

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

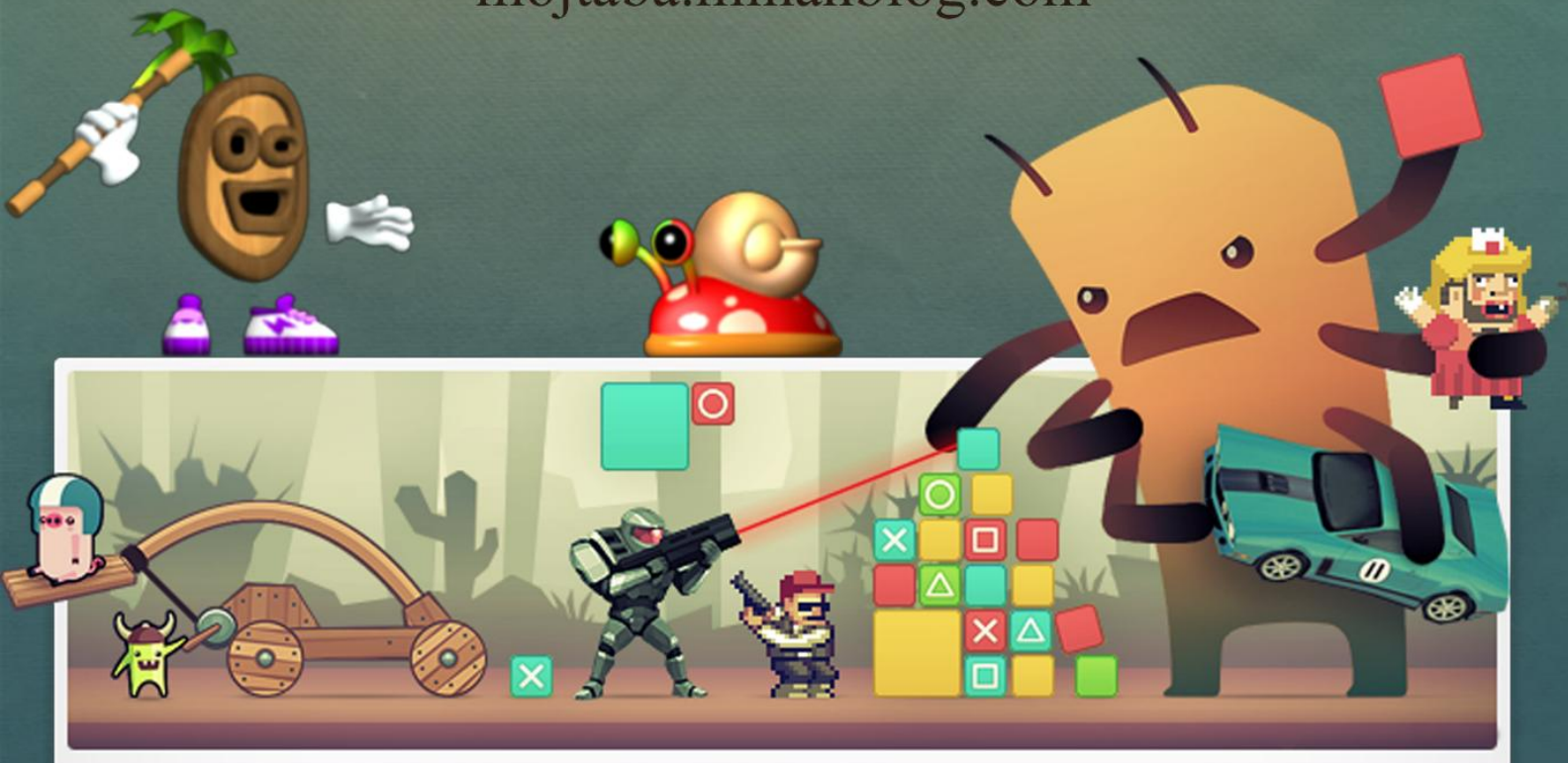
چگونه بازی پلت فرم بسازیم؟

به مناسبت میلاد رسول اکرم (ص)



مجتبی قاسم زاده تهرانی

mojtaba.mihanblog.com




مقدمه

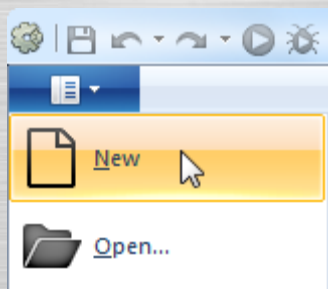
کنستراکت ۲ می‌تونه هر نوع بازی دو بعدی رو بسازه. تو این آموزش اصول ساخت یک بازی پلت فرم رو در کنستراکت ۲ یاد می‌گیرید. تذکر: در این آموزش از اسپرایت‌های Jungle Platform استفاده کردیم که می‌تونید از [این جا](#) دانلودشون کنید یا هم می‌تونید خودتون مشابه اون‌ها رو طراحی کنید.

نصب کنستراکت ۲

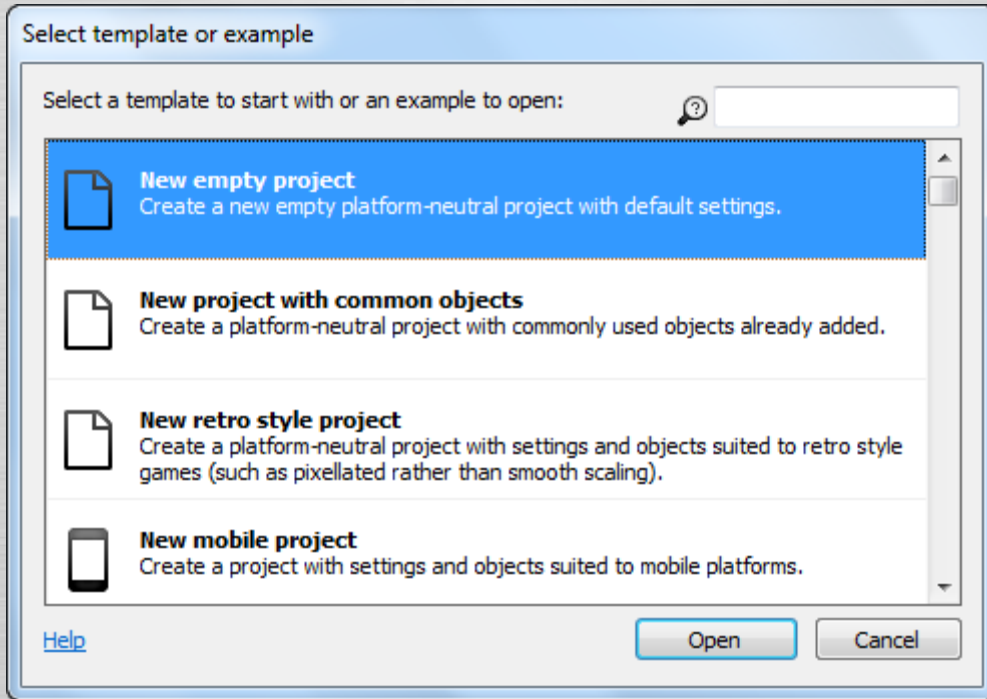
اگه کنستراکت ۲ رو ندارید می‌تونید از [این جا](#) آخرین نسخه‌ی رایگانش رو دانلودش کنید. کنستراکت ۲ فقط مال ویندوزه ولی بازی‌هایی که باهاش ساخته می‌شن می‌تونند توی همه‌جا مثل مکینتاش، لینوکس، آیبی، اندروید و... اجرا بشن. کنستراکت ۲ در حالتی که شما ادمن هم نباشین توی کامپیوترتون نصب می‌شه. کنستراکت ۲ پرتابل هم هست یعنی می‌تونید توی فلش‌تون نصب کنیدش و هر جایی اون رو با خودتون ببرید!

شروع

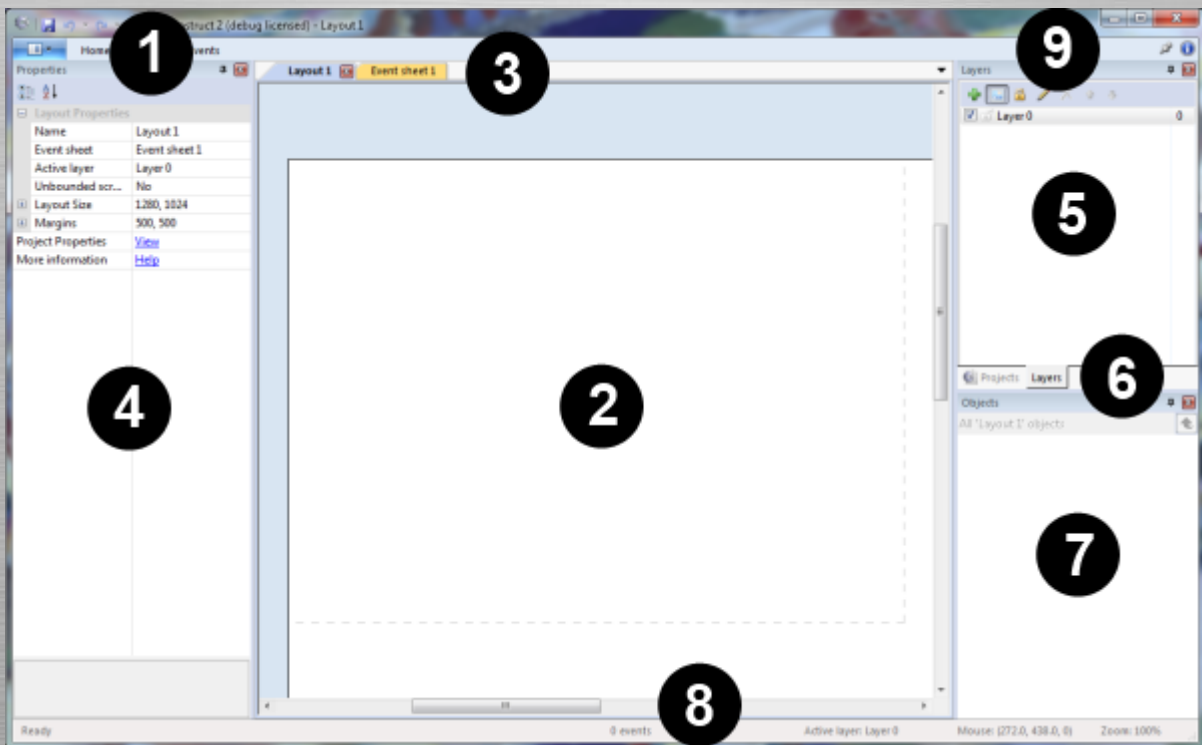
بعد از اینکه کنستراکت ۲ رو نصب کردید، اونو اجرا کنید. بعد روی دکمه‌ی File () کلیک کنید و New رو انتخاب کنید.



بعد پنجره‌ی New Project باز می‌شه، فقط کافیه که روی New empty project کلیک کنید.



