Harbin Institute of Technology
Interactive Design and Development of Digital Media
Course Report

Song Jian

July 2018

Contents

Student No.	Name	email
1150310303	Song Jian	2511146542@qq.com
1150310704	Gong Tianying	540655966@qq.com

1 Introduction

The primitive archetype of the game was a simplified 2-D parkour game. In the game, the player acted as a somehow clumsy dinosaur and aimed at the goal to run as long as he can while avoiding incoming obstacles and perils. Plain as the original game was, we were enlightened to decorate the unornamented game with some more enjoyable machinations. We created a battle system and a perk and ability system to complex the game and set a reachable goal to incite players to try harder until the goal is accomplished. The player, in this modified game version, choose a perk and act as a ninja or shooter, dealing with meandering ghosts whose collisions are hostile and a dragon who shoots mortar bullets at you. Surviving and killing monsters grant you credits and there is an ending of the game, though not all players might be able to see it and begin to doubt its existence.

2 Game Description

The player controls a character to shoot enemies and avoid damage, in the meantime running and surviving before your health points reach zero. Both implementation and gameplay details are shown below. One common knowledge, the game is completely designed by us and powered by Unity3D engine and we used the programming language C#.

2.1 Character

The two characters, named Ninja and Gunslinger inherit from the abstract class Character respectively, are available currently in the MainMenu scene.

2.1.1 Ninja

Ninja, the primary character as indicated by the game title, is a flexible and powerful pawn. Despite the low damage of his darts, Ninja experts in defending and can even parry bullets.

Triple Shot Throw three darts continuously with a slight interval, and naturally implemented with Coroutine

Split Shot Throw three darts in three different directions, providing good protection from close targets

Twice Jump Ninja can jump twice before landing, and the second jump is a flip

Blink Sprint through enemies, dealing damage and avoiding damage of physical collisions with ghosts

Defend Rebounds bullets from enemies

2.1.2 Gunslinger

Gunslinger can shoot .500 Magnum bullet with distance penalty, high damage per one shot, but low fire rate.

Shoot Shot .500 Magnum bullet, with the max damage of 120 and min damage of 50. The numeric value of the damage is linearly waned with the incrementation of the distance the bullet spreads

Shield Can parry one shot when the shield is ready. Cooldown time is 15s

Bullet Eye Slow down time for most actions for 1.4s, during which Gunslinger can shoot for at most three times

2.2 Enemy

There are two kinds of enemies derived from the abstract class Enemy, called Ghost and Patriarch.

2.2.1 Ghost

Ghost is a small enemy spawned at a random height and floats up and down. Albeit not heavily armored, their collisions can cause you 30 damage and your total vitality is 100. Note that there are not medic packages betwixt the way you travel, so shooting at the ghosts is an excellent option.

2.2.2 Patriarch

Patriarch, the unique boss in this game, can shoot 3 bullets at one time. Well the good news is one patriarch killed grants you 1500 scores, while the bad news that a patriarch slashed means a following one coming soon, so get prepared for an everlasting battle.

3 Technical Skills

Here are some basic and high-level skills used in the game.

3.1 Unity Engine

Physics As required by the original game, we use Rigidbody2D and Collider2D to generate physics effects, including falling from above and collision detection.

Coroutine Coroutine in Unity Engine is highly useful and efficient to generate latency of a fixed period of time. Examples of Coroutine can be found almost everywhere, such as throwing three darts with a little interval, spawning enemies every several seconds, and defending from bullets for 1.4 seconds.

UGUI All GUI objects are made with UGUI, including Text, Button, Canvas, CanvasGroup, Image and so on. To perform universal but not perfect UI on machines with varying resolution ratings, we use RectangleTransform.anchor to control the horizontal and vertical ratio of one object according to its parent.

Sound The collocation of AudioSource and AudioClip enables the game to radio waves when one shot or a hit happens. In addition, the background music is "Stronger".

Sprite Sprites are the significant part of one 2D game. We use some external tools (like Photoshop) to draw and cut the pictures required.

