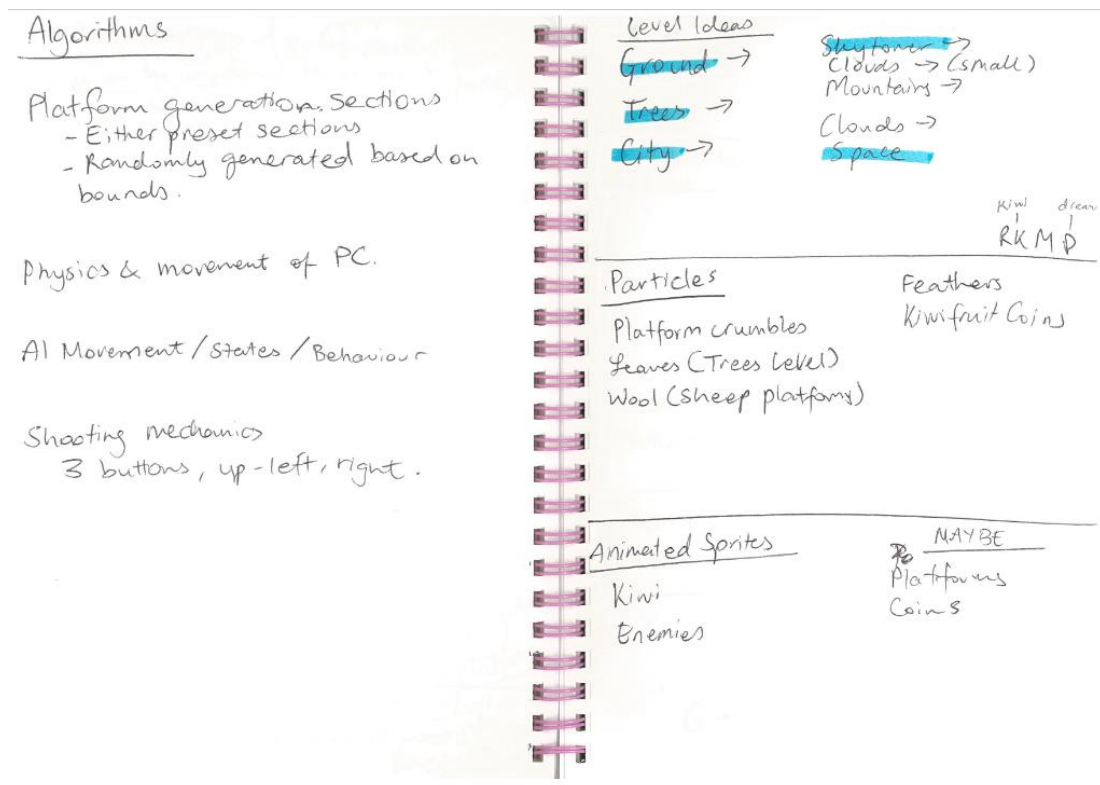
<https://www.deviantart.com/tobitkiwi/art/A-Kiwi-and-an-Acorn-64517176>

