

Coin Pusher Complete Game Kit

-

User Guide

V1.0

Table of contents

| | |
|---|------------------|
| <i>The Game Manager</i> | <i>3</i> |
| I. Set Up the Machine | 3 |
| II. Set Up the User Interface..... | 6 |
| III. Set Up the Physic..... | 9 |
| IV. Set Up Specials coins..... | 11 |
| V. Set Up the Player..... | 13 |
| <i>The Coin Manager</i> | <i>15</i> |
| Set up a coin..... | 15 |
| <i>The Prizes Manager</i> | <i>18</i> |
| Set up a prize..... | 18 |
| <i>The Regen Manager.....</i> | <i>19</i> |
| <i>The Save Manager</i> | <i>20</i> |
| <i>Some Useful methods</i> | <i>21</i> |
| I. GameManagerScript :..... | 21 |
| II. CoinsManagerScript | 22 |
| III. PrizeManagerScript | 22 |

The Game Manager

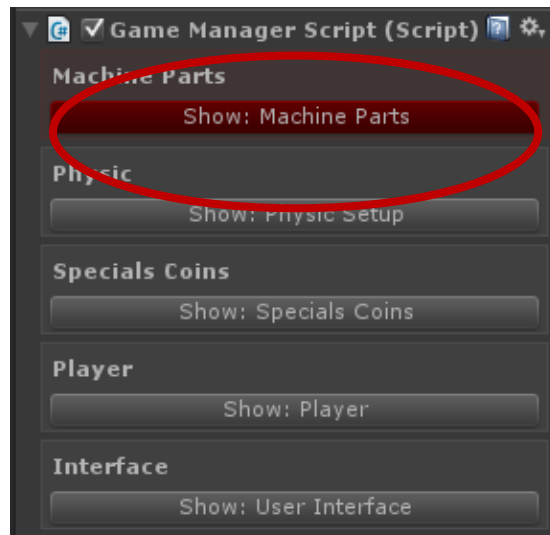
I. Set Up the Machine

1) You must have the following gameobjects in your scene :

- A pusher
- A ground (a floor, a carroussel, i don't no the real name)
- 2 movables side walls (walls must be only ONE gameobject. See example scene)
- Décorations (like back wall)

2) Drop a GameController prefab.

Click On « Show : Machine Parts »



We must now fill in the blanks.



3) Fill in the blanks

Ground :

Drag your « ground » in the correct field.

Walls :

The side walls must be unit in only one gameobject (like an empty parent).

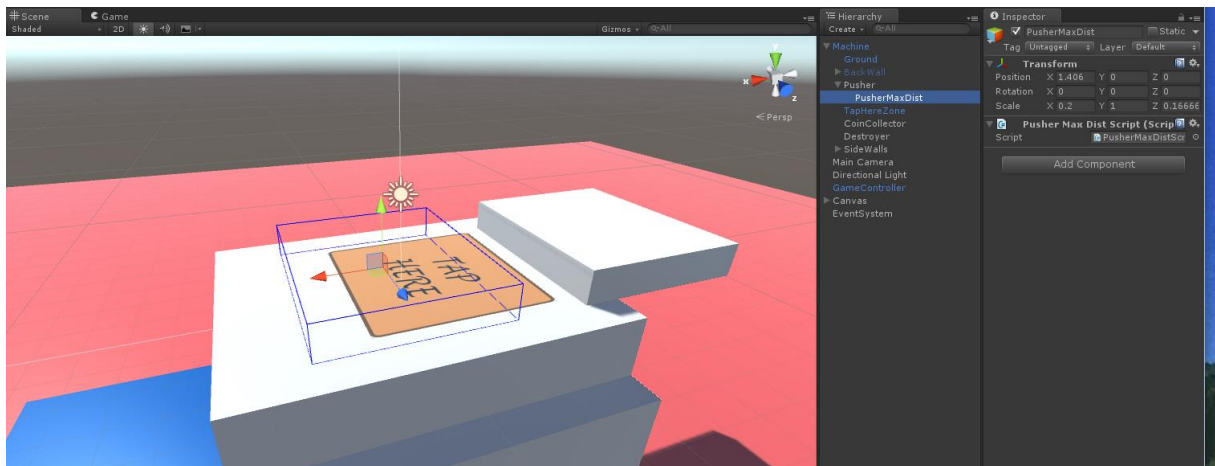
A script is automatically add to this parent. The script will manage the movement of the walls and audio.