حالا باید لیوت 'خالی رو ببینید. لیوت چیه؟ لیوت جاییه که ما توی اون بازی مون رو طراحی می کنیم اشیاء رو به وجود میاریم و مکانشون رو تغییر می دیم. لیوت یه چیزیه مثل مرحله یا منوی بازی که توی نرم افزارهای دیگه بهش room، scene یا frame هم می گن. اگه هنوز متوجه نشدید عکس پایین رو نگاه کنید، شماره ی ۲ همون لیوته:



سایز لیوت

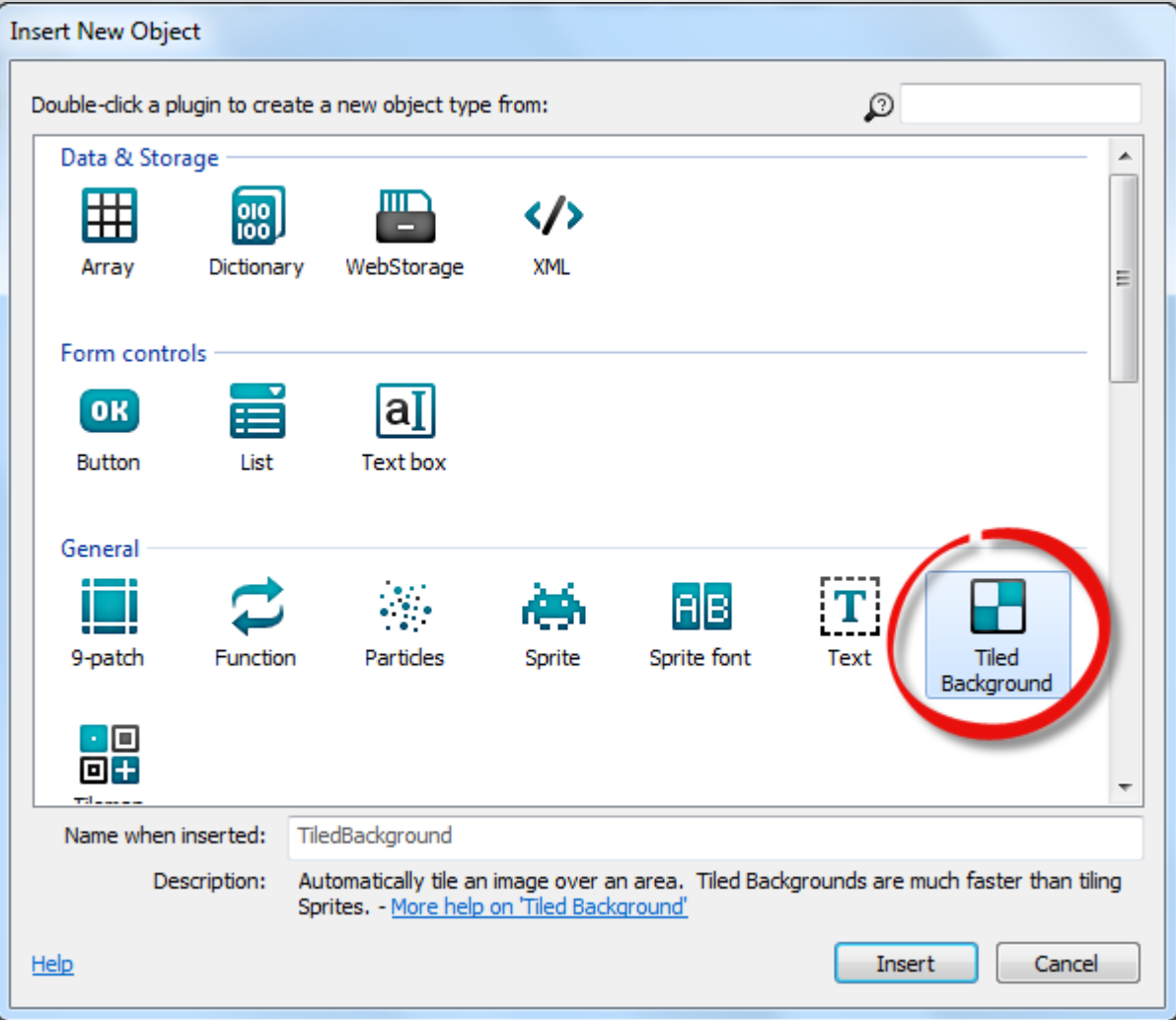
اول از همه ما می‌خواهیم لیوت‌مون (مرحله‌مون) یکم بزرگتر باشه. یه جای خالی تو لیوت کلیک کنید تا توی نوار ویژگی‌ها تنظیماتش رو ببینیم. (توی عکس بالا شماره‌ی ۴ همون نوار ویژگی‌هاست). بعد از توی این نوار همون طور که توی عکس می‌بینید اندازه‌ی لیوت رو بنویسید ۴۰۰۰×۲۰۴۸.


Layout properties	
Name	Layout 1
Event sheet	Event sheet 1
Active layer	Layer 0
Unbounded scro...	No
Layout Size	4000, 2048
⊕ Margins	500, 500
Effects	
Add / edit	Effects
Project Properties	View
More information	Help

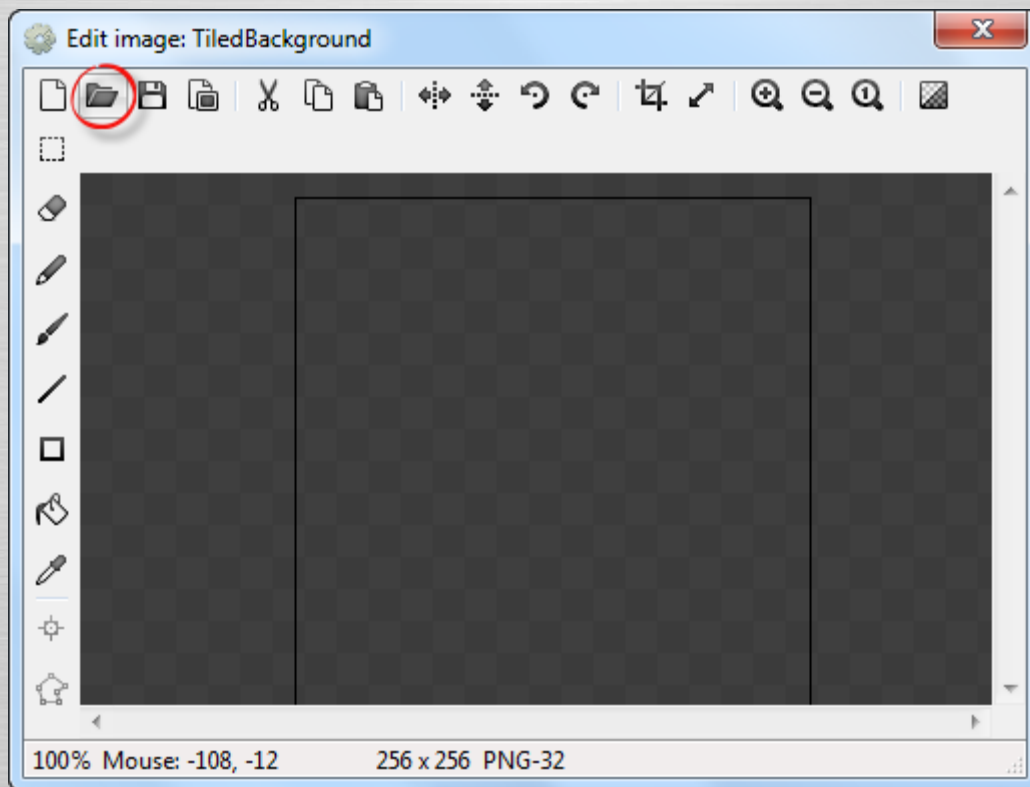
بیاین یه عکس پس‌زمینه برای لیوت‌مون بذاریم. یه جای خالی لیوت دابل کلیک^۳ کنید تا پنجره‌ی Insert New Object باز بشه، روی شیء Tiled Background دابل کلیک کنید تا بیاد توی بازی.

^۲ Properties bar

^۳ همون دوتا کلیک پشت سر هم



برای اینکه بگید پس زمینه رو کجای لیوت بذاره موس تبدیل به به علاوه (+) می شه. این که پس زمینه رو اول کجا بذارید مهم نیست پس فقط به جای لیوت کلیک کنید تا ویرایشگر تصویر^۴ باز بشه. حالا بیاین پس زمینه رو وارد بازی مون کنیم. پس زمینه رو از آدرس `Open Background Image\Background.png` توی اون پکی که دانلود کردید وارد کنید. برای این کار روی آیکن  کلیک کنید.



حالا Background Image\Background.png رو انتخاب کنید و بعد ویرایشگر تصویر رو ببندید. همون طور که می‌بینید فقط یه تیکه از پس زمینه توی لیوت دیده می‌شه. ما باید اندازه‌ش رو عوض کنیم تا همه‌ی لیوت رو بگیره. اول توی لیوت روی پس‌زمینه کلیک کنید تا انتخاب بشه و تنظیماتش توی نوار ویژگی‌ها دیده بشه. برای این که همه‌ی لیوت رو بپوشونه باید Position رو 0,0 و Size رو 4000,2048 (همون اندازه‌ی لیوت) بکنید.

Object type properties	
Name	TiledBackground
Plugin	Tiled Background
UID	0
Global	No
Common	
Layer	Layer 0
Angle	0
Opacity	100
Position	0, 0
Size	4000, 2048
Instance variables	
Add / edit	Instance variables
Behaviors	

چون پس‌زمینه خیلی بزرگه در حالت عادی نمی‌تونید همه‌ش رو یکجا ببینید. پس می‌خواهیم به عقب زوم کنیم تا همه‌ش دیده بشه. کلید کنترل (Ctrl) رو نگه دارید و غلتک موس رو به پایین بچرخونید تا به عقب زوم کنید. اگه هم غلتک موس ندارید (مثلاً توی لپ‌تاپ) از سربرگ View روی گزینه‌ی Zoom out چند بار کلیک کنید. حالا باید تونسته باشید که همه‌ی لیوت رو ببینید، مثل این:



اون قسمتی رو که می بینید خط چین شده اندازه‌ی پنجره‌ی بازی رو (یعنی اون قسمتی از لیوت که موقع بازی کردن دیده می‌شه) در مقایسه با اندازه‌ی لیوت نشون می‌ده. **Ctrl + 0** رو فشار بدین تا به حالت زوم قبلی برگردید، یا هم می‌تونید از سربرگ **View** روی **Zoom to 100%** کلیک کنید. به احتمال زیاد الان اومدید وسط لیوت. حالا برای این که دوباره به گوشه‌ی بالا سمت چپ برگردید از نوارهای پیمایش استفاده کنید یا با غلتک موس درگ^۵ کنید، اگه هم غلتک موس ندارید به جای غلتک موس، اسپیس رو نگه دارید.

اضافه کردن لایه^۶

خوب، حالا می‌خوایم چند تا شیء دیگه اضافه کنیم. اما همه‌ش تصادفی روی پس‌زمینه کلیک می‌کنیم و اونو انتخاب می‌کنیم مگه این که قفلش کنیم تا دیگه انتخاب نشه. بیاین برای این کار از سیستم لایه‌بندی استفاده کنیم.

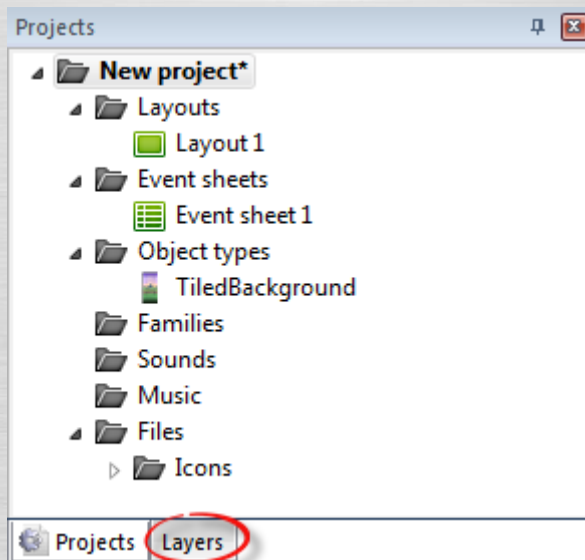
توی هر لیوت می‌تونه چند تا لایه باشه که برای گروه‌بندی اشیاء می‌تونید از شون استفاده کنید. برای این که بهتر بفهمیم لایه چیه این طوری در نظر بگیرید که لایه‌ها چند تا برگه‌ی شیشه‌ای هستن که اشیاء روشن نقاشی شدن. با کمک لایه‌ها خیلی راحت می‌تونید تنظیم کنید کدوم شیء روی اون یکی دیگه دیده بشه. لایه‌ها رو می‌تونید مخفی یا قفل کنید، بهشون افکت پارالاکس^۷ بدید و غیره. مثلاً توی این بازی می‌خوایم همه‌ی چیزهای دیگه جلوی پس‌زمینه‌مون دیده بشن، بنابراین می‌تونیم یه لایه‌ی دیگه روی لایه‌ای که الان وجود داره بسازیم و اشیاءمون رو توی اون بذاریم.

