

Flying Jump micro game v1.0

Features:

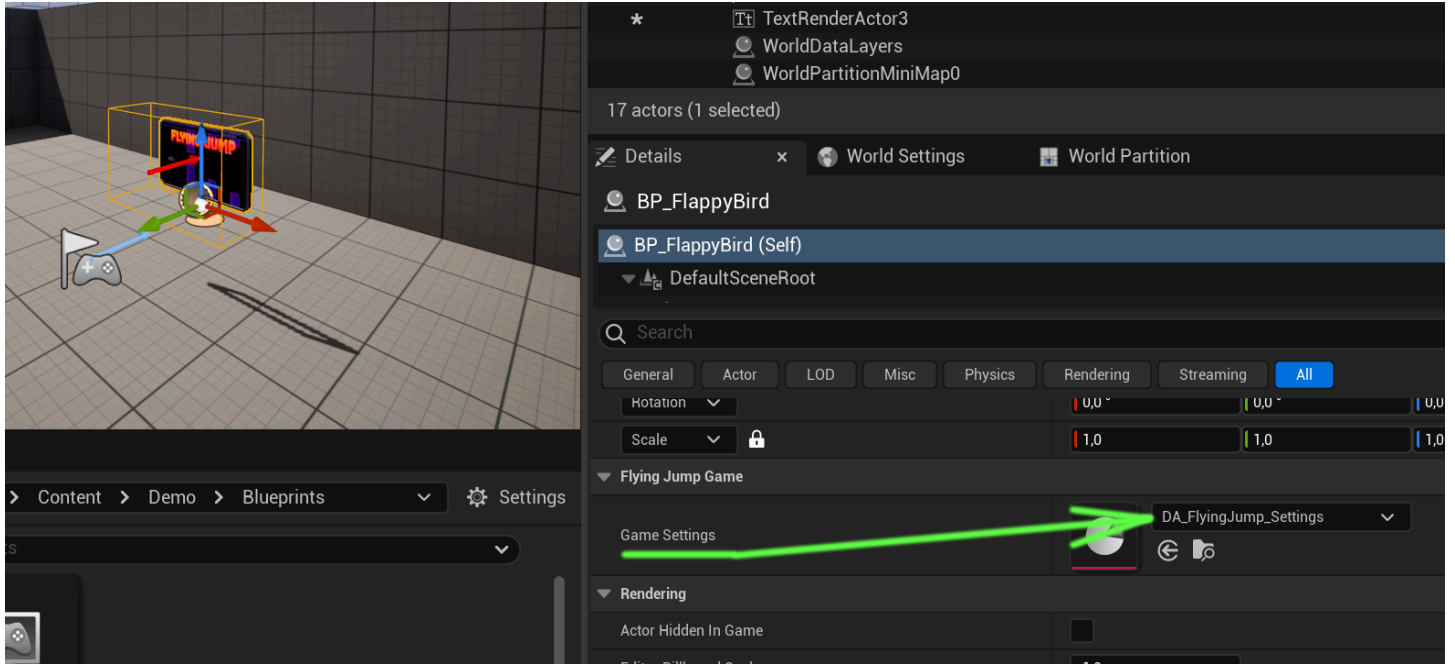
- True old school style included
- Full network support: Dedicated and Listen servers are supported
- Simple and fast art style customization (just replace textures)
- Leaderboard
- Fast rendering within material










By Andrey Harchenko
jocker3d@gmail.com

Settings

For each Flying Jump game instance you can find settings data asset here:



Detailed setting you can set inside of Content/FlyingJump/Blueprints/DA_FlyingJump_Settings.uasset

Default	
Lives	3
Speed	0,07
Speed Increment	0,005
Jumpspeed	0,4
Dropspeed	0,13
Jump Amplitude	0,45
Obstacle Distance	1,0
Score Increment	5
Start Obstacle Offset	1,871
Start Postion UFO	0,706
Num Obstacles to Skip in Safe Mode	2
Emissive Power	0,6
Score SFX	 FlyingJump_score
Game Over SFX	 FlyingJump_gameover
Loose Live SFX	 FlyingJump_Jooselife
Cheat Mode	<input type="checkbox"/>
Background Speed	0,6
Score Text	Score: 
Restart Text	Press R to try once again! 
Exit Text	Press E to exit game 
Start Text	Press E to start game 

Lives: Lives count. Min 1 lives and max 5 lives

Speed: Obstacle movement speed

Speed increment: Speed incrementation factor of obstacle speed after each passed obstacle

Jump Speed: Jump acceleration for Player object

Drop Speed: Player falling down speed

Jump Amplitude: Player jumping amplitude value

Obstacle distance: Distance between obstacles. Keep in mind, only 4 obstacles are supported. Too low a distance (all 4 obstacles on the screen) will provide an incorrect result!