Pusher :

When you drop your pusher in the « pusher gameobject » field, 2 things happens :

- A script is added to your pusher.
- A « PusherMaxDist » gameobject is attach to your pusher (as child). This new gameobject will appear in « scene view » in blue wires, but it never appear in « Game view ».

You must select the your Pusher gameobject, and place it in his starting position. Then select the « PusherMaxDist » gameobject and move it to the final position. Usually, the pusher move only in ONE axis.



TapHereZone:

Create a plane ('GameObject -> 3D Object -> Plane'). This gameobject don't need a renderer (*optional*)

When you drop it, a script is add to the gamobject. This script manage collisions. This plane will be move in a new layer (at runtime) in order to never collid with an other gameobject. This « tapHereZone » will just react when it was touched (or clicked) by the player. So you don't have to care about his positioning with regard to ground or pusher.

CoinCollector :

This is the machine part, directly in front of the player, which collect the coin and prize. (*Blue part in the example scene*)

Create a plane ('GameObject -> 3D Object -> Plane'). This gameobject don't need a renderer (*optional*)

When you drop it, a script is add to the gamobject. This script manage collisons. Thanks to this Destroyer, Coin and Prize script can play a sound(optional), and detroy themself when they are touch this CoinCollector.

As this CoinCollector detect collsions for winning coins and prizes, you can add your own methodes in that « *CoinCollectorScript* ».

Destroyer :

This is the machine part, directly under the machine, which collect all gameobjects fallen on the side of the machine (*Red part in the example scene*)

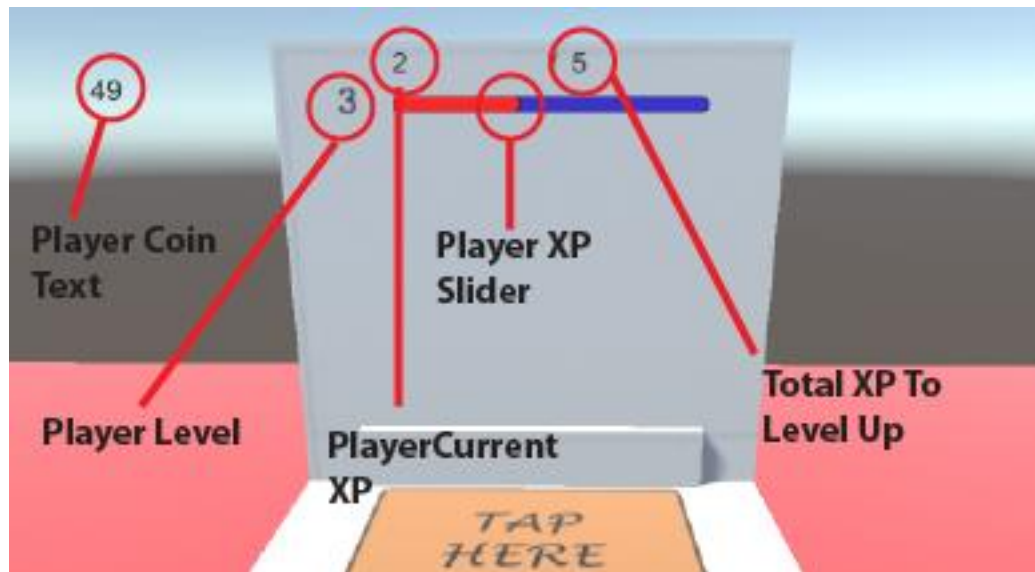
Create a plane ('GameObject -> 3D Object -> Plane'). This gameobject don't need a renderer (*optional*)

When you drop it, a script is add to the gamobject. This script manage collisions. Thanks to this Destroyer, Coin and Prize script can play a sound(optional), and detroy themself when they are loose on the side a the machine.