برای مدیریت لایه‌ها، روی سربرگ **Layers** که معمولاً کنار نوار پروژه‌ست کلیک کنید.

^۵ چیزی رو با موس گرفتن و کشیدن می‌شه درگ و بعد ولش کردن می‌شه دراپ.

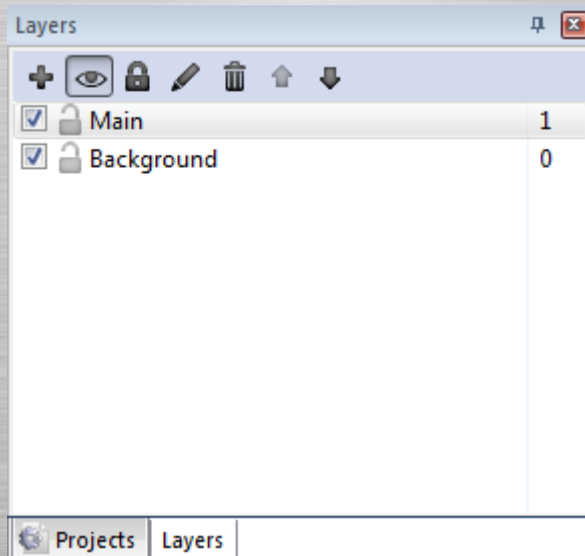
^۶ Layer

^۷ پارالاکس (Parallax) موقع حرکت مثلاً با ماشین چیزهایی که نزدیکمون هستن با سرعت زیاد حرکت می‌کنند، اما چیزهایی که دور هستن با سرعت کمتری حرکت می‌کنند (مثل کوه‌ها). به این اختلاف سرعت منظره‌های نزدیک و دور پارالاکس گفته می‌شه.



باید توی اون لیست Layer 0 رو ببینید (کنستراکت ۲ از صفر شروع به شمردن می کنه مثل برنامه نویسی). روی آیکن مداد (✎) کلیک کنید و اسمش رو بذارید Background. به خاطر این که این لایه مال پس زمینه مونه. حالا روی آیکن به علاوه (+) کلیک کنید تا یه لایه جدید برای اشیاء مون به وجود بیاد. بیاین اسم این یکیو Main بذاریم. اگه روی آیکن قفل که کنار Background هست کلیک کنیم، قفل می شه، که یعنی هیچ کدوم از اشیاء اون لایه رو دیگه نمی تونیم انتخاب کنیم. این برای پس زمینه مون خیلی خوبه و باعث می شه تصادفی انتخابش نشه و جابجا بشه. اما اگه یه موقع خواستید پس زمینه رو تغییر بدید فقط کافیه روی آیکن قفل دوباره کلیک کنید تا قفلش باز شه.

چک باکس ها برای این هستن که بتونید لایه ها رو مخفی کنید (البته فقط موقع ساخت بازی نه اینکه توی بازی هم مخفی شون کنه)، اما فعلاً لازم نیست این کارو بکنیم. نوار Layers شما باید تا حالا این شکلی شده باشه:



حالا روی لایه ی Main کلیک کنید. این کار خیلی مهمه. لایه ای در حالت انتخاب باشه لایه ی فعال محسوب می شه. همه ی اشیاء جدیدی که به بازی مون اضافه می شن می رن تو لایه ی فعال. بنابراین اگه لایه ی Main انتخاب نشده باشه، بدون این که بخوایم اشیاء رو توی لایه ی اشتباه می داریم. لایه ی فعال توی نوار وضعیت هم دیده می شه. موقعی هم که می خواید مکان یک شیء جدید رو تعیین کنید توی یه کادر کوچیک کنار موس اسم لایه ی فعال دیده می شه – همیشه لایه ی فعال رو زیر نظر داشته باشید.

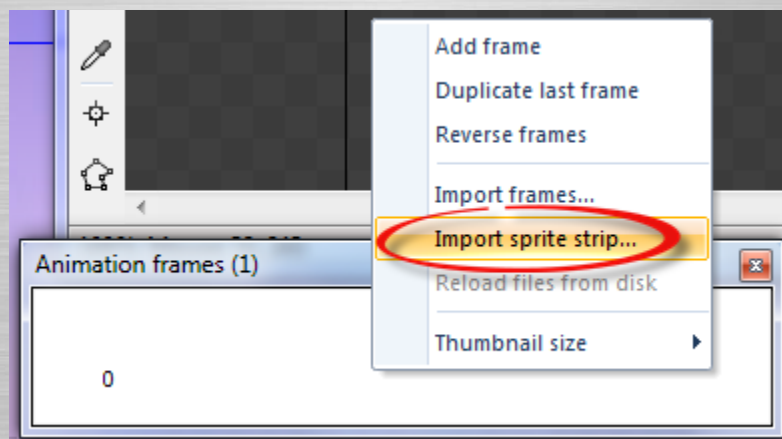
اضافه کردن تایل^۸

یک تایلست توی Tiles\Tiles.png وجود داره، بیاین این‌ها رو توی یه شیء اسپرایت^۹ بذاریم و از اون به عنوان یک شیء تایل استفاده کنیم.

مشابه همون کاری که برای پس‌زمینه انجام دادید، یه جای خالی توی لیوت دابل کلیک کنید تا یک شیء جدید بذارین تو بازی، حالا روی Sprite دابل کلیک کنید. موس دوباره شکل + می‌شه. یه جا وسط صفحه کلیک کنید.

ویرایشگر تصویر باز می‌شه، اما این دفعه پنجره‌های بیشتری داره، چون اسپرایت می‌تونه انیمیشن^{۱۰} هم داشته باشه. اما نمی‌خوایم انیمیشن پخش بشه، باید سرعتش رو صفر کنیم. هر فریم انیمیشن ما یک تایل. پس می‌تونیم با عوض کردن فریم انیمیشن تعیین کنیم کدوم تایل دیده بشه.

یه جای خالی پنجره‌ی Animation Frames که زیر عکسمون هست کلیک راست کنید، و Import sprite strip... رو انتخاب کنید. با این کار می‌تونید تایل‌ها رو ببرید و هر کدوم رو به عنوان فریم انیمیشن وارد کنید.

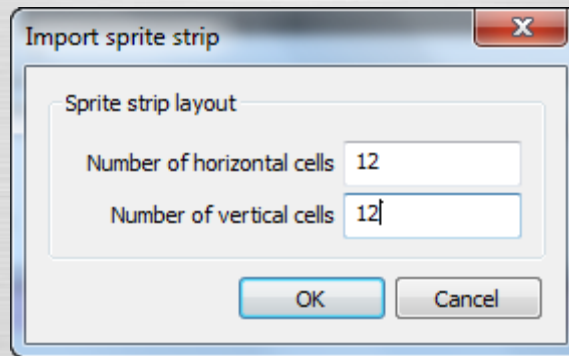


پنجره‌ی باز کردن فایل باز می‌شه، از توی اون پکی که دانلود کردید Tiles\Tiles.png رو انتخاب کنید. کنستراکت ۲ باید بدون توی این تایلست چند تا تایل وجود داره. از طرفی اگه نگاه کنید می‌بینید این فایل Tiles.png یک شبکه‌ی ۱۲×۱۲ از تایل‌هاست، پس توی کادرها ۱۲ و ۱۲ رو وارد کنید و روی OK کلیک کنید.

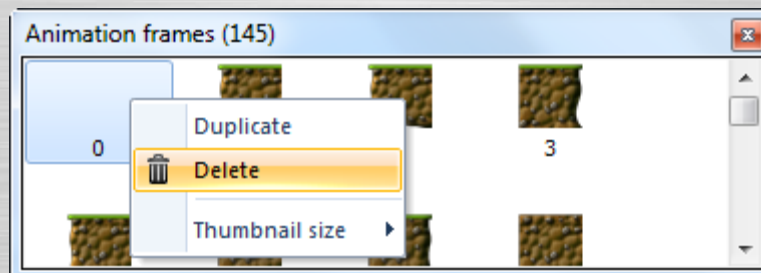
^۸ تایلست (Tileset): عکس‌هایی هستن که از تعداد زیادی عکس‌های کوچیکتر که مثل کاشی‌کاری کنار هم قرار گرفتن تشکیل شدن. به این عکس‌ها تایلست و به هر کدوم از اون عکس‌های کوچیکتر تایل گفته می‌شه. برای کار با تایلست‌ها به تازگی امکان جدیدی به کنستراکت ۲ اضافه شده به نام tilemap که برای کار با تایلست‌ها خیلی بهتره اما چون این آموزش مال نسخه‌های قدیمی‌تر کنستراکت ۲ هست و اون موقع این ویژگی هنوز اضافه نشده بود از روش دیگه‌ای برای استفاده از تایلست‌ها استفاده شده است.

^۹ اسپرایت (Sprite): همون شخصیت‌های بازی رو بهشون می‌گیم اسپرایت.

^{۱۰} انیمیشن: تعدادی عکس ثابت که پشت سر هم نمایش داده می‌شن. به هر کدوم از این عکس‌های ثابت یک فریم گفته می‌شه.

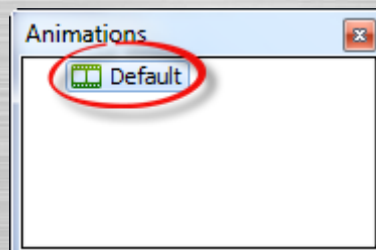


کمی صبر کنید تا عکس تیکه تیکه بشه. حالا باید ۱۴۴ تا تایل رو که همون فریم‌های انیمیشن هستن داشته باشین! دیگه به فریم اول که خالیه احتیاجی نداریم، پس روش کلیک راست کنید و delete رو انتخاب کنید تا حذف بشه.

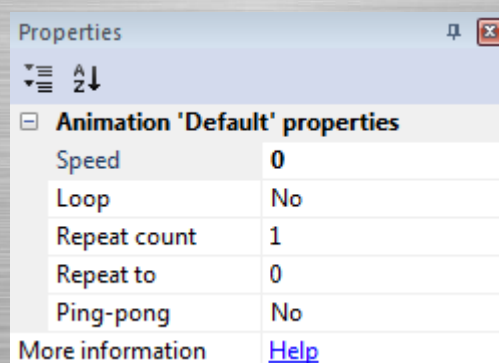


شاید بخواین پنجره‌ی Animation frames رو به کم بزرگتر کنید تا به نمای بهتر از تایل‌ها داشته باشید. درضمن اگه ترجیح می‌دین که عکس‌های کوچیک توی این پنجره بزرگتر بشن می‌تونید به جای خالی کلیک راست کنید و Thumbnail size -> Large رو انتخاب کنید.

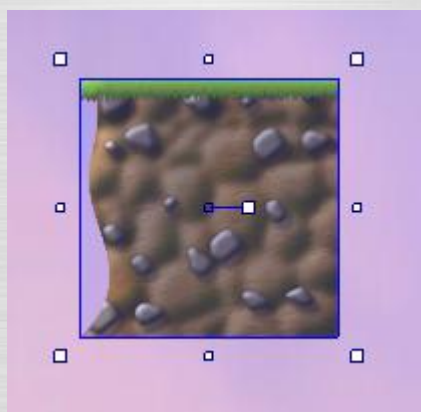
به طور پیش‌فرض سرعت انیمیشن ۵ هست، یعنی توی هر ثانیه ۵ تا فریم پشت سر هم نمایش داده می‌شه و باعث می‌شه تصویرمون مدام تغییر کنه، ما می‌خوایم تایل‌ها توی فریم خودشون ثابت بمونن، پس بیاین سرعت انیمیشن رو صفر کنیم. توی پنجره‌ی Animations انیمیشن Default رو انتخاب کنید.



الآن نوار ویژگی‌ها تنظیماتی رو برای انیمیشن نشون می‌ده. حالا توی این نوار Speed رو 0 کنید.