Prefab We use prefabs to instantiate game objects dynamically with code instead of doing such by dragging game objects from asset folders to scene view.

Other important methods Enumerate other methods used frequently.

- GameObject GameObject.Find(string)
- T GetComponent <T>()
- GameObject Instantiate(GameObject, Vector3, Quaternion)
- Coroutine Object.StartCoroutine(method)
  Note that we use method and parameters instead of nameof(method) for performance issues

3.2 Language C#

Based on .NET 4.7.1, we use C# 6.0 to accomplish the workloads of scripts.

inheritance There are three abstract classes, Bullet, Enemy and Character for other child classes to derive from, with most of their methods virtual, including OnCollisionEnter2D, OnTriggerEnter2D, Awake, Start, MoveForward, Update at least.

encapsulation Some of the member variables are defined as private, while others are public. Though there are six kinds of accessability in C#, we only used public, protected and private (ample in most occasions).

polymorphism Virtual methods of abstract classes are widely override in derived classes.

yield Keyword *yield* are generally used to create coroutines. In most occasions, we used clause yield return new WaitForSeconds(1.4f); to yield an interval of 1.4 seconds.

lambda expression C# provides three kinds of lambda, while only two of them are available in C# 6.0, normal lambda and member lambda except getter and setter lambda. Clause public int DistanceScore =>(int)(survivalDistance * distanceScoreWeight * distanceScoreFactor); is a good example of member lambda, used to define a public getter property.

delegate and event Delegate and event are useful attributes and frequently used in this game. We used EnemyDieEvent to tell some objects one enemy just died, and the GUI class to refresh goals. Other events, such as CharacterSurviveEvent, BulletEyeCoolDownDecreaseEvent, also have their unique functions.

exception We only used UnityEngine.UnityException to raise an exception to avoid some compile error where an unexpected Enumeration object was inputted.

is and as To judge whether an object is an instance of a class dynamically, keyword *is* is suitable, after which we use *as* to explicitly cast the object to the specific class.

3.3 Other highlights

Health bar Instead of a static Text on the Canvas, we designed a health point bar, the length of which is automatically changed according to the health point of the player. In addition, the color of it modifies from green to yellow, and finally turns red when reaching a really low ratio, which is, technically speaking, implemented with Color.Lerp.

Cool down We show cool down in a friendly way to player: a grey image shrinking slowly with a big number on it indicating the exact cool down time. When the skill is ready, the 0 number is removed.

Rich Text We use Rich Text to make the description of characters more colourful for players.

4 Learning and Thinking

Through this assignment, we have learned a lot about game development and become more proficient in C#. I also developed a strong interest in this course. After finishing this course, we should continue to learn more advanced knowledge. Finally, this course at the meantime cultivated our sense of teamwork, during which we made new friends.
And we sincerely appreciate the assistance from our teacher.

