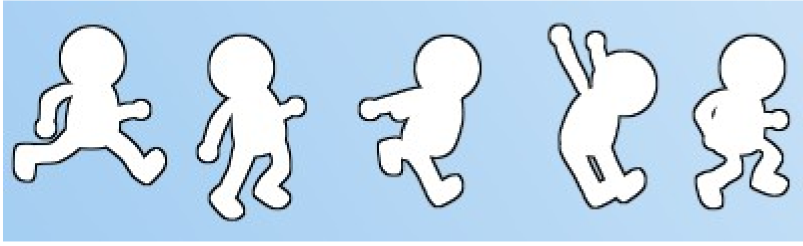
2D Jump'n'Run Framework

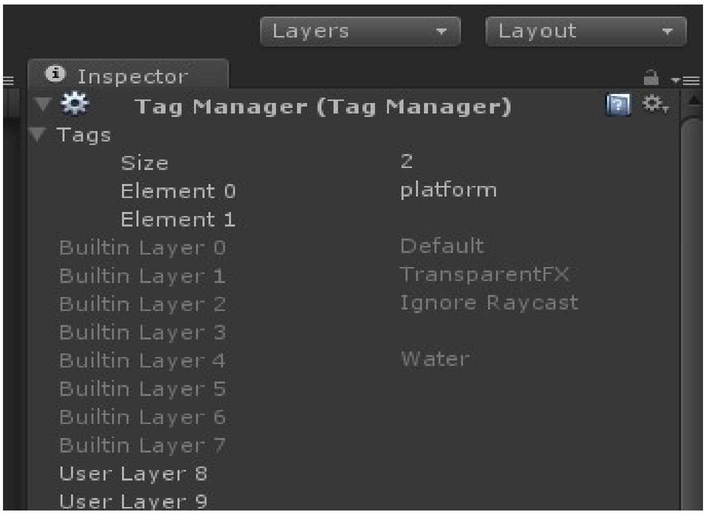


Hello and thank you for buying this Framework! Since this is an early version, please report any bugs or feature requests in the Unity Forum Thread for this Framework!
I hope you have fun and can develop the jump'n'run game of your dreams!

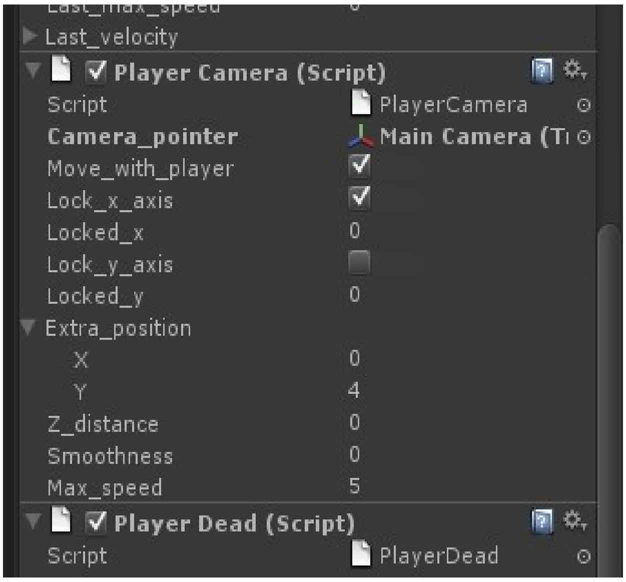
Index

Your first player!

Since you're reading this, you should already have imported the package!
So let's start by adding the needed tags:
Select any object (like „Main Camera“ and click on „Untagged“ and then „Add Tag“.
At the top of the list, there is the word „Tag“, click on it and add, if not there, the tags „plattform“,“climbable“,“climbgotrough“,“Enemy“,“Projectile“.
It should look like this



Tag Manager (Tag Manager)
Tags
Size 2
Element 0 platform
Element 1
Builtin Layer 0 Default
Builtin Layer 1 TransparentFX
Builtin Layer 2 Ignore Raycast
Builtin Layer 3
Builtin Layer 4 Water
Builtin Layer 5
Builtin Layer 6
Builtin Layer 7
User Layer 8
User Layer 9