Obstacle distance: Score increment when obstacle is passed

Start obstacle offset: Initial obstacle distance from player. **DON'T RECOMMENDED TO CHANGE IT TO LOWER VALUE!** Default value: 1,871

Start Position UFO: Vertical starting Player position

Num Obstacles To Skip In Safe Mode: How much obstacles must be skipped in safe mode

Emissive Power: Rendering material emissive power

Score, Game Over, Loose live SFX: Sound waves for game states

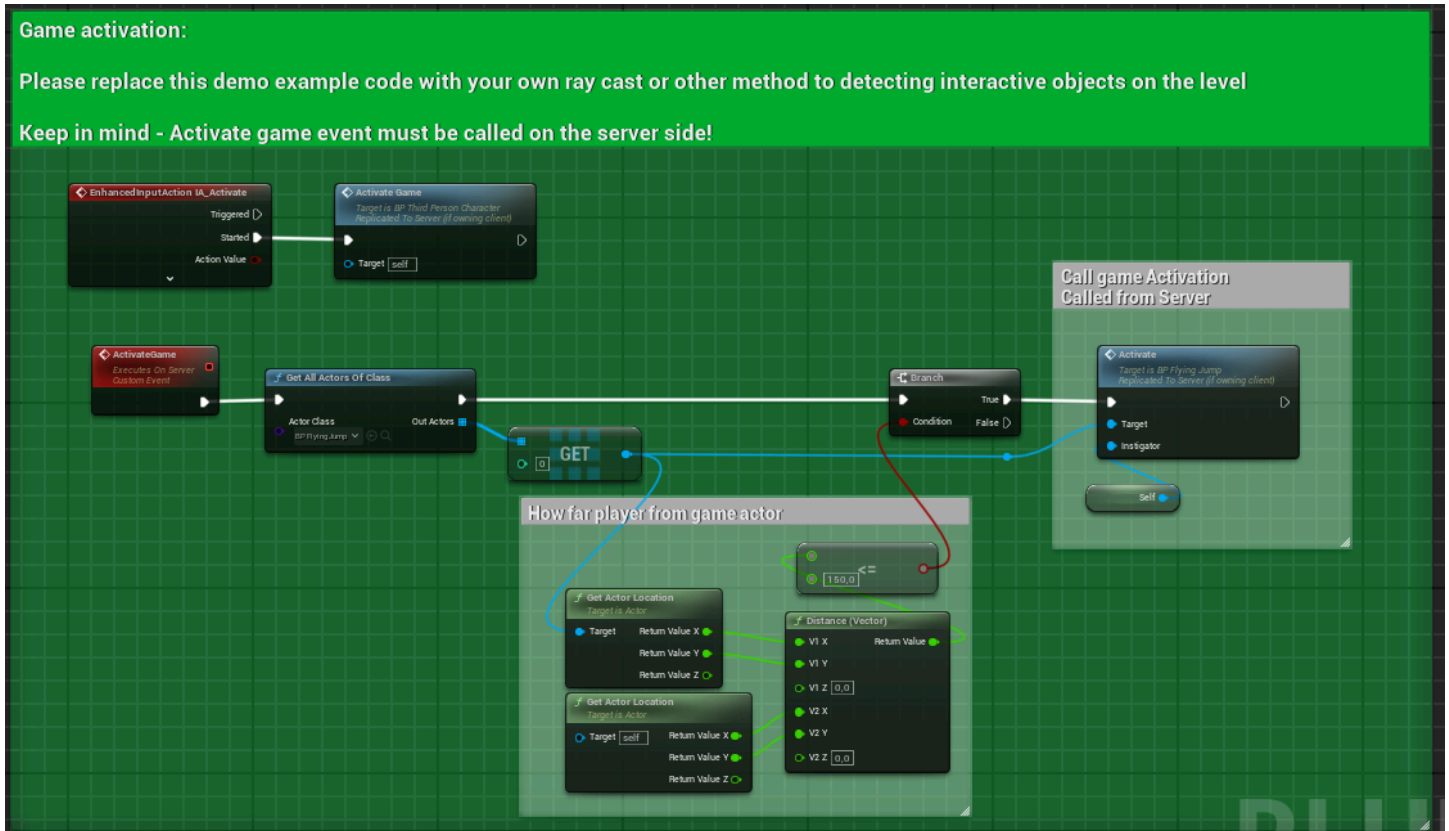
Cheat mode: Cheat mode - unlimited lives

Background speed: Background horizontal movement speed

Score, Restart, Exit and Restart texts: Text for different game stages

Game Activation

Example of activation code you can find inside
Content/FlyingJump/Demo/Blueprints/BP_ThirdPersonCharacter.uasset



This is a simple example of activating the Flying Jump, using very basic code for demonstration purposes.