5 Future Work

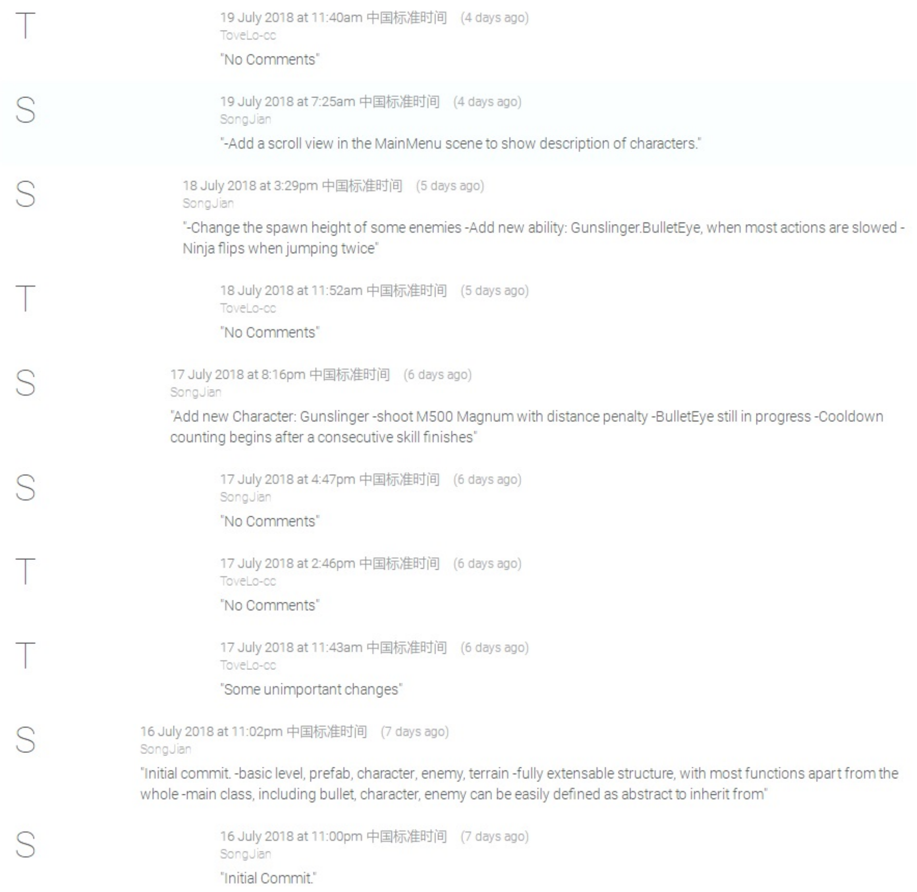
There are still some potential improvements in our assignment.

Achievement Achievements, however, push the players to play the game again and again in ways beyond our imagination. Nevertheless, the event system in our game is designed partially to support expansions of achievements and statistics.

Animation We failed to generate an animation for characters, such as running, blinking, shooting and so on. The experiment only showed to yield an animation with given key frames by scripting controllers, rather than to create the key frames.

Network Most games, including "standalone" ones, send data to the server and that is one goal we are still working at.

Particle System A fantastic game can never be without particle system. Still, more work can be done here.



T

19 July 2018 at 11:40am 中国标准时间 (4 days ago)
ToveLo-cc
"No Comments"

S

19 July 2018 at 7:25am 中国标准时间 (4 days ago)
SongJian
"-Add a scroll view in the MainMenu scene to show description of characters."

S

18 July 2018 at 3:29pm 中国标准时间 (5 days ago)
SongJian
"-Change the spawn height of some enemies -Add new ability: Gunslinger.BulletEye, when most actions are slowed -Ninja flips when jumping twice"

T

18 July 2018 at 11:52am 中国标准时间 (5 days ago)
ToveLo-cc
"No Comments"

S

17 July 2018 at 8:16pm 中国标准时间 (6 days ago)
SongJian
"Add new Character: Gunslinger -shoot M500 Magnum with distance penalty -BulletEye still in progress -Cooldown counting begins after a consecutive skill finishes"

S

17 July 2018 at 4:47pm 中国标准时间 (6 days ago)
SongJian
"No Comments"

T

17 July 2018 at 2:46pm 中国标准时间 (6 days ago)
ToveLo-cc
"No Comments"

T

17 July 2018 at 11:43am 中国标准时间 (6 days ago)
ToveLo-cc
"Some unimportant changes"

S

16 July 2018 at 11:02pm 中国标准时间 (7 days ago)
SongJian
"Initial commit. -basic level, prefab, character, enemy, terrain -fully extensable structure, with most functions apart from the whole -main class, including bullet, character, enemy can be easily defined as abstract to inherit from"