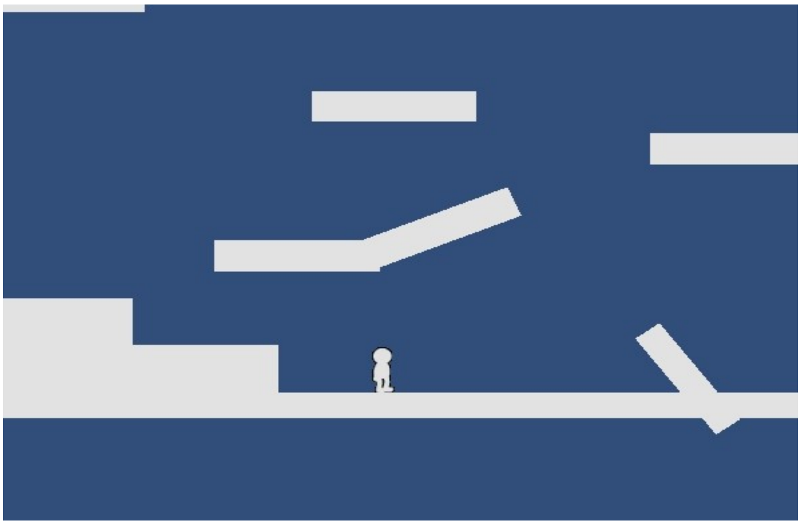
Last_max_speed 0
Last_velocity
☑ Player Camera (Script)
Script PlayerCamera
Camera_pointer Main Camera (Tı
Move_with_player ☑
Lock_x_axis ☑
Locked_x 0
Lock_y_axis □
Locked_y 0
Extra_position
X 0
Y 4
Z_distance 0
Smoothness 0
Max_speed 5
☑ Player Dead (Script)
Script PlayerDead

After that, in the project windows, find the prefab folder and add the 2DPlayer prefab to the scene and set his Z position to 0.Click on the 2Dplayer in the scene and look for the „Player Camera“ script in the object inspector.

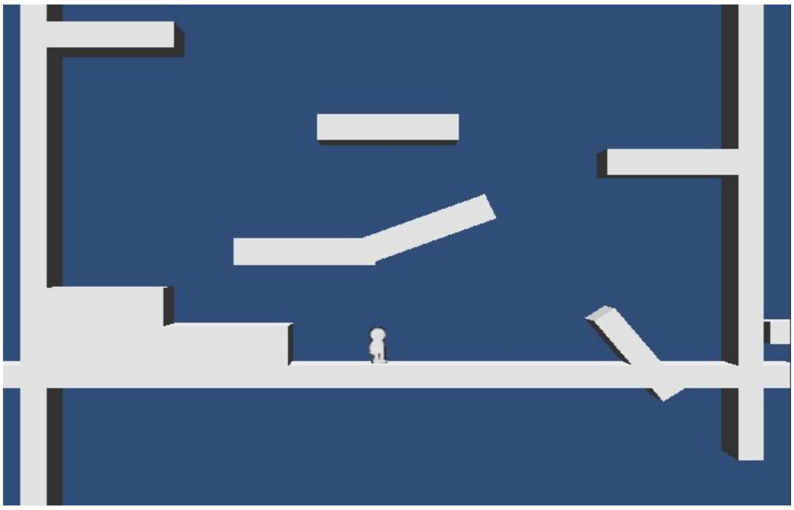
Now assign the Main Camera of the scene to the „camera_pointer“, like this:

Now, click on the Main Camera in your scene. You have to decide if you want a 2D or 2.5D look. For 2D, set the „Projection“ setting to „Orthographic“ and for 2.5D „Perspective“.

2D



2.5D



After that, go to Edit → Project Settings → Input and edit the buttons there!

Note: If you're using 2D you have to set the size of the orthographic view.
Moving on the Z axis will do nothing!
You should now be ready to test your level!

Animation2D

The Animation2D script is your system for all your character animations.

To add a new animation, resize the „Animations“ array by +1 and set the variables:

Name – The name of your animation
Graphics – Array of your animation frames
Speed – Playback speed
Loop – If true, the animation will loop.
Play after – Enter a clip name that will play after this clip is finished

Variables of Animation2D

Start Animation – This animation will be playing at start (useful for deko objects)
Renderer_Pointer – The pointer to the projection plane.
Current_Animation – The ID of the Animation that is playing right now.

Animation3D

Animation3D is mostly the same, but you only got 4 variables:

Name - Name of the clip
Clipname – The animation clip name of the animation.
Play_After – plays another animation, after the current one is played.
blend_animation – Should the animation clip be blended in.

I don't have a 3D character and used lerpz or simple testing. Please see if it works and contact me if you have any problems!

Player Movement

This script contains all the important bits for a smooth movement.