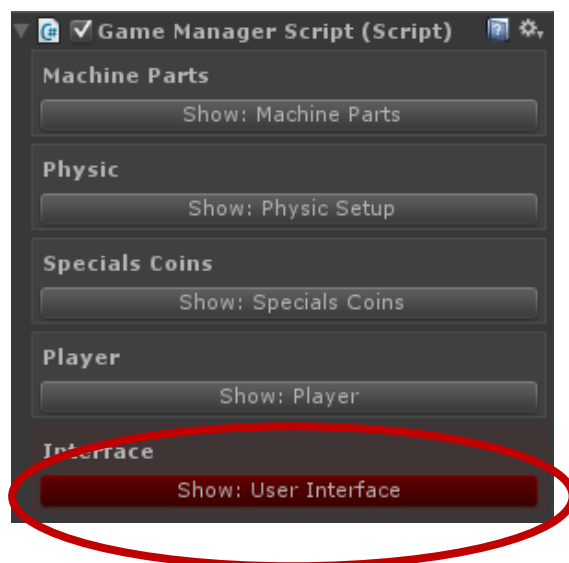
The machine is fully set up.

NB : all parts of your machine (except the « TapHere ») must have a collider (non trigger). The pusherand walls must have a rigidbody.

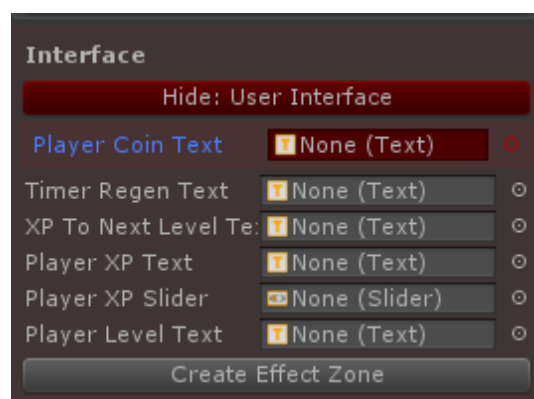
II. Set Up the User Interface



Click On « Show : User Interface »



We must now fill in the blanks.



1) Fill in the required blank

Player Coin Text :

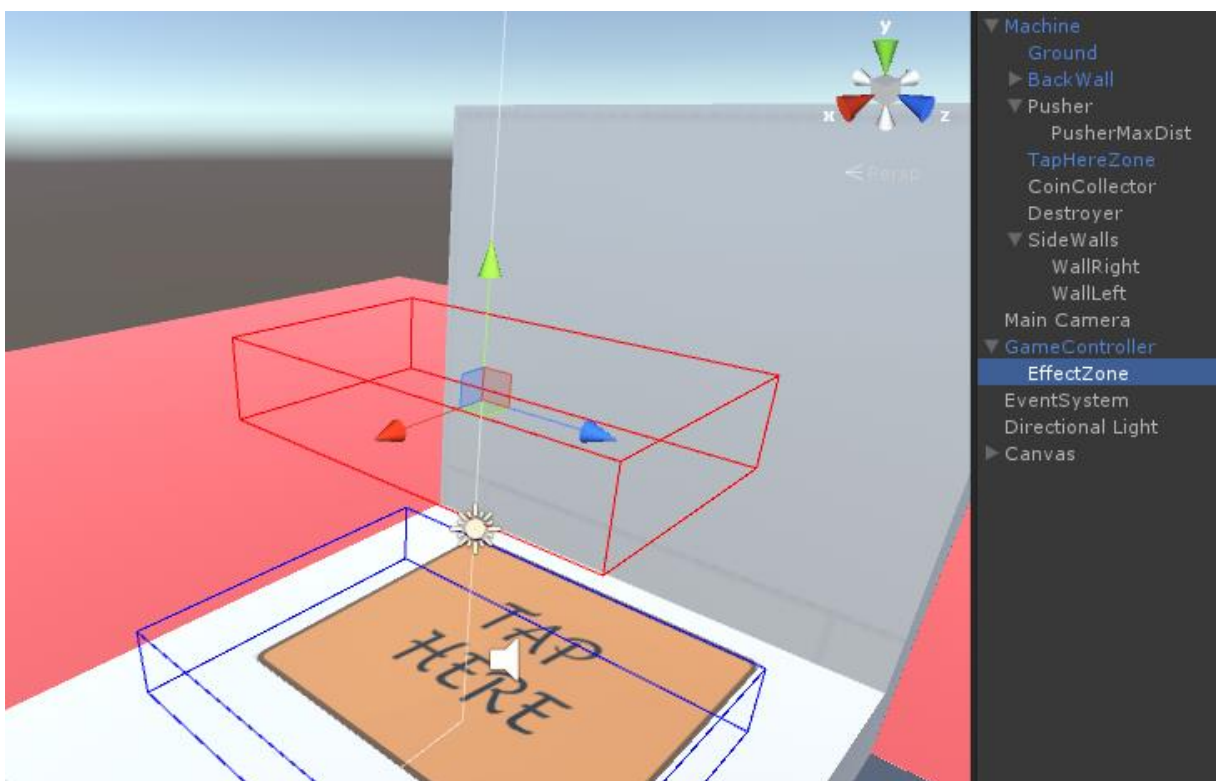
You need to drop a « Text uGui ». (*GameObject* -> *UI* -> *Text*)

