



PANG 3D RECORD

TEQUILA DESIGN INTERN TEST

Alberto Blanco
alber09055@gmail.com

Content

Introduction	2
Game Description.....	2
Game Base.....	2
Controls	2
Keyboard	2
XBOX Controller:	2
Blueprints	3
PangGameMode.....	3
SideScrollerCharacter	3
GameInfo.....	3
LevelBlueprint (PangLevel).....	3
Laser	3
Ball.....	4
Platform.....	4
GUI and Menus.....	4
Menu	4
References.....	5

Introduction

For the Technical Design Intern Test I received from Tequila Works, I have created a version of the classic game PANG.

The purpose of this project is to demonstrate my abilities in videogame design and development, as well as my knowledge of Unreal Engine 4. For this reason, the developed game is a small product, with only one game mode and scenario, and no props or 3D models have been created outside UE4. Everything used on the game was created within the engine.

In this document I will explain the game flow, the design choices I made, and finally I will go through the blueprints in detail, for a better understanding of the project.

Game Description

For this PANG version there are no levels as such. There is only one scenario that is created based on the difficulty parameters (which I will explain later). Depending on this parameters, the number of balls, platforms and the spawn rate for the balls is different. The spawn locations are always random inside a bounding box.

Balls keep spawning every X seconds, and the game only finishes when the player loses all his lives.

Game Base

To start the developing process of the game I used the SideScroller template, so the moving controls were already implemented and simulating 2D was less time consuming.

Some changes have been made to this template:

- Changed the camera to an ortografic one, to simulate 2D.
- Disabled the Jump Control.
- Made some changes in the Game Mode Blueprint (explained in the blueprint section).

Controls

Keyboard

Move: A, D.

Fire: F.

Pause: Escape (Recommend to play on Standalone so Escape don't close the execution).

XBOX Controller:

Move: Left Joystick.

Fire: B.

Pause: Start.

Blueprints

In this section I will list all the blueprints I created and why I decided to do them the way they are.

All blueprints that were created or modified from the original template are located inside the Pang3DAssets folder, so they can be easily differentiated from the template content (except SideScrollerCharacter and PangGameMode).

All blueprints are fully commented inside the project, so this section will not go into functionality details.

Concrete design decisions will be marked in **bold** so they are easier to locate.

PangGameMode

In this blueprint I store the information and functionality of the punctuation and lives system. This way, if I want to do different game modes in the future it will be easier. **Every time we respawn the player, we delay the spawning of the character. Once we have spawned the character, it can't be hurt and can't shoot for one second, so the player has time to react.**

SideScrollerCharacter

It is the default SideScroller character but with some changes. It can't jump, and it can shoot laser. It also communicates with the Game Mode to update and retrieve information.

GameInfo

GameInfo is just a blueprint that stores persistent information between levels.

isMainMenu and isGameOver are Booleans that determine which Menu to show when we are in the Menu level.

Difficulty is a string that determines difficulty.

LevelBlueprint (PangLevel)

The level blueprint in the game level has different functionalities:

- Change the main camera to the static camera every time that the player respawns.
- **Read the GameInfo difficulty and initialize the game according to it. This means setting the number of balls and platforms to spawn, and the spawn rate for the balls.**
- **Spawn the balls and platforms in a random way using two Actors that determine the maximum bounds for the spawn.**
- Manage the HUD.

Laser

This is the object that the player shoots. I think it was the best 3D approach to the original pang gun.

The laser consists on a **mesh inside another mesh** (so I was able to change the pivot of the object without having to take it to blender or 3DS max) which scales on the vertical axis every tick. If the laser overlaps with a ball it calls the ball Split function (explained later) and it destroys itself. If it overlaps a platform or the ceiling, it only destroys itself.

When the player has shot a laser, he can't shoot again until the current laser is destroyed as in the original game.

Ball

The ball blueprint is the only one that manages every size of ball. It does this using a parameter on the constructor that indicates which size it is going to have.

For the bouncing, **the ball uses Impulse every time it hits a surface**. This decision was taken because the original game ball physics were pretty constant, and when using a physics material it was harder to simulate that. The impulse is calculated using the normal of the hit object.

When a new ball is spawned it receives a vertical upward impulse as well as the horizontal one, so if a ball is destroyed really close to the player, the new balls don't kill him.

Every time a ball is shot, we call its **Split** function, which checks if it can be split (size is bigger than 1) and if this is true, it spawns to balls in the same spot, but with opposite directions. Finally, the ball destroys itself.

Platform

A platform blueprint has been created, but it is a bit different to the original game, as **it is a movable platform**.

The blueprint consists of a mesh, which was created using UE4 BSP and then converted to a static mesh.

The movement is done using a direction (left or right) and a speed (between .5 and 1.5) which are generated randomly, and a maximum moving distance. When the platform hits a wall or another platform, or it reaches its maximum distance, it inverts its direction.

GUI and Menus

For this exercise I have created a Main Menu (called Menu), which executes on a different level, and a HUD (called GUI) which executes over the game and displays the remaining lives and the score. There is also a Pause Menu.

Menu

All the menus have been developed using the same Widget Blueprint, this means the Main Menu, the Configuration Menu and the Game Over Menu. This way it is easier and faster to navigate through menus.

In the configuration Menu we can change the difficulty and the resolution of the game.

Pause Menu

It is an overlay Menu that lets you pause the game and change the resolution or quit the game.

References

All the assets in the project have been created using UE4 Engine and its tools.

The character is the Unreal standard character.

The only two external assets are the font and the background image used in the menus, which were downloaded from the internet.

Personal Information

Name: Alberto Blanco Barrios.

E-mail: alber09055@gmail.com.

Github: <https://github.com/alber0905>

This Project github repository: <https://github.com/alber0905/UE4Pang>

Portfolio: <https://sickmyduckinc.github.io/>