Variables of Player Movement:

- player_graphic_pointer - The pointer to the player graphic.
- Plus_Rotate_Angle – To fix rotation of a wrong facing 3D character.
- Player_can_move – if untrue, no input will be taken.
- Hold_button_to_run - need to press a button to get into the run state
- Max_walking_speed – maximal speed of walking. After that it's running.
- Max_running_speed – maximal speed of running.
- Gain_move_speed – How fast the player accelerated
- Lose_move_speed – How fast the player comes to a stop
- Stopping_lose_move_speed – How fast the player stops, if we go in another direction
- X_axis_name - Input name for the x axis. Edit this for a multiplayer!
- Run_button_name – The button we need to press so we can run.
- Current_Speed – The calculated speed we're moving
- Current_Mode – The movement mode we're in.
- Movement_direction - 1 = right; -1 = left;
- Last_velocity – The velocity of the last fixed update
- BoostGraphic – This object will be visisble as long as we boost
- Can_extra_boost - If true we can use the boost
- Time_till_boost - How long we need to run before switch to boosting
- Max_boost_speed – how fast we are in boost mode
- Animation_[...] - Name of the animation clip.
- Use_mobile_input - If true, only mobile Input will work
- Mobile_joystick – The virtual Joystick
- Mobile_run – The run button (joystick)

Player Jumping

Everything jumping related!

Variables of Player Jumping:

Jump_button_name – The name of the button so we can jump
Air_control – How much control we have in the air
Rotate_in_air – if true, the player will turn left and right in the air
Jump_strength – How high we can jump
Max_jump_speed – How fast we can get in the air
Multi_jumps – How many times we can jump in the air.
Jump_force – Extra force in air. Useful so jumps are less „floaty“
Player_movement – Player Movement component. Will be set automatic
Animation_component – Will be set automatic
Wall Left/Right – True, if there is a wall on one side.
Last_velocity – The velocity of the last update
Ignore_layer – Objects we can walk trough (trigger, etc.) will be ignored
Can_glide – can we use gliding?
Glide_Air_Control – How good we can change the direction in the air
Glide_max_speed – how fast we can get while gliding
Glide_force – To slow us down or add any other force like wind
Glide_button_name – The name of the button you entered in „Input“
Glide_Max_Fallspeed – Set this to slow the falling down
Glide_only_after_doublejump – Useful, if gliding and jumping is the same button.
Can_Walljump – Can we walljump?
Min_speed_for_walljump – We need to be this fast to walljump
Jumpattack_damage – Damage an enemy will take trough jumping

Walljump_speed – The speed after we walljumped
Walljump_up_power – The upforce to jump higher after a walljump
Walljump_reset_multijump – Can we multijump after each walljump?
Time_for_Walljump – Time after hitting the wall to press jump again
Animation_[...] - Animation clip name
Mobile_jump – The virtual jump button (joystick)
Mobile_glide – The virtual glide button (joystick)

Player Dead

This is mostly empty for now. Only good for respawning.

Variables of Player Dead:

Last_respawn – The last respawn point we triggered.
Reset_Camera_x – resets the camera after respawn
Reset_Camera_y – resets the camera after respawn

Player Health

Controls the health and lifes of the player

Max Health – Maximum the player can get
Health – Current Health
Max lifes – Maximum lifes the player can get
lifes - current lifes
recover_time – time the player can't get hurt after a attack
transparent_while_recover – makes the player_plane transparent
passable_on_death – the player will freeze in place and ignores collision
animation_dead – name of the dieing animation.
Circle_fadeout – use the fadeout effect?
wait_for_respawn – wait before the circle appears
wait_for_fade – time before we respawn when we start the fadeout

Player Camera

Everything camera related.

Variables of Player Camera:

Camera_pointer – Pointer of the camera for this player. Edit this for multiplayer.
Move_with_player – The only option right now. Moves with the player.
Lock_x_axis – Lock x Movement
Locked_x – Position where the camera is locked
Lock_y_axis – Lock y Movement
Locked_y – Position where the camera is locked
Extra_position – sometimes usefull for a better overview for the player
Z_distance – How far is the camera away. **Note: Minus values for 2.5D camera, plus values for 2D cameras!**
Smoothness – how smooth the camera should be
max_speed – how fast the camera can get

Coming Soon: Multiplayer related stuff, more movement options

Player Climbing

Use TriggerClimbing to set areas where the player can climb. Also, every object with the tag „climbgotrough“ will be passable for the player while climbing. This is useful to emulate the classic 2D feeling of climbing to a new floor of a building etc. Think of donkey kong arcade, where mario can climb trough the platforms but also run over it, while he's not climbing.

Variables of Player Climbing

Can_climb – You don't need to set this, true if player can climb
climb_object – You don't need to set this, the current trigger
climbing – You don't need to set this, are you climbing right now
Climb_speed – How fast you can move while climbing
Locked_x – Is setted by triggerclimbing
Jump_while_climbing – can the player jump while climbing?
Animation_[...] - Name of the animation clip.
Unclimb – You don't need to set this.

PlayerAttack

Let's attack somebody!