All it does is get a reference to a copy of the Flying Jump in the level (using GetAllActorsOfClass), and if the player is within 150 cm, it triggers the Activate server event on the Flying Jump side.

You should implement (or use) your own logic to determine whether the Flying Jump can be activated - for example, by performing a raycast or using another method to check if activation is possible for this object in the level. You might consider adding your own interface to the Flying Jump with two methods: `Activate(server)` and `CanBeActivated?`

It is not recommended to use the example activation code in your actual project.

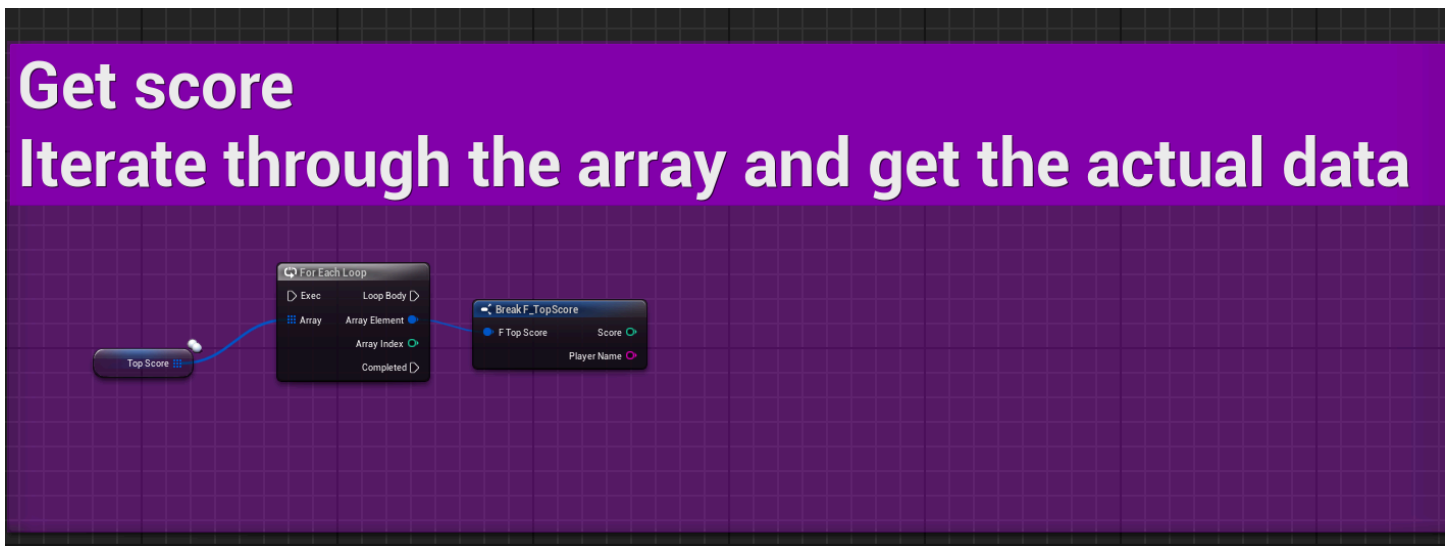
Leaderboard

The game features a leaderboard that displays the top 3 highest scores. When the game ends - Game over or Exit from game, the player's score is evaluated. If the score is higher than the current lowest top score, it is added to the leaderboard. The leaderboard is then sorted from highest to lowest.

If the score is too low to make the top 3, it is not added.