## Other notable planning sketches/brainstorming

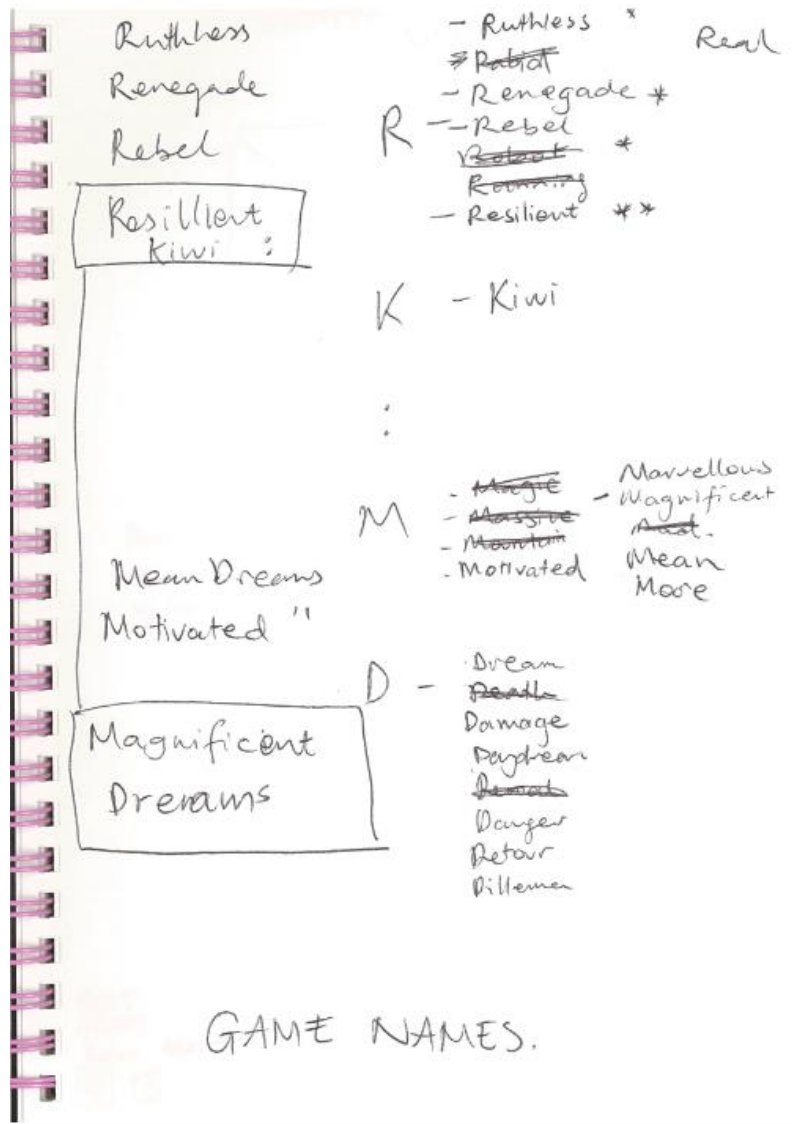
- 1) Brainstorming the theme and style of our game based on our initial abstract idea of wanting to make a vertical, scrolling platformer.



- 2) Identifying possible algorithms to define in our game and listing some levels, particles and animated sprite ideas we can implement in our game



3) Trying to figure out a name for our game based on our team name acronym



## Cheat Features

Cheat features are activated by pressing F key, but is subjective for change in the future.

When a cheat is active the player cannot have their score added to the leader board.

### *Infinite Jetpack*

A jetpack that never runs out.

### *Never Fall*

Platform spawns underneath the player when they fall beyond the bottom of the screen.

### *Invincibility*

Allows the player to never die from enemy impact.

### *Infinite Money*

Gives the player unlimited money.

### *Background chooser*

Change the background to start from.

## Bibliography

Hooper, S. (2020). *COMP710 Game Programming: W05 Studio Session 08 Lec 08c - Individual Game Project, 2D Game Exemplars* [PowerPoint slides]. Blackboard. <https://blackboard.aut.ac.nz/>

Nuclear Monkey Software. (2004). *Narbacular Drop: Game Design Document*. DigiPen (USA) Corporation.  
[http://www.nuclearmonkeysoftware.com/documents/narbacular\\_drop\\_game\\_design\\_document.pdf](http://www.nuclearmonkeysoftware.com/documents/narbacular_drop_game_design_document.pdf)

Nystrom, R. (2009). *Game Programming Patterns: Game Loop*.  
<https://gameprogrammingpatterns.com/game-loop.html>

## Team Sign Off

All team members agree with the content in this Game Design Document for RKMD game.

Date signed: Friday 2<sup>nd</sup> of October, 2020.

Maya Ashizumi-Munn:

A handwritten signature in black ink, appearing to be 'MA' followed by a stylized flourish and a period.

Kadin Honeyfield:

A handwritten signature in black ink, appearing to be 'KH' followed by a stylized flourish.

Devin Grant-Miles:

A handwritten signature in black ink, appearing to be 'DGM' followed by a long horizontal stroke.

Robert Dumagan:

A handwritten signature in black ink, appearing to be 'RD' followed by a stylized flourish.