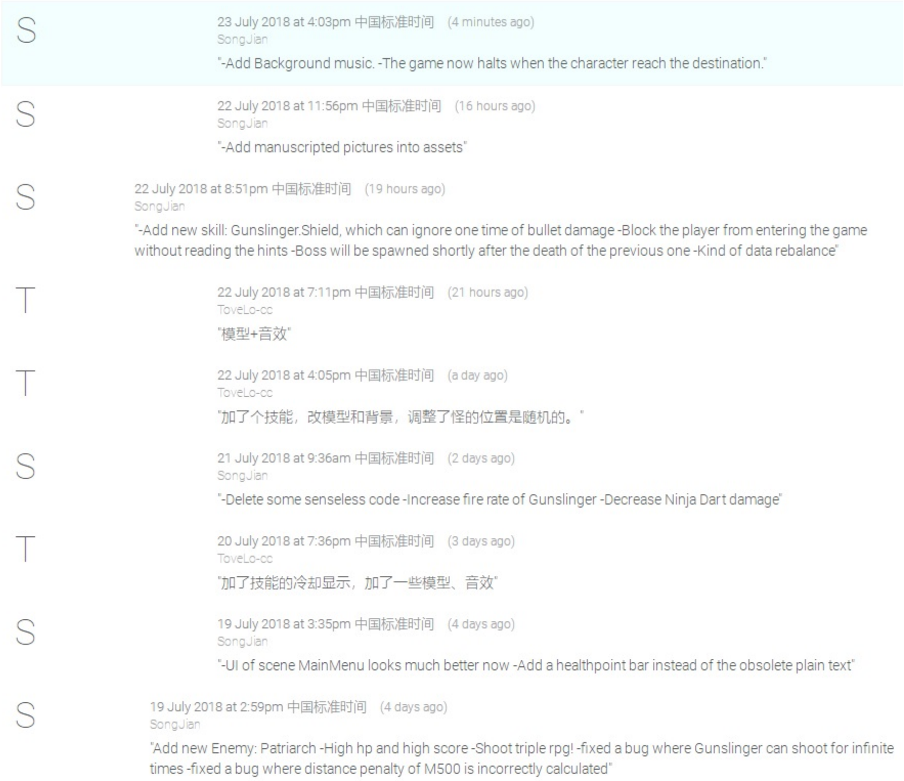
S

16 July 2018 at 11:00pm 中国标准时间 (7 days ago)
SongJian
"Initial Commit."

Figure 1: Collaborate History 1

6 Mission Distribution

Total contribution:
Song 60%
Gong 40%
During the development, we used the Unity collaborate to publish our manipulation respectively and merged our effects together. Here is the website of this project (NinjaRunning):
https://developer.cloud.unity3d.com/collab/orgs/mutationdelegate
/projects/ninjarunning2/ commits/
You can see our collaborate history here.



23 July 2018 at 4:03pm 中国标准时间 (4 minutes ago)
SongJian
"-Add Background music. -The game now halts when the character reach the destination."

22 July 2018 at 11:56pm 中国标准时间 (16 hours ago)
SongJian
"-Add manuscripted pictures into assets"

22 July 2018 at 8:51pm 中国标准时间 (19 hours ago)
SongJian
"-Add new skill: Gunslinger.Shield, which can ignore one time of bullet damage -Block the player from entering the game without reading the hints -Boss will be spawned shortly after the death of the previous one -Kind of data rebalance"