Read current leader board:

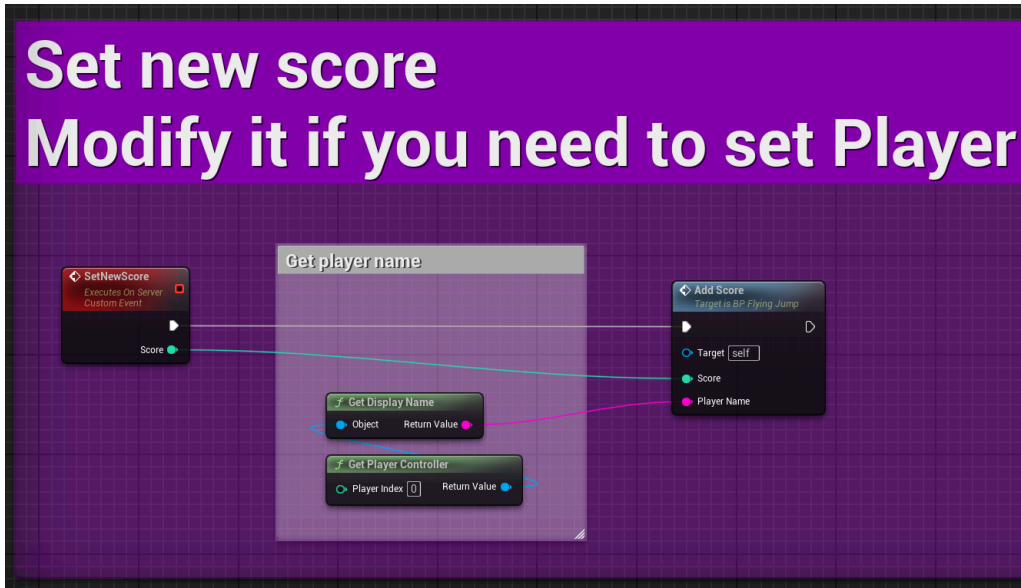
Inside of Content/FlyingJump//Blueprints/BP_FlyingJump.uasset you can find predefined code:



You just need to iterate through the array and get score data. Then save the inside of your project e.g. game save data or server data, etc...

Set initial leader board values:

Inside of BP_FlyingJump you can find predefined code:



The **AddScore** node you must use to add score into the leaderboard table. Keep in mind AddScore must be called on the server. Only 3 scores are supported, no matter in what order you add high scores, they will be sorted in descending order.

Inputs:

Score - integer

Player name - string. Max is 22 characters length is supported

Artstyle modification

Under the Content/FlyingJump/Textures folder you can find textures for obstacles, Player, start and game over screens:

T_FlyingJump_obstacle01, 02, 03, 04 and 05 are used for obstacles

T_FlyingJump_player - player (UFO) texture

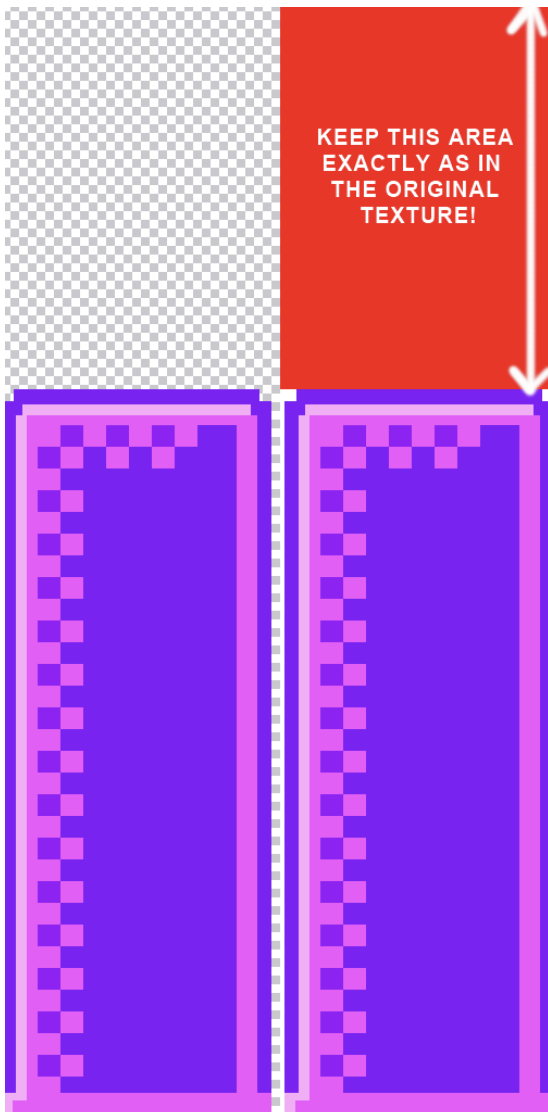
T_FlyingJump - start screen texture

T_FlyingJump_bg - horizontal seamless background texture

T_FlyingJump_gameover - game over screen

Please don't change the proportion of textures! But you free to change resolution

If you want change obstacle texture please keep original visual size of obstacle:



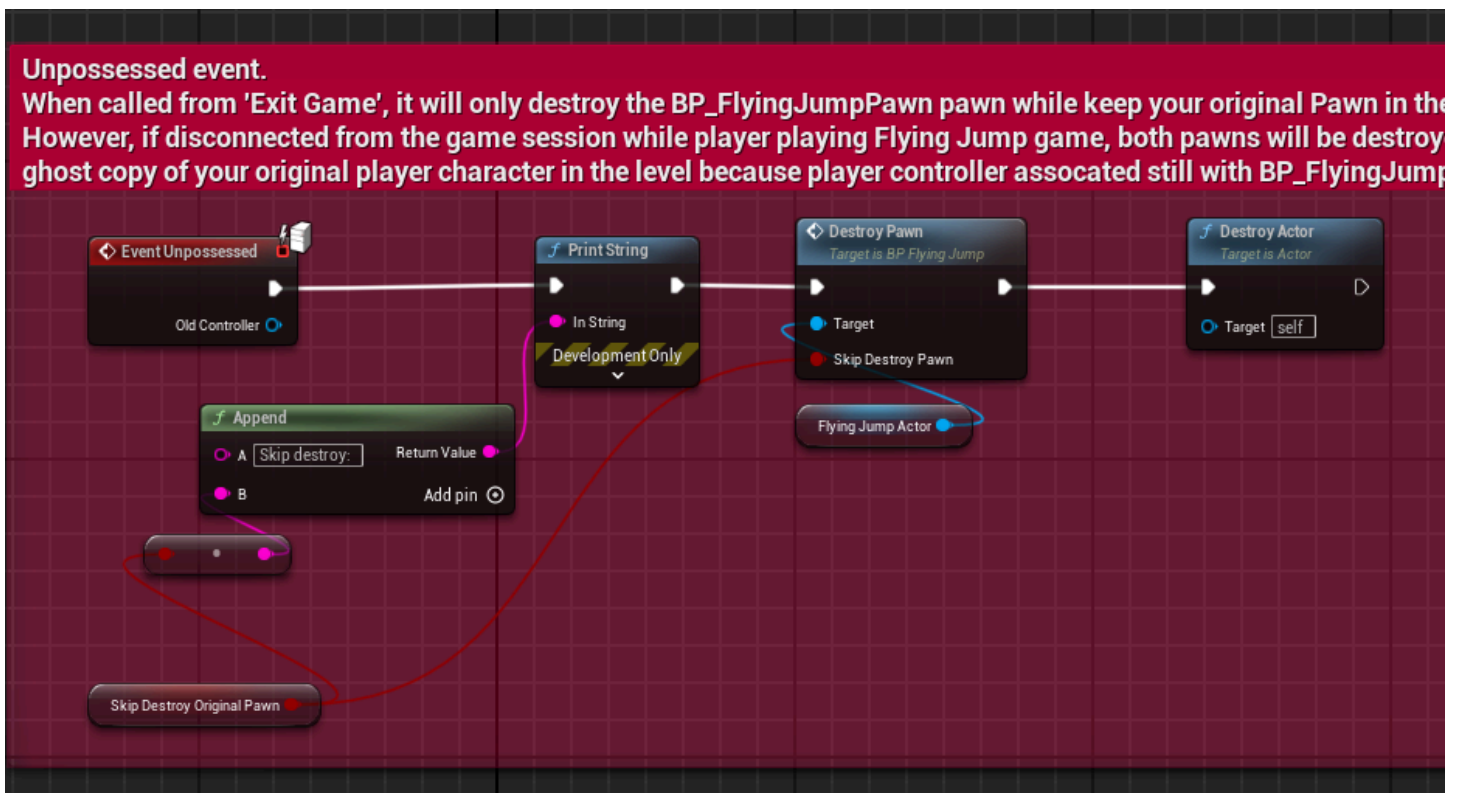
BP_FlyingJumpPawn

BP_FlyingJumpPawn will be spawned when the Flying Jump game is activated and will be destroyed after exit from the game.

Unpossessed event behavior

When normal exiting the game, this event only destroys the BP_FlyingJumpPawn while preserving your original Pawn in the level. However, if a player disconnects during gameplay, both pawns are destroyed to prevent leaving behind a ghost character, since the player controller remains linked to the BP_FlyingJumpPawn

The Unpossessed event code you can find inside of Content/FlyingJump/Blueprints/Core/BP_FlyingJumpPawn.uasset



Input

All game inputs - start, jump, exit and restart you can find here:
FlyingJump/Content/Inputs/IMC_FlyingJump.uasset

Default is:

Key E - start / exit from game

Key R - restart game

Space bar - jump

THANK YOU!