حالا دیگه تایل ها تغییر نمی کنن – اون ها روی فریم خودشون می مونن. ویرایشگر تصویر رو ببندید. اسپرایت تایل تون باید توی لیوت باشه!



حالا روش کلیک کنید و توی نوار ویژگی ها اسمش رو از Sprite به Tile تغییر بدید. خیلی خوبه که همیشه به اشیاء تون اسم های مناسب بدین.

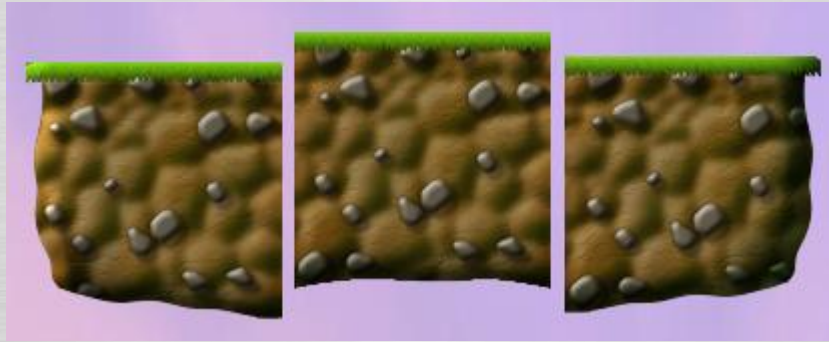
ساختن منظره با تایل ها

شیء اسپرایت توی نوار ویژگی ها یه تنظیم داره به اسم Initial frame که یعنی فریم اولیّه (مشخص می کنه که در حالت عادی کدوم فریم رو نمایش بده که همون طور که می دونید در این جا فریم ها همون تایل ها هستن). ما می تونیم برای عوض کردن تایل فعلی مون از این ویژگی استفاده کنیم. شیء تایل رو انتخاب کنید و Initial frame رو به ۱ تغییر بدین:

Properties	
Animations	Edit
Size	Make 1:1
Initial visibility	Visible
Initial animation	Default
Initial frame	1
Collisions	Enabled
More information	Help

همون طور که دیدید تصویر تغییر کرد تا تایل شماره ۱ انیمیشن رو نمایش بده. از این راه می تونید در سراسر مرحله از تایل های مختلف استفاده کنید.

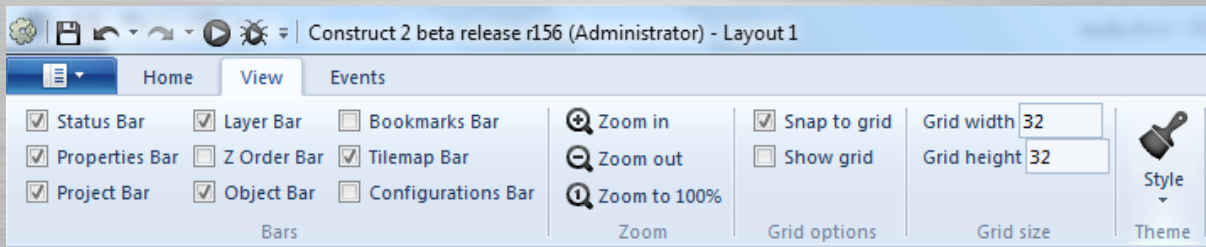
کلید کنترل رو نگه دارید و تایل رو درگ کنید تا از روش کپی بشه. این کار رو ۳ بار دیگه انجام بدید تا توی یک ردیف ۳ تا تایل داشته باشیم. وسطی رو انتخاب کنید و Initial frame رو به ۱۰ تنظیم کنید. همین کار رو برای سمت راستی هم انجام بدین، ولی این بار Initial frame رو ۱۳ بذارید. حالا باید یه چیزی مثل این داشته باشید:



اگره نمی‌دونید چی شماره‌ای مال کدوم تایله، فقط روی شیء دابل کلیک کنید. ویرایشگر تصویر با پنجره‌ی Animation frames که می‌تونید توش بررسی کنید که کدوم شماره مربوط به کدوم تایله دوباره باز می‌شه. درضمن اگرخواستین یک تایله دیگه از یه نوعی که قبلاً ساختین به وجود بیارین لازم نیست Initial frame رو تغییر بدید چون با نگه داشتن کنترل و درگ کردن یه تایله از همون نوع یک کپی ساخته می‌شه.

فعال کردن قابلیت چسبیدن به گرید^{۱۱}

اگره بخواین همین طوری با دست تایله‌ها رو توی یک خط منظم مرتب کنید پدروتن در میاد! می‌تونید با قابلیت چسبیدن به گرید خیلی راحت این کار رو انجام بدین. روی سربرگ View کلیک کنید و تیک Snap to Grid رو بزنین. اندازه‌ی پیش‌فرض گرید که 32×32 هست خوبه و لازم نیست تغییرش بدین. تایله‌های ما 128×128 هستن که یه مضربی از 32 هست. این طوری هم چسبوندن تایله‌ها به هم راحت می‌شه و هم برای تعیین مکان تایله‌ها بهمون یه مقداری آزادی می‌ده.



حالا وقتی تایله‌ها رو با موس درگ می‌کنید می‌فهمید که به یک گرید 32×32 دارن می‌چسبند. حالا خیلی راحت ۳ تایله رو به هم می‌چسبونیم.

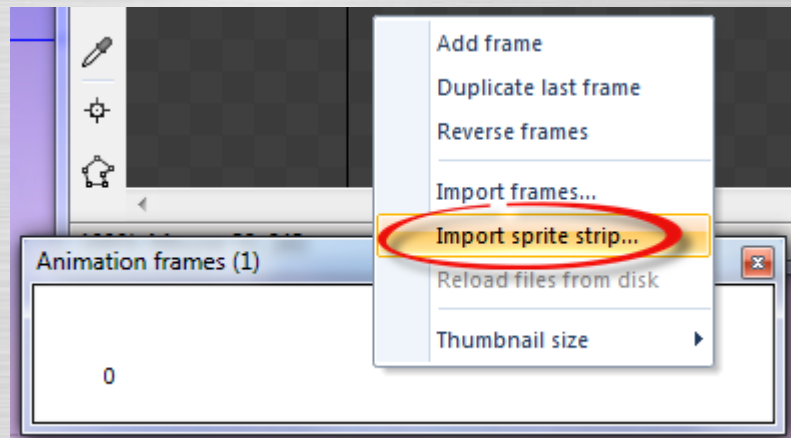


اضافه کردن اسپرایت پلیر^{۱۲}

^{۱۱} شبکه‌ای از خطوطی است که برای اینکه بتوانیم اشیاء را مرتب کنار هم بچینیم، اشیاء به آن‌ها می‌چسبند.
^{۱۲} پلیر (Player): بازیکن

بیان برای پلیر به اسپریت اضافه کنیم. مثل قبل، به جای خالی دابل کلیک کنید تا به شیء جدید بذارین تو بازی، بعد Sprite رو انتخاب کنید. وقتی + دیده شد، به جا بالای تایلها کلیک کنید تا ویرایشگر تصویر ظاهر بشه.

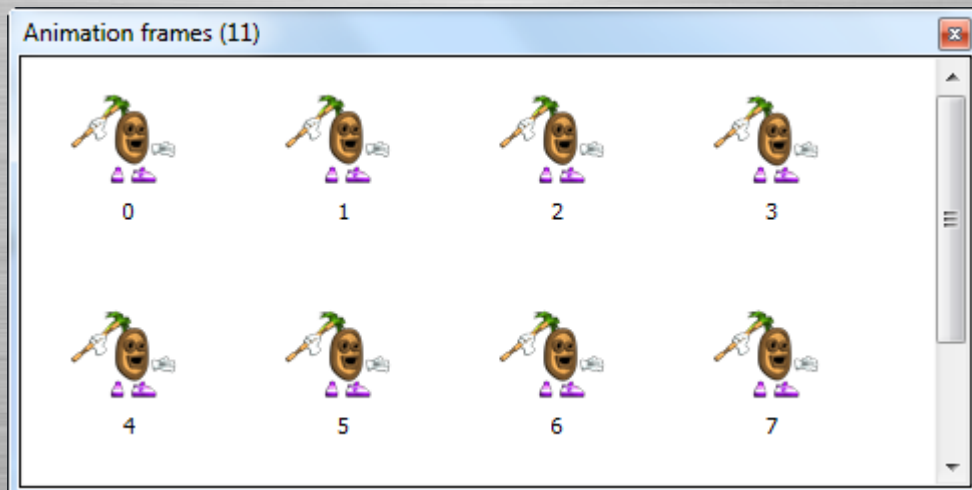
بیان به اسپریت استریپ^{۱۳} برای وقتی که پلیر کاری انجام نمیده و بی کاره بیاریم. مثل قبل، روی پنجره Animation frames کلیک راست کنید و گزینه Import sprite strip... رو انتخاب کنید.



فایل Player\Idle\Idle.png رو انتخاب کنید. اسپریت استریپ به طور کامل یکی از فریمهای انیمیشنمون نیست. درضمن می تونید فریمها رو به صورت تک تک با گزینه Import frames هم بیارید تو، ولی فعلاً بذارین از همین روش خودمون استفاده کنیم.

چون استریپها فقط یه ردیف دارند، کنستراکت ۲ تعداد فریمها رو حدس میزنه، باید درست فهمیده باشه که فریمها ۱۱ تا هستند. روی OK کلیک کنید تا فریمها رو بیارین تو.

باز هم اول همه فریمها اون فریم خالی پیش فرض وجود داره. کلیک راست کنید و دوباره حذفش کنید. حالا باید ۱۱ تا فریم پشت سر هم برای انیمیشن بی کار بودن پلیرداشته باشید.



بریدن

^{۱۳} اسپریت استریپ: مشابه تابلست است با این تفاوت که در تابلست هر بخش یک تایل محسوب می شود و در اسپریت استریپ هر بخش یک فریم انیمیشن، به عبارت دیگر تابلست انیمیشن نیست و باید سرعت انیمیشنش را صفر کنیم ولی اسپریت استریپ را نه.

شاید فهمیده باشین که دور و بر پلیر مقداری فضای خالی وجود داره. معمولاً وقتی تصاویر رو میارید این مشکل پیش میاد. اما کار بدیه همین طوری ولش کنیم چون الکی حافظه رو تلف می‌کنه و می‌تونه تشخیص برخوردارهاش با بقیه‌ی اشیاء رو نامعتبر کنه.



برای خلاص شدن از شر این فضای اضافی کافیه که کلید شیفت رو نگه دارید و روی دکمه‌ی Crop (C) که توی نوار ابزار ویرایشگر تصویر وجود داره کلیک کنید.



اگه شیفت رو نگه ندارید فقط فضای اضافی همین فریمی که الان توش هستید از بین می‌ره و فریم‌های دیگه هنوز اون فضای اضافی رو دارند. پس حتماً موقعی که روی دکمه‌ی Crop کلیک می‌کنید شیفت رو هم نگه دارید تا فضای اضافی همه‌ی فریم‌های انیمیشن از بین بره. حالا باید عکس پلیر برش خورده باشه و قسمت‌های خالی و اضافی‌اش حذف شده باشه.

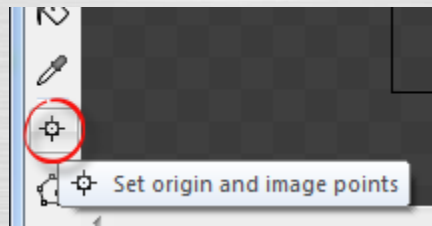


تنظیم نقطه‌ی مبدأ^{۱۴}

به طور پیش‌فرض نقطه‌ی مبدأ وسط عکس قرار داره. تو بازی‌های پلت‌فرم بهترین جا برای نقطه‌ی مبدأ پاهای پلیره. چون اگه موقع اجرای انیمیشن قدش کوتاه و بلند بشه به طرف بالا کوتاه و بلند می‌شه و روی زمین راه می‌ره. در غیر این صورت ممکنه به طرف پایین قدش کوتاه و بلند بشه و توی زمین گیر کنه!