The player' coins will be displayed in this Text.

Effect Zone :

Click on the « Create Effect zone » button.

This will create a gameobject only visible in the SceneView (same as *pusherMaxDist*). This new gameobject is a child of *GameController* and is named *EffectZone*.



It appear in red in the SceneView. When a coin (or a prize) is collect or destroy, an Effect in create (if setup on the gameobject). This effect will be spwan andomly in the « EffectZone » Area.

All effects call with the method (in *GameManagerScript*):

```
public void CreateEffect(GameObject effectGO) ;
```

with be spwan in this area.

2) Fill in the optional blank

Timer Regen Text:

A simple countDown. When timer reach « 0 », player get a coin (if needed).

XP To Next Level Text :

The Total XP needed to levelUp.

Player XP Text :

The player current XP.

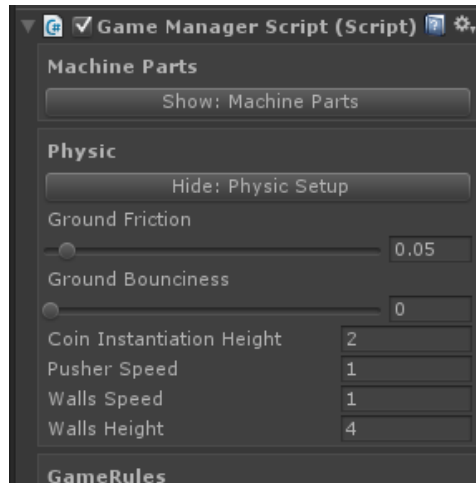
Player XP Slider:

A uGui Slider (*GameObject -> UI -> Slider*)

A visual representation of (Player current XP) / (Xp To Next Level).

III. Set Up the Physic

You can change machine physic rules to adapt the game to your vision.



Ground Friction :

A value between 0 and 1. A value of zero feels like ice, a value of 1 will make it come to rest very quickly unless a lot of force or gravity pushes the object.

In fact, the « ground friction » value is applied to the pusher too.

Ground Bounciness:

A value between 0 and 1. How bouncy is the surface? A value of 0 will not bounce. A value of 1 will bounce without any loss of energy.

In fact, the « ground bounciness » value is applied to the pusher too.

This 2 parameters, with (coin and prize mass) change considerably the coin and prize movement and game difficulty. It's this parameters which makes an excellent pusher game or common pusher game. So take your time to select the perfect values !

Coin Instantiate Height :

The height of the spawn point for coins over the « TapHereZone ».

This height will be double for Prizes.

Pusher Speed :

At speed 1, pusher do a loop in 6 seconds.

At speed 2, pusher do a loop in 3 seconds

Walls Speed :

At speed 1, walls are completely up in 3 seconds

At speed 2, walls are completely up in 1.5 seconds...

Walls Height :

When walls are down, the distance required to reach the UP position.

IV. Set Up Specials coins



Wall Up Duration :

How long the walls stand up ?

Joker Coin Number :

How many coins will be transform ?

Big Coin GameObject :

When the coin « CoinBigToken » is collected, a giant coin is spawn . This coin fall on the ground and blow up all the others coins.

You must drop here the GiantCoin, not the coinBigToken.

Big Coin Explosion Power :

The power of the blow up ;

Big Coin Explosion Radius :

Imagine a sphere at the contact point between GiantCoin and ground. All gameobjects in this sphere will be blow up.

Multipl XP Duration :

For the « CoinDoubleXp », how long the XP win will be multiply.

Multipl Value Duration :

For the « CoinDoubleValue », how long the coin 's value will be multiply.

Coin Expander Multiplier :

For the « CoinExpand ». When this coin is collected, all coins on the ground will be scale up. Choose the multiplier here.

