Unity Game “Dreamland” Development

Sijie Meng

1. Project description

Develop the Unity2D game Dreamland on the computer in C# language and run it on the Android mobile phone. Dreamland is a puzzle game designed to provide a relaxed and enjoyable game experience for players of all ages. The core gameplay of Dreamland is to tap the screen to control the character and earn scores and diamonds. The top three best scores will be displayed in the ranking scene, and the diamonds can be used to purchase the character’s skin.

2. Development conditions

- Development language: C#
- Development environment: MacOS
- Development tools: Unity
- Graphics and audio: Photoshop CC, QuickTime Player

3. Game demand design and analysis

(1) Game main menu

- Start game
- Change skin of game character
- View rank
- Sound settings
- Reset game

(2) Game scene

- Pause or continue the game
- Display current score
- Display current number of diamonds
- Generate the prefabs of character and platforms

(3) Game over scene

- Display final score
- Display best score
- Display the number of diamonds
- View Rank

- ♦ Go back to main menu
- ♦ Restart game

(4) Shop scene

- ♦ Display total number of diamonds
- ♦ Display different skins
- ♦ Buy skins
- ♦ Use the selected skin to play the game
- ♦ Back to main menu
- ♦ Hint insufficient diamond

(5) Rank scene

- ♦ Display the top three best scores

(6) Reset scene

- ♦ Provide options for resetting

4. Main menu scene implementation

(1) Create user interface

- ♦ Image
- ♦ Particle System
- ♦ Animation
- ♦ Button
- ♦ Text

(2) Create main menu scene script

- ♦ MainPanel.cs

5. Game scene implementation

(1) Create user interface

- ♦ Image
- ♦ Animation
- ♦ Button
- ♦ Text

(2) Create prefabs

- ♦ Character prefab
  - Rigidbody 2D

- Box Collider 2D
- Circle Collider 2D
- Audio Source

♦ Death effect
- Particle System

♦ Diamond
- Rigidbody 2D
- Polygon Collider 2D

♦ Platforms
- Rigidbody 2D
- Box Collider 2D

(3) Create random scene background

♦ Background.cs

(4) Create management container

♦ ManagerVars.cs

(5) Generate platform and combined platform

♦ PlatformSpawner.cs

(6) Generate game character and role control implement

♦ PlayerController.cs

(7) Create game management class

♦ GameManager.cs

(8) Camera tracking implement

♦ CameraTracker.cs

(9) Create platform script

♦ PlatformScript.cs

(10) Create object pool

♦ ObjectPool.cs

(11) Generate random diamond and obstacle

♦ PlatformSpawner.cs

(12) Collision detection

♦ PlayerController.cs

(13) Game over judgement

♦ PlayerController.cs

(14) Instantiate death effect

- PlayerController.cs

(15) Game scene script

- GamePanel.cs

6. Game over scene implementation

(1) Create user interface

- Image
- Particle System
- Button
- Text

(2) Create game data class

- GameData.cs

(3) Display the score and number of diamonds

- GameManager.cs

(4) Create game over scene script

- GameOverPanel.cs

7. Shop scene implementation

(1) Create user interface

- Image
- Animation
- Button
- Text
- Scroll Rect

(2) Serializable game data class

- GameData.cs

(3) Buy and change the skin

- GameManger.cs
- ShopPanel.cs

(4) Hint scene pop up

- Button
- Image

- Text
- Hint.cs

(5) Shop scene script

- ShopPanel.cs

8. Rank scene implementation

(1) Create user inrerface

- Image
- Button
- Text

(2) Sort and store scores

- GameManager.cs

(3) Rank scene script

- RankPanel.cs

9. Reset scene implementation

(1) Create user interface

- Image
- Button
- Text

(2) Reset scene script

- ResetPanel.cs

10. Add sound and sound effect

(1) Create sound class

- Audio Source
- ManagerVars.cs

(2) Create sound scripts

- ButtonAudio.cs
- BestScoreMusic.cs
- BuySound.cs
- CantBuySound.cs

(3) Music on and of controller

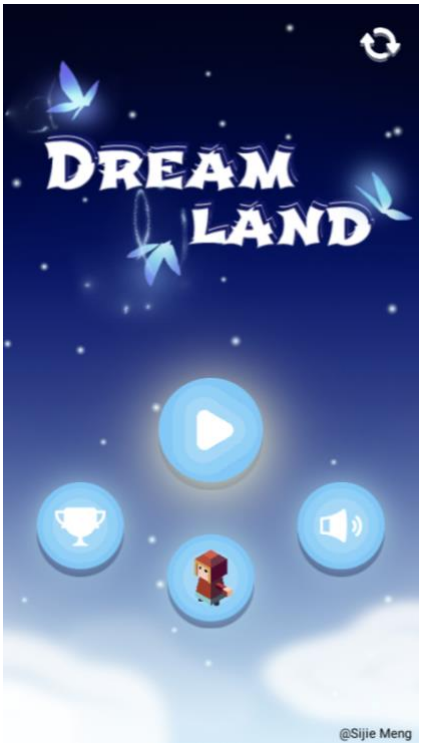
- GameManager.cs

11. Build and run on Android phone
    (1) Install sdk and jdk
    (2) Export the game to apk file
    (3) Install on the phone
    (4) Test and debug