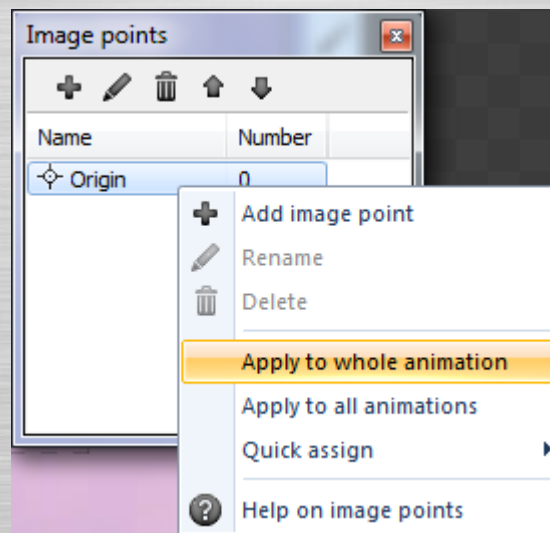
برای این که تنظیم کنید نقطه‌ی مبدأ کجا باشه، روی تو ویرایشگر تصویر روی Set origin and image points کلیک کنید.

^{۱۴} Origin: نقطه‌ای روی شیء هست که کاربردهای زیادی داره مثلاً موقع قرینه شدن نسبت به اون نقطه قرینه می‌شه، موقع عوض شدن انیمیشن نقطه‌ی مبدأ انیمیشن جدید روی مبدأ انیمیشن قدیمی قرار می‌گیره و...



حالا باید فهمیده باشیم که به نقطه‌ی قرمز روی پلیر ظاهر شد. اون نقطه همون نقطه‌ی مبدأ هست. می‌تونید کلیک کنید تا مکانش عوض بشه. اما چون ما می‌خوایم جای نقطه دقیقاً وسط و پایین عکس باشه کافیست کلید کنترل رو نگه دارید و از صفحه کلید عددی سمت راست کیبوردتون عدد ۲ رو فشار بدید تا نقطه‌ی مبدأ دقیقاً وسط و پایین قرار بگیره (البته باید Num Lock روشن باشه). اگه از لپتاپ استفاده می‌کنید که اون صفحه کلید عددی رو نداره توی پنجره‌ی Image points روی Origin کلیک راست کنید و Quick Bottom -> assign رو انتخاب کنید.

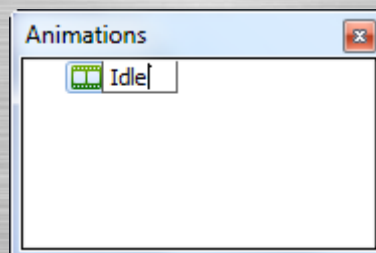
اگه بخوایم این کار رو برای تک تک فریم‌ها انجام بدیم پدرمون در میاد! اما خوشبختانه یه میانبر براش وجود داره. توی پنجره‌ی Image points روی Origin کلیک راست کنید و Apply to whole animation رو انتخاب کنید.



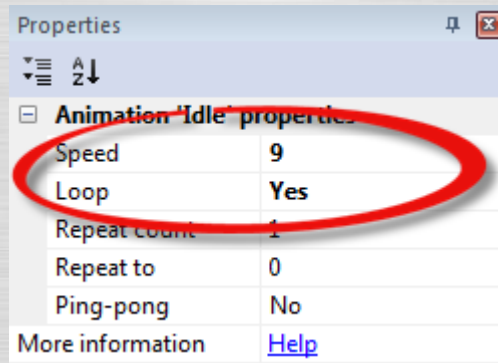
خودشه! مبدأ برای همه‌ی فریم‌های انیمیشن درست تنظیم می‌شه.

تکرار انیمیشن

توی پنجره‌ی Animations روی انیمیشن Default کلیک کنید و اسمش رو به Idle (به معنی بی‌کار) تغییر بدین.



توی نوار ویژگی‌ها ^{۱۵}Speed رو بذارید ۹ و ^{۱۶}Loop رو بذارید Yes.



توی پنجره‌ی Animations روی انیمیشن Idle کلیک راست کنید و Preview رو انتخاب کنید تا پیش‌نمایش انیمیشن رو ببینید. حالا باید ببینید که پلیئر خیلی آرام سرش رو به بالا و پایین تکون می‌ده. تموم شد! پنجره‌ی پیش‌نمایش که باز شده رو ببندید و ویرایشگر تصویر رو هم همین‌طور. باید پلیئر تون رو توی لیوت ببینید.

توی نوار ویژگی‌ها اسم شیء تون رو به Player تغییر بدین، چون ما می‌خوایم کارامون نظم داشته باشه و اسم‌های شیء‌های مختلف رو با هم قاطی نکنیم.

اضافه کردن رفتار^{۱۷}

کنستراکت ۲ یه عالمه رفتار داره. این رفتارها باعث می‌شن اشیاء یه کارایی رو که از قبل براشون تعریف شده انجام بدن، که بیشتر اوقات یه عالمه توی زمان صرفه‌جویی می‌شه. امکانش وجود داره که همه‌ی کارهایی رو که رفتارها انجام می‌دن با سیستم ایونت^{۱۸} انجام بدین، اما معمولاً کار سختیه و وقت زیادی رو ازتون می‌گیره، به خاطر همین این رفتارهای دم دستی بازی تون رو خیلی سریع راه می‌اندازن و اجرا می‌کنند.

یکی از این رفتارها رفتار Platform هست که ما رو از شر پیچیدگی‌های ساخت حرکت پلیئر راحت می‌کنه. اما نکته‌ی مهمی که اینجا وجود داره اینه که این رفتار باید به یک شیء مستطیلی شکل و نامرئی داده بشه و پلیئر همیشه روی این مستطیل باشه. اگه شیئی که بهش رفتار پلت فرم می‌دیم انیمیشن نداشته باشه رفتار Platform خیلی بهتر کار می‌کنه، چون تغییر فریم‌های انیمیشن می‌تونه باعث بشه شیء توی دیوار گیر کنه و دیگه نتونه حرکت کنه و رفتار Platform سردرگم می‌شه. در ضمن جلوی بعضی از برخوردهای احمقانه رو می‌گیره، مثلاً باعث می‌شه دیگه نتونه با دماغش از یک لبه آویزون بشه.

پس اول ما یک مستطیل نامرئی می‌خوایم تا حرکت رو به اون بدیم. توی لیوت دابل کلیک کنید تا دوباره یه اسپریت جدید به وجود بیارید، حالا یه مربع مثل مربع زیر بیارید تو.



مثل قبل نقطه‌ی مبدأ باید پایین باشه. به همین دلیل دوباره روی Set origin and image points کلیک کنید و ۲ رو فشار بدید (یا هم از منوی quick assign استفاده کنید) تا مبدأ رو پایین عکس قرار بدید.

^{۱۵} سرعت انیمیشن و تعیین می‌کنه توی هر ثانیه چند تا فریم پشت سر هم به نمایش در بیاد.

^{۱۶} Loop یعنی وقتی انیمیشن یک دور پخش شد دوباره از اول پخش بشه.

^{۱۷} Behavior

^{۱۸} ایونت (Event): رویداد

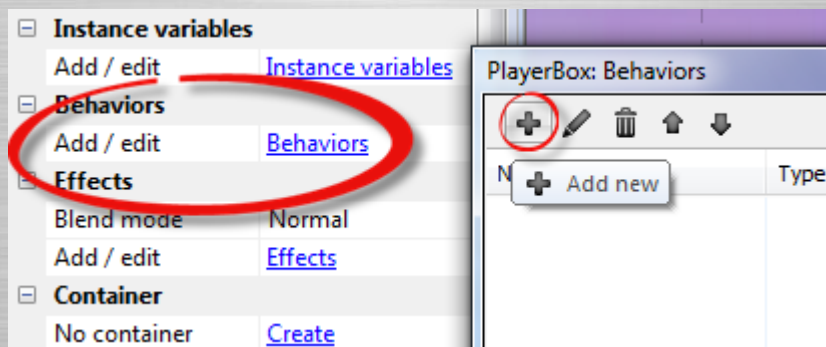


ویرایشگر تصویر رو ببندید. حالا باید این مربع رو توی لیوت ببینید. اندازهش رو تغییر بدید تا تقریباً هم اندازه‌ی بدن پلیمر بشه مثل شکل زیر که اندازهش ۵۳×۱۰۷ هست.

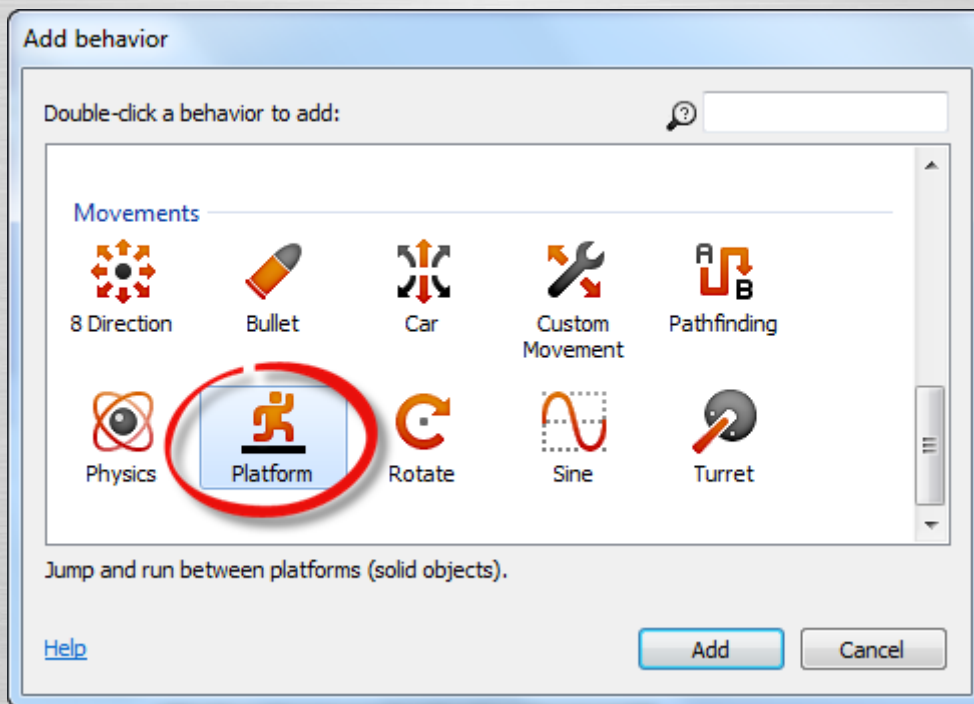


اسمش رو به **PlayerBox** تغییر بدید، چون می‌تونیم بگیم که این یک جعبه‌ست که برای حرکت و بررسی برخورد پلیمر با بقیه‌ی اشیاء استفاده می‌شه. درضمن از نوار ویژگی‌ها **Intial visibility** رو به **Invisible** تغییر بدین که توی بازی نامرئی بشه چون ما نمی‌خوایم دیده بشه.

حالا می‌خوایم به شیء **PlayerBox** رفتار **Platform** بدیم. از نوار ویژگی‌های **PlayerBox**، روی **Behaviors** که با رنگ آبی نوشته شده کلیک کنید. توی پنجره‌ای که ظاهر می‌شه روی دکمه‌ی به علاوه کلیک کنید.



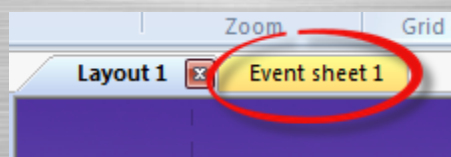
روی رفتار **Platform** دابل کلیک کنید.