Shower Coin Number:

For the « CoinShower». When this coin is collected, XX coins will be instantiate randomly over the tapHereZone. Choose XX here.

Shower Coin Delay:

The delay between 2 coins instantiation ;

V. Set Up the Player

| Player | |
|------------------------------|------|
| Hide: Player | |
| Starting Coin | 5000 |
| Tap Action Regen Time | 1 |
| Tap Action Limit | 4 |
| Regen Offline Speed | 1 |
| Regen Offline Max Coin | 5000 |
| Regen Online Speed | 10 |
| Regen Online Max Coin | 60 |
| Manage XP : | Yes |
| XP per Level | |

Starting Coin :

When the game start, how many coin the player have ?

Tap Action Regen :

If tap action limit reach zero, how many time the player must wait before getting a new « tap action ».

Tap Action Limit :

How many times the player can tap very quickly.

Regen Offline Speed :

How many seconds are needed to get One coin when app is not running.

Regen Offline Max Coin :

Maximum number of coins that will be regenerate durring offline.

Regen Online Speed :

How many seconds are needed to get One coin when app is running.

Regen Online Max Coin :

Maximum number of coins that will be regenerate durring online.

Manage Xp (YES or NO) :

The GameController must manage the XP for you ?

If you select YES, you must fill in at least ONE level in the « Xp Per Level » array.

If you select NO, the XP will not be manage at all (except if you write your own script).

Xp per Level:

NB : player start at level 1.

- Size : set the max level.
- Element 0 : always 0 !!
- Element 1 : how many XP to get level 2
- Element 2 : how many XP to get level 3
- ...

If you want do something when player level up , like create a gameEffect, play a sound, give a special coin or anything else, you can write your code in the `PlayerLevelUp()` methode (in GameManagerScript -> 'Level XP Methodes' section).

The Coin Manager

Set up a coin

1) Create a coin :

All your coins must have :

- A collider (non-trigger) : to detect collisions
- A rigidbody : the coin must fall. It's important to set up the mass with precision.
- A « CoinStandardScript » in order to manage XP, Value, sounds and effects for this coin.
- Your standard coin (the base coin, classic coin, cormal coin,..) must be tagged as « CoinStandard »

When a coin is perfectly setup, make a prefab of it

(simply drag a GameObject that you've created in the scene into the Project View)

2) Set up the CoinManager :



Click on your GameController (in the Hierarchy Window).

In order to create a new coin, you must increase the Size value by 1.

Coin Name : Write a name in the « Coin Name » empty field

Coin Go : Drop your coin gameobject empty field

Coin Weight : the drop rate probability for this coin. (it's not percent just a weight !)

3) Special coin :

In order to create special coins, the simplest way is :

- tag your special coin (ex : « coinWall »)
- write in the « CoinCollectorScript »

```
if (collision.gameObject.tag == "CoinWall")
{
    gameManagerScript.MoveWalls();
}
```

- write in the « GameManagerScript » (at the end of the script)

```
public void MoveWalls()
{
    walls.GetComponent<WallsScript>().MoveWalls(wallDuration,
                                                wallSpeed);
}
```

In this way, your script will always be clear, and your « special capabilities » will always be accessible. Of course, do as you want and only if it makes sense (*ex : CoinBigToken function isn't call in GameManagerScript but directly in the coinScript.*)

For more information about the Special coins included, See below.

CoinWall :

When this coin touch the coinCollector, sideWalls move up. Coins and prizes on the ground can't fall.

CoinPrize :

When this coin touch the coinCollector, A « Prize » is spawn.

CoinJoker:

When this coin touch the coinCollector, XX standards coins are transformed in Random Special Coin.

CoinBigToken :

When this coin touch the coinCollector, A « BigCoin » is spawn. When this « BigCoin » touch the ground, a « blow » is create. All coins are affected.

CoinDoubleValue :

When this coin touch the coinCollector, for some time, all coin collected give two coins.

CoinDoubleXP:

When this coin touch the coinCollector, for some time, all coin collected give more XP.

CoinExpand :

For the « CoinExpand ». When this coin is collected, all coins on the ground will be scale up.

CoinShower :

For the « CoinShower». When this coin is collected, XX coins will be instantiate randomly over the tapHereZone.