22 July 2018 at 7:11pm 中国标准时间 (21 hours ago)
ToveLo-cc
"模型+音效"

22 July 2018 at 4:05pm 中国标准时间 (a day ago)
ToveLo-cc
"加了个技能，改模型和背景，调整了怪的位置是随机的。"

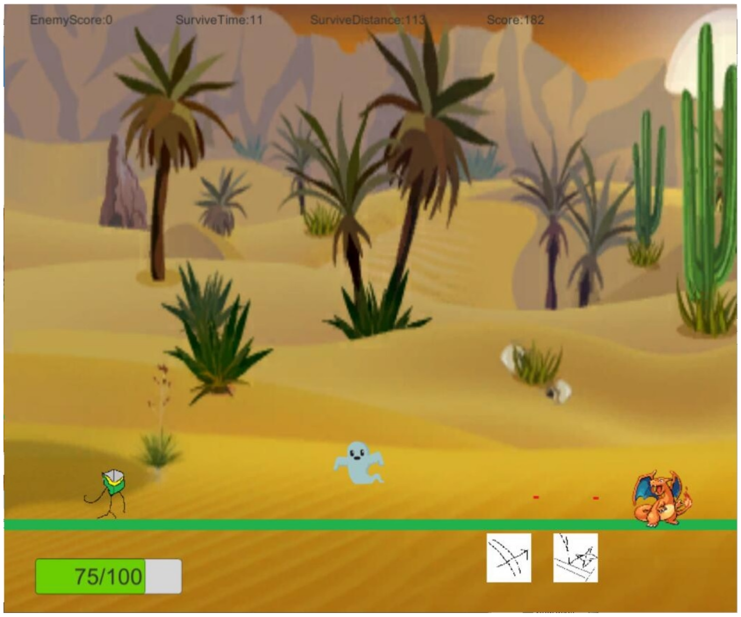
21 July 2018 at 9:36am 中国标准时间 (2 days ago)
SongJian
"-Delete some senseless code -Increase fire rate of Gunslinger -Decrease Ninja Dart damage"

20 July 2018 at 7:36pm 中国标准时间 (3 days ago)
ToveLo-cc
"加了技能的冷却显示，加了一些模型、音效"

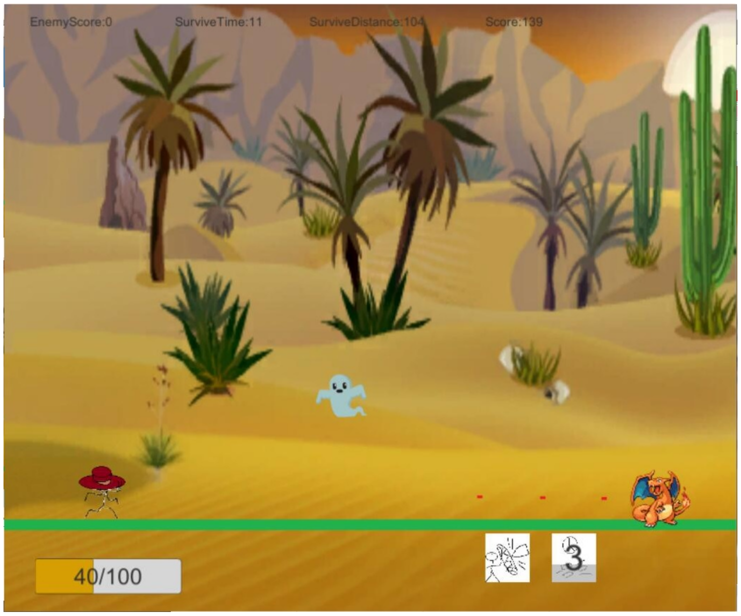
19 July 2018 at 3:35pm 中国标准时间 (4 days ago)
SongJian
"-UI of scene MainMenu looks much better now -Add a healthpoint bar instead of the obsolete plain text"

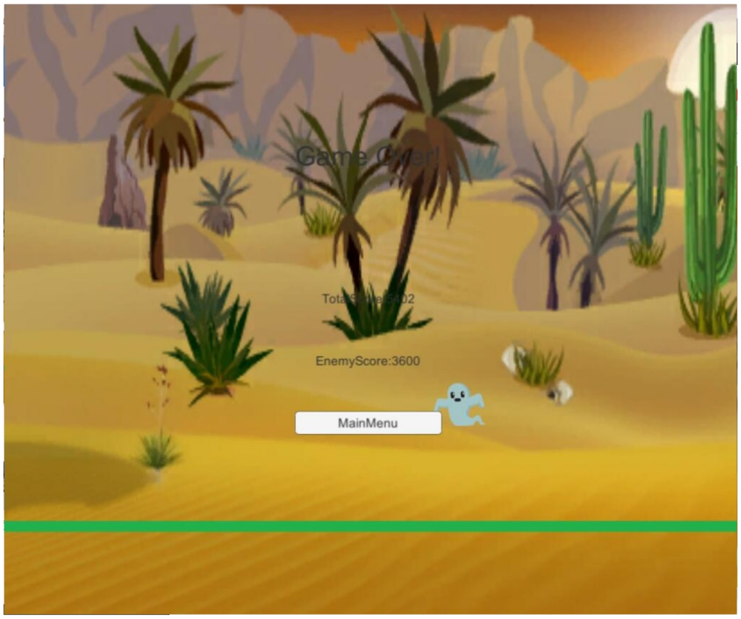
19 July 2018 at 2:59pm 中国标准时间 (4 days ago)
SongJian
"Add new Enemy: Patriarch -High hp and high score -Shoot triple rpg! -fixed a bug where Gunslinger can shoot for infinite times -fixed a bug where distance penalty of M500 is incorrectly calculated"