توی نوار ویژگی‌ها چند تا تنظیم جدید برای حرکت پلت فرم اضافه می‌شه. شما می‌تونید تنظیمات حرکت مثل سرعت و شتاب رو تغییر بدید. بیاین پرش‌ها رو یه کم روح دارتر بکنیم. Jump strength رو بذارید ۱۱۰۰ و Gravity رو بذارید ۲۵۰۰. درضمن، ما می‌خوایم هر جا پلیرمون حرکت می‌کنه صفحه هم دنبالش بره، پس دوباره روی آیکون به علاوه کلیک کنید و این بار رفتار Scroll To که برای همین کار هست رو انتخاب کنید. بعد از اینکه این دو تا رفتار رو اضافه کردید پنجره‌ی رفتارها رو ببندید.

۱۹ اضافه کردن رفتار Solid

رفتار پلت فرم باید بدونه که روی کدوم اشیاء حق داره راه بره. پلیر روی اشیائی که رفتار Solid ندارند راه نمی‌ره و از روشون سقوط می‌کنه. حالا اسپرایت Tile رو انتخاب می‌کنیم و بهش رفتار Solid می‌دیم تا پلیر بتونه روش راه بره. حالا باید بگیم که پلیر باید همیشه روی PlayerBox باشه. برای این کار به Event Sheet 1 برید. اینجا جاییه که ما با کمک سیستم ایونت کنستراکت ۲ تنظیم می‌کنیم که در چه وقتی چه چیزی اتفاق بیفته.



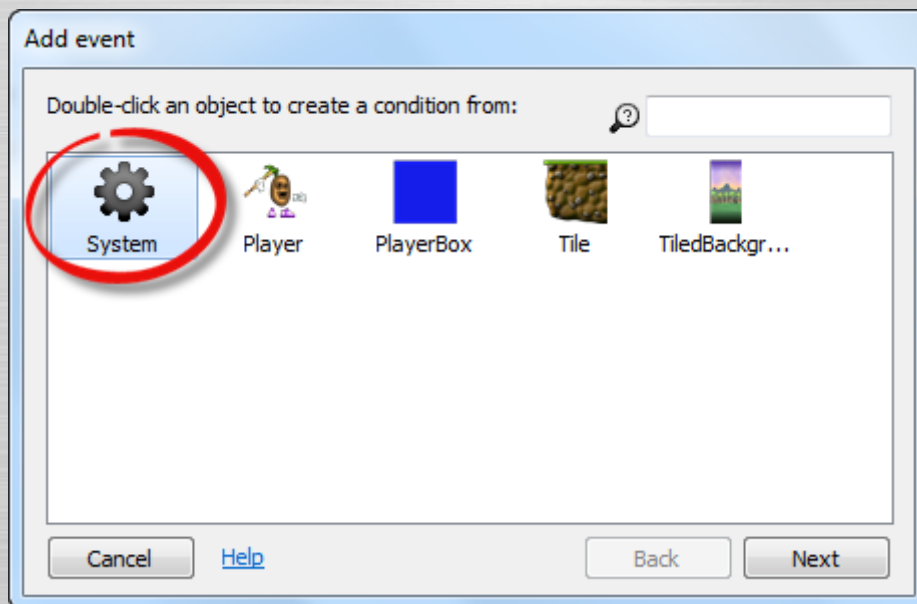
ایجاد ایونت

ایونت‌ها یک سری شرایط رو بررسی می‌کنند و در صورتی که اون شرایط برقرار باشند، اکشن‌ها^{۲۰} اجرا می‌شن.

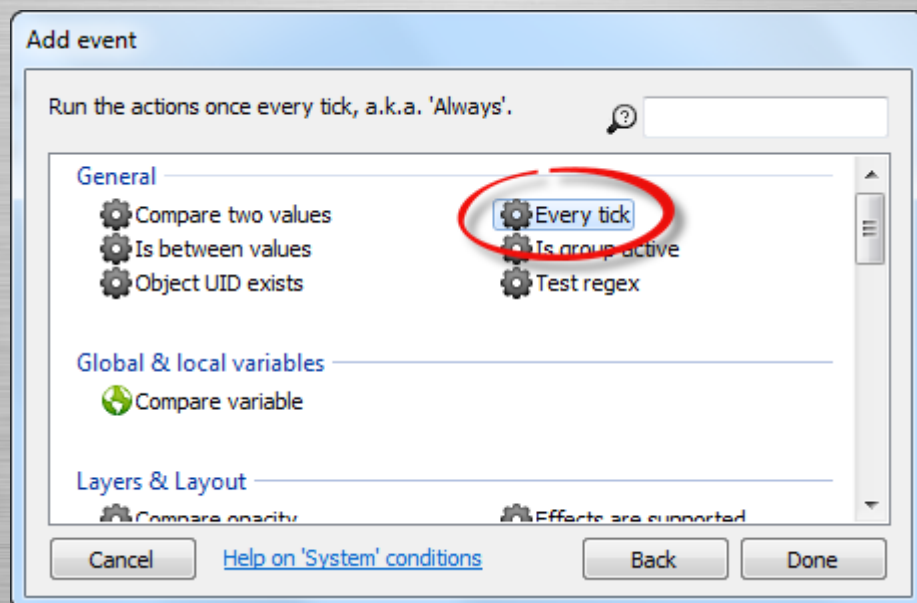
^{۱۹} جامد
^{۲۰} اکشن (Action): عمل

در اینجا ما می‌خواهیم پلیئر همیشه روی PlayerBox قرار داشته باشه. برای انجام این کار باید بگیم در هر لحظه پلیئر بیاد روی PlayerBox.

یه جای خالی ایونت شیت^{۲۱} دابل کلیک کنید تا یه ایونت جدید به وجود بیارید.



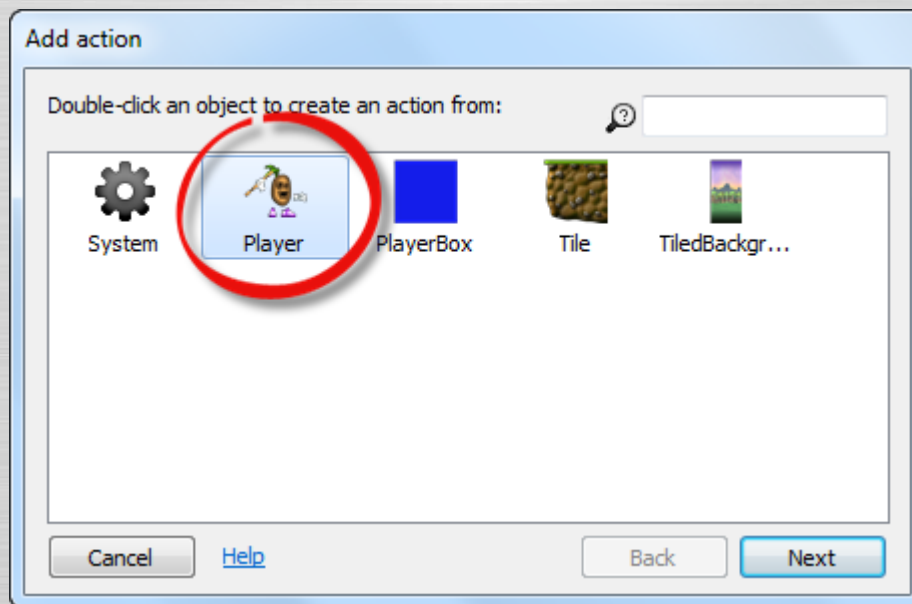
روی شیء System دابل کلیک کنید. بعد توی پنجره‌ی بعدی روی Every tick دابل کلیک کنید.



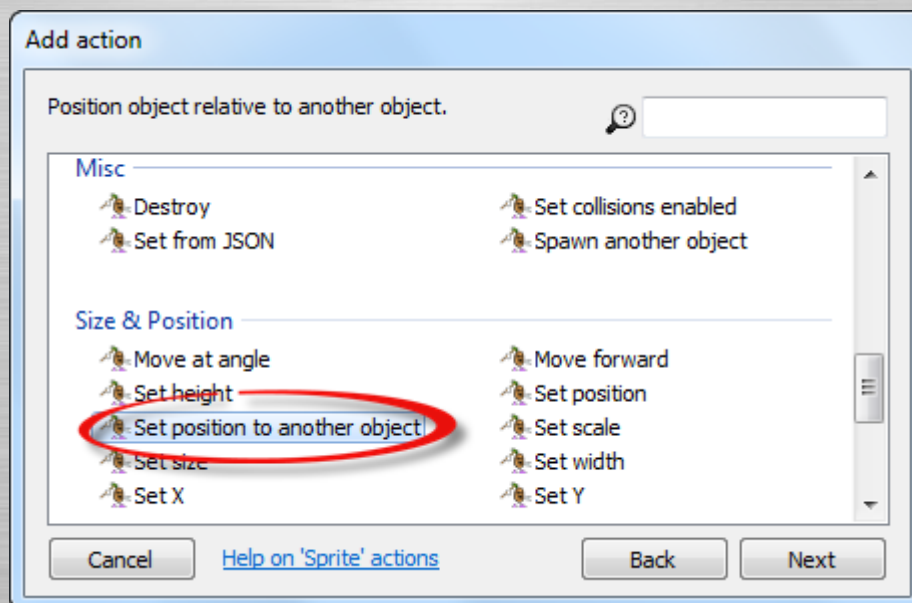
این ایونت رو که ایجاد کردیم اکشن‌هاش رو هر لحظه اجرا می‌کنه:



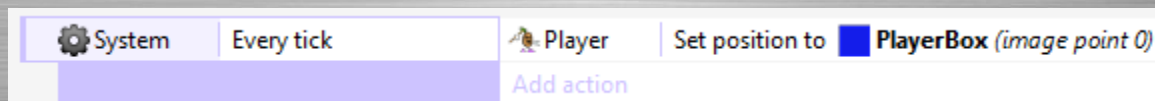
روی Add action کلیک کنید.



ما می‌خواهیم تعیین کنیم که مکان پلیر کجا باشه پس چون مربوط به پلیر می‌شه باید روی Player دابل کلیک کنید.



روی **Set position to another object** دابل کلیک کنید تا بگین که پلیر روی یک شیئی باشه. توی پنجره‌ی بعدی می‌گه روی کدوم شیء باشه؟ شما هم روی **<click to choose>** کلیک کنید و بعد شیء **PlayerBox** رو انتخاب کنید. عددی رو که جلوی **Image point** نوشته عوض نکنید و بذارین همون صفر بمونه (که این یعنی همیشه مبدأ پلیر روی مبدأ **PlayerBox** باشه). روی **Done** کلیک کنید. ایونت‌مون باید این شکلی شده باشه:



با این کار این احساس رو ایجاد می‌کنیم که پلیر داره راه می‌ره با اینکه پلیر همیشه روی شیء **PlayerBox** هست و این **PlayerBox** هست که داره راه می‌ره.

با کلیک روی دکمه‌ی **Play** (پلیر) که توی نوار عنوان قرار داره بازی رو اجرا کنید.



با کلیدهای جهت حرکت کنید و بپرید. شما اولین بازی پلتفرم خودتون رو راه انداختید و اجرا کردید!