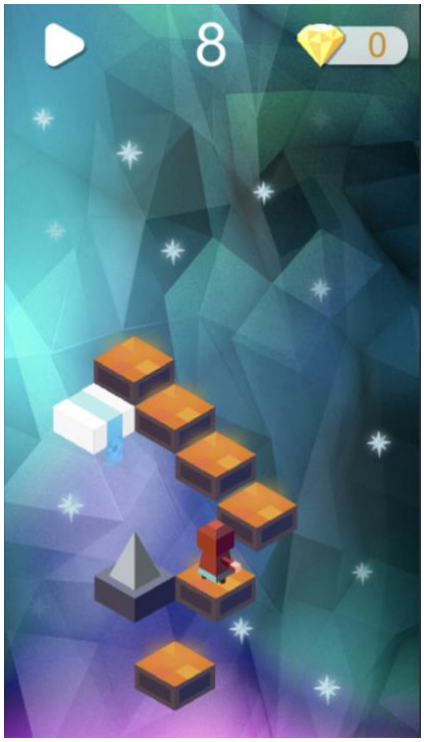
12. Game results show

(1) Game logo

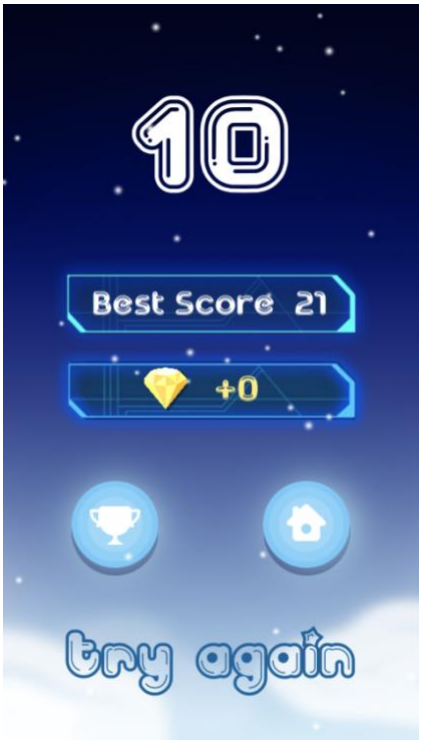
(2) Main menu scene

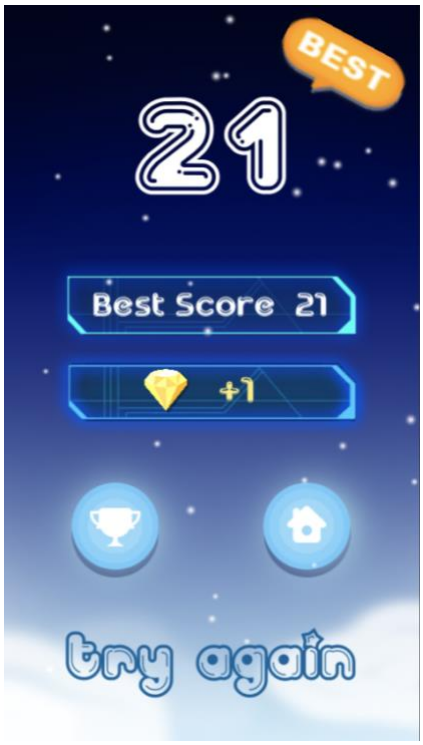


(3) Game scene

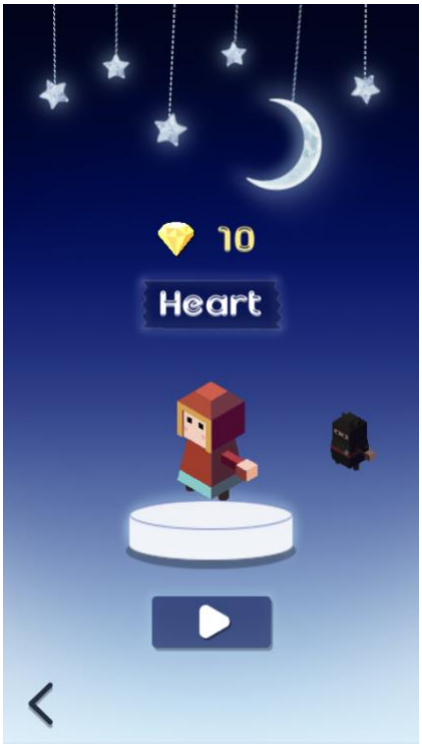


(4) Game over scene





(5) Shop scene


10
Heart

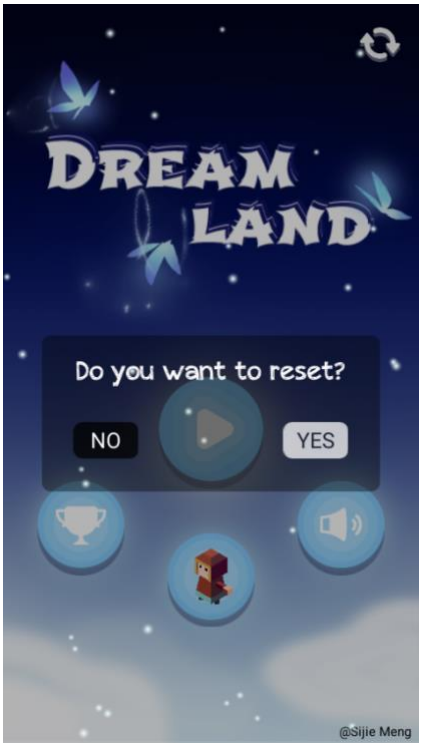

10
Groot
5


5
Groot


0
Ninja
Oops!
You didn't have enough diamond!
2

(6) Rank scene

(7) Reset scene



13. Summary

The game demands and functions are all implemented, which includes the game user interface and image production, C# code implementation, background music and sound effect production, game effect and animation production. The game can be run on both personal computer and mobile device.