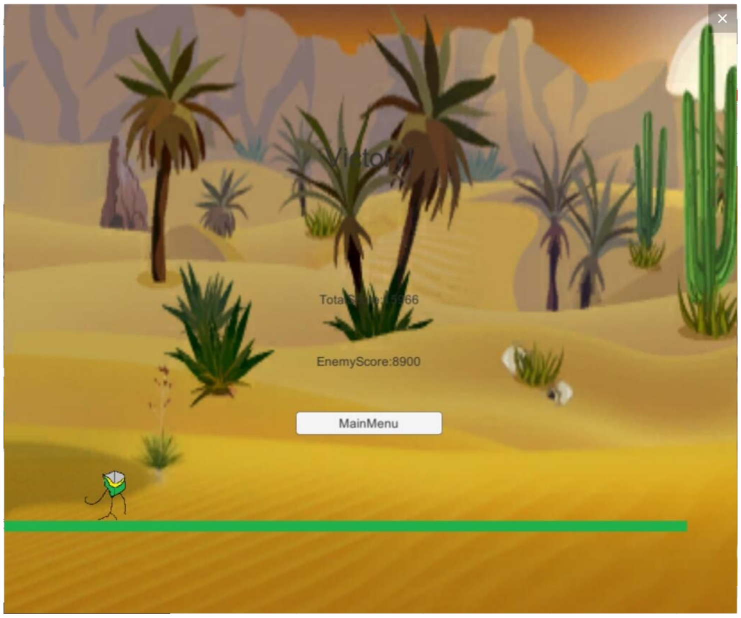
Figure 2: Collaborate History 2



7 Screen Shots


EnemyScore:0
SurviveTime:11
SurviveDistance:104
Score:139
40/100
3

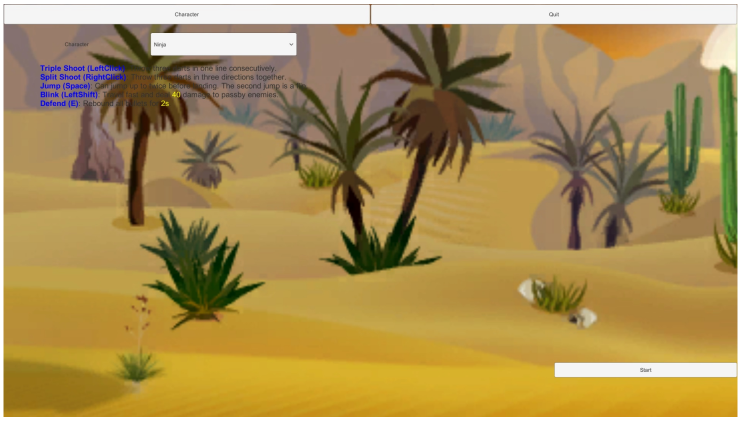

Game Over!
TotalScore:5402
EnemyScore:3600
MainMenu



Victory!

TotalScore:15966

EnemyScore:8900



Character: Ninja

Triple Shoot (LeftClick): Throw three darts in one line consecutively.
Split Shoot (RightClick): Throw three darts in three directions together.
Jump (Space): Can jump up to twice before landing. The second jump is a flip.
Blink (LeftShift): Travel fast and deal 40 damage to passby enemies.
Defend (E): Rebound all bullets for 2s.