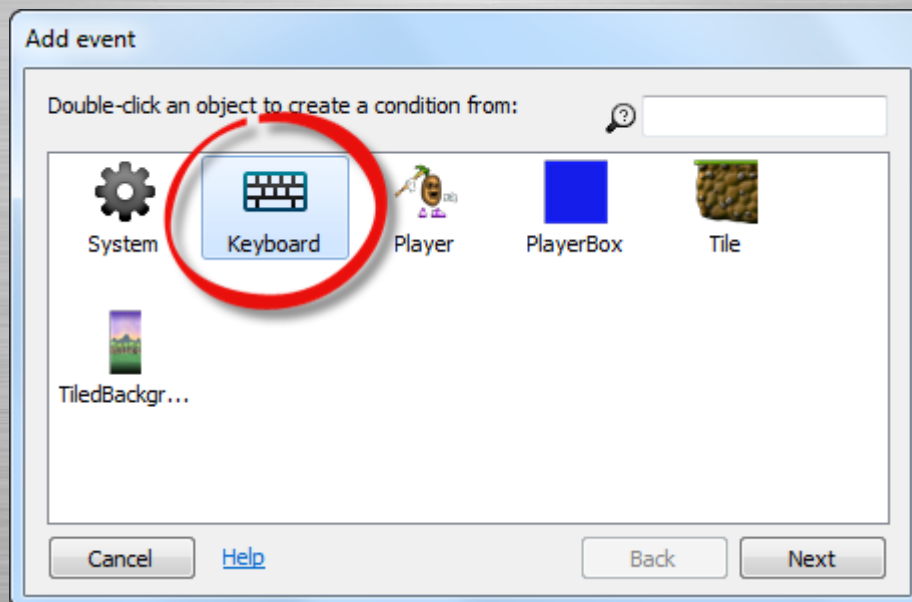
مشکلات

اگه یکم دور و بر حرکت کنید متوجه این مشکلات می‌شید:

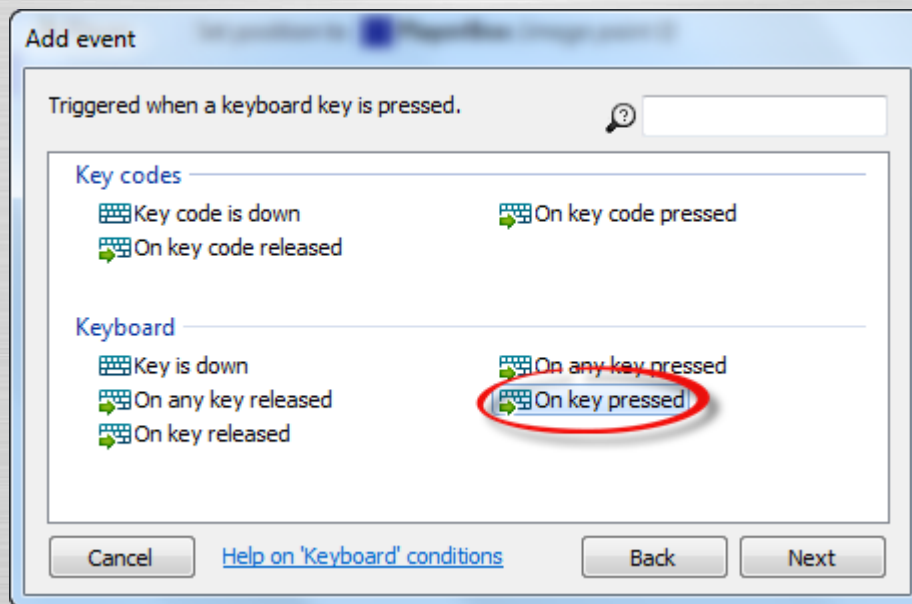
۱. موقع حرکت به سمت چپ پلیر هنوز صورتش به سمت راسته.
 ۲. هنوز هیچ انیمیشن دیگه‌ای به غیر از انیمیشن بی‌کار بودن وجود نداره.
- بیاین این مشکلات رو برطرف کنیم! اول از همه شماره‌ی ۱: بیاین تنظیم کنیم وقتی پلیر به چپ حرکت می‌کنه روش به سمت چپ باشه و برعکس.

با استفاده از سربرگ‌ها به لیوت برگردید. یه جای خالی دابل کلیک کنید تا دوباره یه شیء جدید بذاریم تو بازی، شیء Keyboard رو انتخاب کنید. این شیء فقط برای اینه که ایونت‌ها و اکشن‌های مربوط به کیبورد فعال بشن و لازم نیست اونو جایی قرار بدید. به جای این که یه انیمیشن جدید وارد کنیم که رویش به سمت چپ، خیلی راحت از اکشن Set mirrored استفاده می‌کنیم، این اکشن تصویر رو به صورت افقی قرینه می‌کنه و باعث می‌شه پلیر روشو به سمت چپ کنه به جای راست. به ایونت شیت برگردید، می‌خوایم بگیریم اگه کلید چپ فشار داده شد تصویر پلیر قرینه بشه.

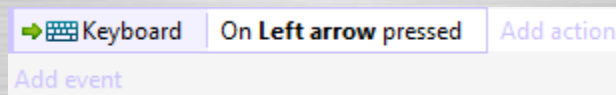
یه جای خالی دابل کلیک کنید یا روی Add event کلیک کنید تا یه ایونت جدید به وجود بیارید. حالا روی شیء Keyboard دابل کلیک کنید، چون فشار دادن کلید چپ به کیبورد مربوط می‌شه.



یه لیست از شرط‌های مربوط به کیبورد ظاهر می‌شه. روی On key pressed دابل کلیک کنید تا تنظیم کنید اگه کلیدی فشار داده بشه چه اتفاقی بیفته.



کنستراکت ۲ باید بدونه که با فشار دادن کدوم کلید اکشن‌ها رو اجرا کنه، پس در پنجره‌ی بعدی روی <click to choose> کلیک کنید و بعد کلید چپ رو فشار بدین و بعد روی OK کلیک کنید و بعد هم روی Done کلیک کنید.



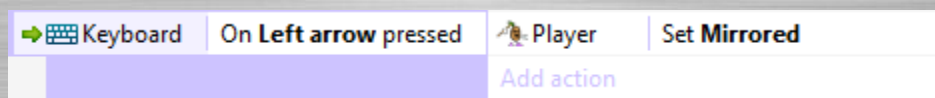
حالا می‌خوایم اکشن‌مون یعنی قرینه کردن افقی تصویر پلیئر رو اضافه کنیم. مثل قبل، باید اول شیء رو انتخاب کنید، بعد نوع شرط یا اکشن رو انتخاب کنید، بعد پارامترهای اضافی رو وارد کنید. پس روی Add action که سمت راست ایونت قرار داره کلیک کنید (حواستون باشه اشتباهی روی Add event کلیک نکنید!)

بعد روی Player دابل کلیک کنید.

بعد روی Set mirrored دابل کلیک کنید.

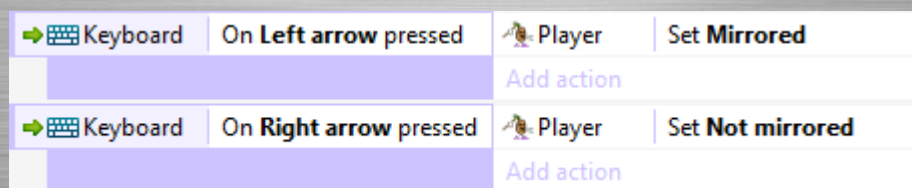
State که به طور پیش‌فرض روی Mirrored تنظیم هست رو تغییر ندین و روی Done کلیک کنید.

حالا باید اینو داشته باشید:



حالا یه ایونت دیگه اضافه کنید - یه جای خالی کلیک راست کنید یا روی Add event کلیک کنید.

حالا دوباره همه‌ی کارایی رو که الان گفتیم انجام بدین ولی این بار به جای کلید چپ کلید راست کیبورد رو فشار بدین تا ایونت‌تون بشه On right arrow key pressed و در آخرش هم State رو به Not mirrored تنظیم کنید. آخرش باید این شکلی بشه:



حالا وقتی ما چپ رو فشار می‌دیم تصویر پلیر به صورت افقی قرینه می‌شه یعنی روشو می‌کنه سمت چپ و وقتی راست رو فشار می‌دیم دوباره تصویر پلیر از حالت قرینه بودن خارج می‌شه یعنی روشو دوباره می‌کنه سمت راست. بازی رو اجرا کنید و امتحانش کنید!

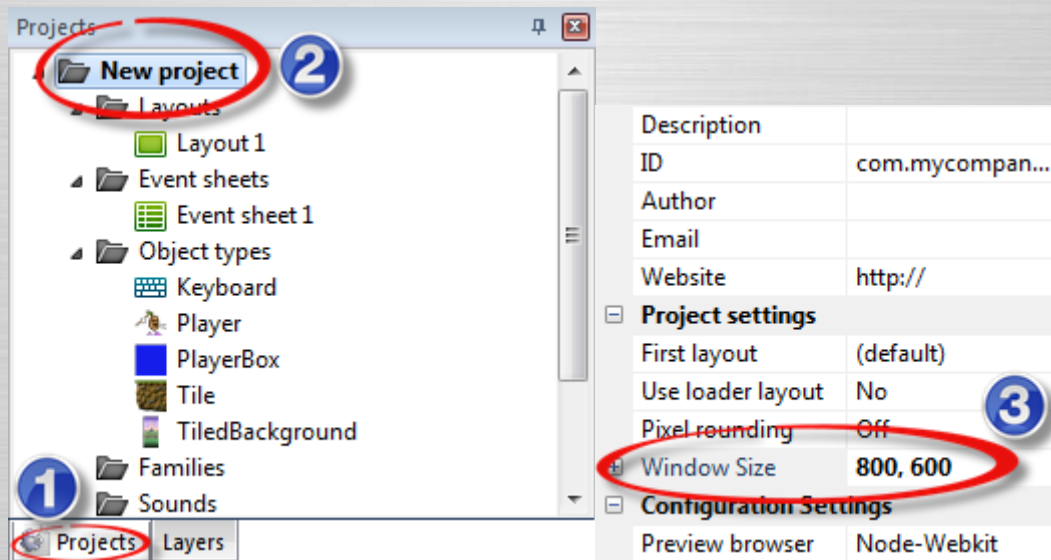
اضافه کردن انیمیشن‌های بیشتر

بیاین بقیه‌ی انیمیشن‌های پلیر رو بهش اضافه کنیم. اول چند تا زمین دیگه می‌ذاریم تا پلیر یه جایی باشه که بتونه روش بپره تا راحت بتونیم ببینیم انیمیشن‌هامون چی جور می‌کنند.

به لیوت برگردید تا بتونید پلیر و تایل‌هایی که در اصل زمین مون هستند رو ببینید. هر سه تا تایل رو با کلیک و درگ کردن انتخاب کنید. حالا کنترل رو نگه دارید و تایل‌ها رو درگ کنید تا از روشون کپی بشه. چند بار این کارو بکنید تا چند تا تیکه زمین باشه که پلیر بتونه روشون بپره. ممکنه اگه به عقب زوم کنید تسلط بیشتری داشته باشید و راحت تر این کار رو انجام بدید.



برای اینکه نمای بهتری داشته باشیم بیاین اندازه‌ی پنجره‌ی بازی رو هم کمی بزرگتر کنیم. در نوار پروژه روی اسم پروژه کلیک کنید تا تنظیمات پروژه نمایش داده بشه. اندازه‌ی پنجره رو به ۸۰۰×۶۰۰ تغییر بدید.



حالا که به جایی برای پریدن داریم، پنجره‌ی بازی مون هم برای اینکه قسمت بیشتری رو ببینیم بزرگتر شده، بریم سراغ اضافه کردن بقیه‌ی انیمیشن‌های پلیر.

یادتون باشه که ما تنظیم کردیم وقتی چپ رو فشار بدیم تصویر پلیر قرینه بشه. پس فقط باید انیمیشن‌هایی رو وارد کنیم که روشن به سمت راسته.

روی انیمیشن که توی لیوت قرار داره دابل کلیک کنید تا ویرایشگر تصویر باز بشه، ما می‌خوایم این چهار تا انیمیشن رو وارد کنیم:

۱. بی‌کار می‌خواد بدوه – Idle to run (tween)

۲. دویدن – Run

۳. پرش در هنگام دویدن – Jump from run

۴. پرش در هنگامی که پلیر سر جای خود ایستاده باشه – Jump from stand

برای اضافه کردن هر کدوم از این انیمیشن‌ها به این ترتیب عمل کنید:

۱. در پنجره‌ی Animations کلیک راست کنید (این پنجره به طور پیش‌فرض، سمت راست عکس شناوره و توش انیمیشن Idle لیست شده) و روی Add animation کلیک کنید.

۲. اسم انیمیشن رو وارد کنید. ممکنه بخواید اسم‌های ساده‌تری انتخاب کنید که راحت تر حفظشون کنید مثل IdleToRun.

۳. روی انیمیشن جدیدی که ایجاد کردید کلیک کنید تا توی پنجره دیده بشه که فقط باید به فریم خالی باشه.