Attack_button – Name of the button to attack
Spawn_attack – the prefab we will spawn
spawn_distance – how far away from the player
ignore_player_collision – the attack prefab won't interact with the player
max_shots_in_scene – max amout of shots at the same time. 0 = unlimited
mobile_attack – The mobile input

Helper

Helper are components that will help you by creating your worlds.

HelperRotate – Rotates an Object, useful for a group of platforms
HelperPlatform – Helps you to move your platforms. Also, locks the rotation to 0 and sets the Tag
HelperJumpad – If only_on_top is true then you can only use it from above.
HelperFadeout – A nice Circle Fadeout effect.
HelperProjectile – Stores the owner of the projectile.

Trigger

Trigger will trigger something! NO WAY! They can do many stuff and will make your world more life like!

TriggerCamera – Edits the camera settings of the player. **Note: If you're using a 2.5 camera, use negative values for z_distance!**
TriggerDeadZone – If the player touches this zone he will die (the character! Not the player that play the game!)
TriggerRespawner – If a player walks trough this trigger, he will respawn there
TriggerClimbing – Set areas where you can climb! You can lock the X-Axis.
TriggerGoal – The level end!
TriggerAiTurnaround – Turns an Ai around, so it wont pass.
TriggerAttack – Will hurt the player. Use this for Aimovement
TriggerPickUpItem – Multi-Script for all pickupable things

Coming Soon: Traps, collectable Items and Powerups, SwimmingZones

AiMovement

The base Ai class for a very stupid but effective enemy.

current_mode – what is the ai doing?
ai_graphic_pointer – same as player plane, the 2d/3D object
Ai_can_move – can the ai move?
Walk_speed – just that
run_speed – just that
random_stop_chance – 0.0 to 1.0 chance that it will stop walking
random_stop_length – stops inbetween those two seconds
sight distance – how far can the enemy see?
sight_interval – use the trace every x frames
sight_interval_count – adjust this for mobile platforms. Make it that there won't be all traces for all enemies at the same time!
Direction – the current and start direction. -1 and 1
attack_target – the current target
turn_to_target – target is behind it? Turn around!
turn_to_target_interval – check after this frames if the target moved
run_torwards_target – runs to the target after sight
play_animation_at_sight – just that
lose_target_at_distance – we won't chase the player forever!
attack_target_at_distance – near this to attack
wait_till_attack – attack after this time, as soon as the animation plays
attack_impact – how much the player jumps at an attack
delay_attack – ajust this so the enemy won't attack mindless
attack_damage – just that
attack_collider – prefab of the trigger collider for the attack
slow down near target – use this for fast enemies!
slow_down_distance – near this to slow down
slow_down_by – movement_speed / slow_down_by = new speed
ai_health – current health
ai_maxhealth – max health the enemy got
disappear_after_death – just that
disappear_after_seconds – how long does it take
recover_time – you can't mindless attack the enemy!
spawn_items – Fill this array and these items will spawn!

The attack collider is a collider with the TriggerAttack script. Create for example a box and move it a bit to the right (x = 1 or something) and adjust the hight and thickness. When the enemy attacks, it will spawn this collider and the player will get hit, if he is inside the trigger.

Attacks

All the wonderful attacks!

AttackFireball – Will bounce around – kinda Mario-Style.

RageSpline/CheatMaster

RageSpline: For all RageSpline User, there is a testscene for you to test a RageSpline terrain with your player. If you have any problems, please play around with the vertex count of the meshcollider, the more the easier for the player to move over the object. If you have any problems, please contact me!

CheatMaster: If you use CheatMaster, add the cheatMasterAddon.js script to the same object that got the cheatMaster.js script. You should now be able to use the commands „up“ „down“ „bigger“ „smaller“ „nogravity“ and „gravity“. I will add ne cheatwords to this file over time!

Mobile functions

The code of this framework is written to work on mobile plattforms like the iPhone and android. To test this, go to the RageSpline testlevel, activate the „mobile_input“ boolean of the playerMovement script of 2DPlayer and activate the gameObjects and childrens of „Mobile_Setup“. The testlevel is now ready to be played on mobile platforms!

In the script-folder, you will find the „Joystick“ script. This is a modified version of the one from unity. It's now able to create buttons too and got some more functions like emulate the „ButtonDown“ function of the Input namespace.

Use „freeze_position“ to freeze the joystick and makes it more like a button.

Smooth Moves

If you want to use smooth moves for your player, you have to edith the „AnimationSM.js“ file. Open it up and delte the „/*“ and „*/“ on line 4 and 82. Now, delete everything below „///DELETE THE FOLLOWING FUNCTIONS AND VARS IF YOU WANT TO USE SMOOTH MOVES!!!“.
After that, you can use the AnimationSM component just like the Animation2D and 3D components.