The Prizes Manager

Set up a prize

1) Create a Prize :

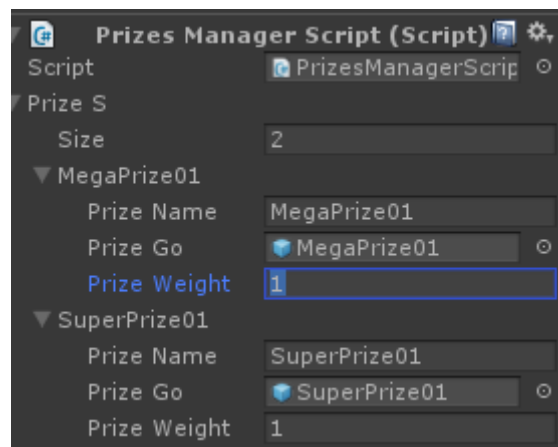
All your Prizes must have :

- A collider (non-trigger) : to detect collisions
- A rigidbody : the prize must fall.
- A « PrizeBasicScript » in order to manage XP, Value, sounds and effects for this coin.

When a prize is perfectly setup, make a prefab of it

(simply drag a GameObject that you've created in the scene into the Project View)

2) Set up the PrizesManager :



Click on your GameController (in the Hierarchy Window).

In order to create a new prize, you must increase the Size value by 1.

Prize Name : Write a name in the « Prize Name » empty field

Prize Go : Drop your prize gameobject empty field

Prize Weight : the drop rate probability for this prize. (it's not percent just a weight !)

The Regen Manager

This Manager is self-ruling. Values can be setup in the GameManager -> GameRules.

NB : If you don't want to use this feature, you simply have to set the value to « zero ».

How works the « Offline coin Regeneration » ?

When player suspend the game (and go in background for mobile), the current date and time are save in a playerPrefs.

When player start a new game, the new date and time are compare to the old and coins are calculated.

The Date and Time use are the Mobile system date and time.

The Save Manager

This manager handle with load and save multiple values

PlayerCoins, PlayerXp, and all Coins and Prizes on the ground are automatically save when game is pause or loosefocus (go to the background).

When the game start, this values are automatically load by the GameManagerScript.

If you need to save calue (ex : if scene must change), you just have to call

SaveGame() from GameMaangerScript. (same for LoadGame()).

Some Useful methods

I. GameManagerScript :

LoadGame() : load Coins and Prizes on the ground. Load PlayerCoins and PlayerXP.

SaveGame() : save Coins and Prizes on the ground. Load PlayerCoins and PlayerXP.

RandPosInAGameObject(*gameobject go*) : return a Vector3. This point is located inside the Gameobject go. Use for instantiate in the EffectZone or TapHereZone.

FindAllPrizesOnTheGround() : return an Array of GameObject. Look for All GameObject which is manage by « PrizeManager ».

FindAllCoinsOnTheGround() : return an Array of GameObject. Look for All GameObject which is manage by « CoinManager ».

FindAllCoinsAndPrizesOnTheGround() : return an Array of GameObject. Look for All GameObject which is manage by « PrizeManager » and « Coinmanager ».

SetPlayerActualCoins(*int coinsToAdd, (option) bool resetPreviousValue*) : Set the number of coins for the player. If *resetPreviousValue* is false, coin will be added. If true, coinToAdd will be the actual playerCoin

GetPlayerActualCoins() : return an integer. Number of coins for the player.

SetplayerXP(*int xpToAdd, (option) bool resetPreviousXp*) : Set the Xp for the player. If *resetPreviousValue* is false, xp will be added. If true, xpToAdd will be the actual player XP

GetplayerXP() : return an integer. Xp for the player.

Checklevel() : refresh the Interface and recalculate the playerLevel. If player levelUp, PlayerLevelUp() will be fired.

CreateEffect(*GameObject effectGo*) : Instantiate the effectGo in the « effectZone ».

NB : You can find most of the SpecialCoin method in this GameManagerScript.

All variables in GameManagerEditor are Publics, you can access to them by script. So you can modify according to a Prize'collection, or anything else.

II. CoinsManagerScript

GetOneCoin() : return a random Coin.

GetOneSpecialCoin() : return a random SpecialCoin.

III. PrizeManagerScript

GetOnePrize() : retrun a random Prize.

Thank for reading.

If you need help, you can contact me at

jocelyn.assets@gmail.com