۴. در پنجره‌ی Animation frames کلیک راست کنید (این پنجره به طور پیش‌فرض، زیر تصویر قرار داره و توش فریم‌های انیمیشن لیست شده) و Import sprite strip... رو انتخاب کنید.

۵. اسپریت استریپ رو انتخاب کنید. توی پک موجود، هر فریم توی یک عکس PNG به صورت جدا هم وجود داره. از این راه هم می‌تونید فریم‌ها رو بیارید تو اما استفاده از اسپریت استریپ سریع‌تره، پس اسپریت استریپ رو انتخاب کنید.

۶. نگاه کنید که کنستراکت ۲ تعداد فریم‌ها رو درست تشخیص داده باشه، بعد روی OK کلیک کنید.

۷. فریم اول که خالیه و ما بهش احتیاجی نداریم رو حذف کنید.

۸. توی فریم اول، نقطه‌ی مبدأ رو بذارید روی پاهای پلیر. سعی کنید مکان نقطه‌ی مبدأ با نقطه‌ی مبدأ انیمیشن بی‌کار مطابق باشه. همون طور که برای انیمیشن بی‌کار نقطه‌ی مبدأ رو روی همه‌ی فریم‌ها اعمال کردید این کار را این‌جا هم انجام بدید. (تو پنجره‌ی Image Points روی Origin کلیک راست کنید و Apply to whole animation رو انتخاب کنید).

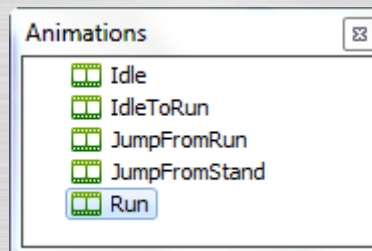
۹. شیف‌ت رو نگه دارید و روی دکمه‌ی Crop کلیک کنید تا یک دفعه‌ی قسمت‌های خالی و اضافی همه‌ی فریم‌ها حذف بشن.

۱۰. توی نوار ویژگی‌ها، به هر انیمیشن ویژگی‌های زیر رو بدین:

Speed: ۲۲ برای JumpFromStand، ۱۵ برای بقیه

Loop: Yes برای Run، و No برای بقیه

این کارها رو برای همه‌ی انیمیشن‌ها انجام بدین، پلیر آخر کار باید پنج تا انیمیشن داشته باشه.



ویرایشگر تصویر رو ببندید. حالا باید بگیم در وقت مناسب انیمیشن مناسب نشون داده بشه. رفتار Platform ایونت‌های لازم برای این

کار رو در اختیارمون می‌ذاره، ایونت‌هایی که وقتی اجرا می‌شن انیمیشن پلیر باید عوض بشه. برای پلیر هم فقط از اکشن Set

animation استفاده می‌کنیم تا انیمیشنش رو عوض کنیم. حواستون باشه که رفتار Platform روی شیء PlayerBox هست، و

بعد ما می‌خوایم انیمیشن شیء Player که دیده می‌شه رو عوض کنیم.

انیمیشن‌ها از الگوی زیر پیروی می‌کنند یعنی در این موقعیت‌ها انیمیشن‌هاشون عوض می‌شه:

موقع توقف: Idle

شروع به حرکت: IdleToRun

وقتی IdleToRun تموم شد: Run

راه اندازی این‌ها هم با ایونت‌ها کار ساده‌ای هست.

راه انداختن ایونت‌های انیمیشن

امیدوارم تا حالا با مراحل ساخت ایونت آشنا شده باشید.

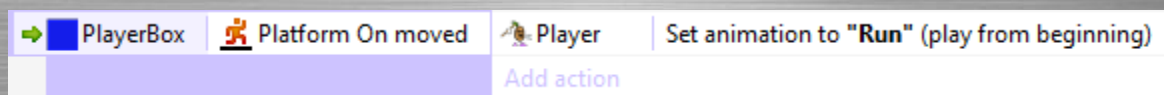
۱. شیء رو انتخاب کنید.

۲. شرط یا اکشن رو انتخاب کنید.

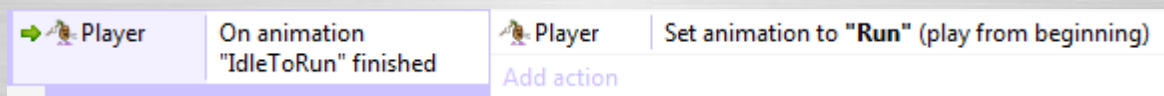
۳. اختیاری: اگه اکشن یا شرط پارامتر داشت اون پارامتر رو وارد کنید.

به ایونت شیت برگردین. اول از همه، بیاین با استفاده از ایونت On moved که مال رفتار پلت فرم شیء PlayerBox هست، تنظیم

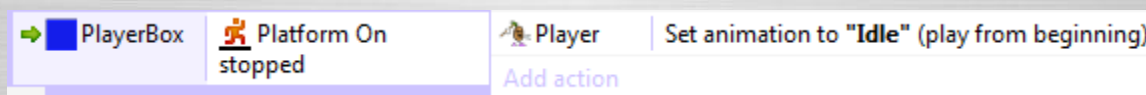
کنیم که موقع شروع حرکت، انیمیشن پلیر به IdleToRun تغییر کنه:



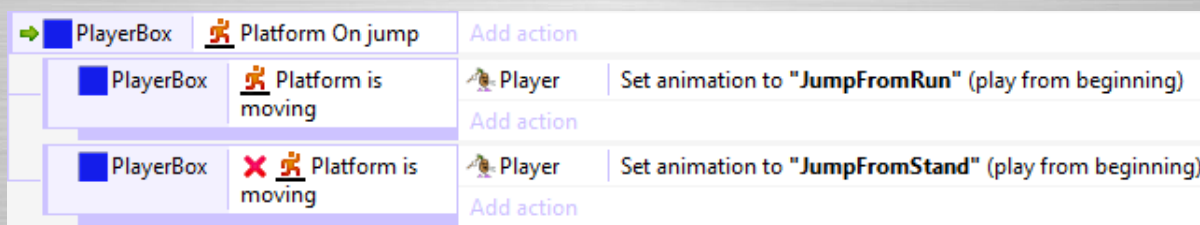
می‌خوایم بعد از این که انیمیشن IdleToRun تموم شد، انیمیشن به Run تغییر کند. ما می‌تونیم این کار رو با استفاده از ایونت On finished که مال شیء Player هست انجام بدیم:



اگه پلیر بدوه و بعد بایسته باید انیمیشن به Idle تغییر کند. این کارو می‌تونیم با ایونت On stopped که مال شیء PlayerBox هست انجام بدیم:



بازی رو اجرا کنید. الان پلیر وقتی می‌دوه انیمیشنش به دویدن تغییر می‌کنه و وقتی می‌ایسته انیمیشنش به بی‌کار تغییر می‌کنه. اما انیمیشن‌های پریدن هنوز اجرا نمی‌شن. به خاطر این که ما دو انیمیشن برای پریدن داریم اول باید بررسی کنیم که پلیر ایستاده یا نه تا بتونیم انیمیشن مناسب رو اجرا کنیم. این کار رو با ایونت زیر می‌تونید انجام بدین:



در اینجا از ساب-ایونت استفاده کردیم، ساب-ایونت‌ها نسبت به ایونت‌های معمولی یکم بیشتر از سمت چپ صفحه فاصله دارند و اینجا زیر ایونت Platform On jump دیده می‌شن. ساب-ایونت‌ها بعد از اجرا شدن ایونت مادر (همون ایونتی که این ساب-ایونت‌ها زیر مجموعه‌ش هستن) اجرا می‌شن. الگوی ساب ایونت این طوره:

وقتی شیء دارای رفتار Platform می‌پرد، اگر در حال حرکت بود، انیمیشن به JumpToRun تنظیم بشه وگرنه به JumpFromStand

برای ایجاد ساب-ایونت روی ایونت مادر کلیک راست کنید و Add -> Add sub-event رو انتخاب کنید، یا ایونت مادر رو انتخاب کنید و کلید S رو فشار بدین.

برای اینکه ایونت بالا رو ایجاد کنید باید یک شرط رو برعکس کنید (همونی که کنارش علامت ضربدر داره). وقتی یه شرط رو برعکس می‌کنیم یعنی دقیقاً برعکس شرطی که الان هست انجام بده. در اینجا برعکس شدن شرط Platform is moving یعنی Platform is NOT moving. برای برعکس کردن یک شرط ابتدا معمولی شرط رو وارد کنید و بعد روش کلیک راست کنید و Invert رو انتخاب کنید. همچنین می‌تونید انتخابش کنید و کلید I رو فشار بدین.

فقط یه کار دیگه مونده: پلیر وقتی می‌پره توی همون انیمیشن پرشش باقی می‌مونه حتی وقتی میفته روی زمین. برای رفع این مشکل از ایونت زیر استفاده می‌کنیم که خیلی شبیه ایونت قبلی می‌مونه. در ایونت On land که به معنی روی زمین بودن است، اگر پلیر در حال حرکت بود انیمیشن به دویدن (Run) وگرنه به بی‌کار (Idle) تغییر کند.

→ PlayerBox	Platform On landed	Add action
PlayerBox	Platform is moving	Player Set animation to "Run" (play from beginning)
		Add action
PlayerBox	Platform is moving	Player Set animation to "Idle" (play from beginning)
		Add action

وقتی همه‌ی این کارها رو کردید بازی رو اجرا کنید. تبریک، حالا یه کاراکتر پلیر دارین که به طور کامل همه‌ی انیمیشن‌هاش درسته! در حالت ایستادن یا دویدن بپرید تا تغییر انیمیشن‌ها رو ببینید و ببینید وقتی که روی زمین می‌افتید اگه ایستاده باشید انیمیشن بی‌کار و گرنه انیمیشن دویدن اجرا می‌شه.

حالا وقتشه که یه دشمن بسازیم!

اضافه کردن دشمن

امیدوارم تا حالا با مراحل اضافه کردن اسپرایت و انیمیشن آشنا شده باشید. به خاطر همین برای این‌که آموزش زیاد طولانی نشه دوباره تکرار نمی‌کنیم. بیاین یه اسپرایت جدید برای دشمن وارد کنیم.

۱. یه شیء اسپرایت جدید به لیوت اضافه کنید.
۲. اسپرایت استریپ Enemies\Snail رو انتخاب کنید.
۳. اولین فریم انیمیشن رو که خالیه حذف کنید.
۴. تنظیم کنید وقتی انیمیشن تموم شد دوباره از اول پخش بشه، و سرعتش هم ۱۵ باشه.
۵. نقطه‌ی مبدأ رو بذارین پایین حلزون و این نقطه رو به همه‌ی انیمیشن اعمال کنید.
۶. قسمت‌های خالی اضافی همه‌ی فریم‌ها رو حذف کنید (shift + crop)
۷. ویرایشگر تصویر رو ببندید و اسم شیء رو SnailEnemy بذارید.
۸. همون طور که می‌بینید اون رو روی زمین قرار بدین.



حالا می‌خوایم از الگوی زیر پیروی کنیم:

- اگه پلیر از بغل به حلزون برخورد کنه، چشمک می‌زنه^{۲۲}.
- اگه پلیر از بالا روی حلزون بیره، حلزون می‌میره.

^{۲۲} روشن و خاموش می‌شه

برای چشمک زدن پلیمر، پلیمر رو انتخاب کنید و بهش رفتار Flash بدین. حواستون باشه پلیمر واقعی رو انتخاب کنید، نه شیء PlayerBox رو (چون ما می‌خواهیم اون پلیمری که دیده می‌شه چشمک بزنه) برای این کار از اکشن‌های رفتار Flash استفاده می‌کنیم. به نمای ایونت شیت بروید، و ایونتی جدید اضافه کنید:

PlayerBox -> On collision with another object -> SnailEnemy

وقتی با هر زاویه‌ای به حلزون می‌خوریم این ایونت اجرا می‌شه. ما می‌تونیم برای اینکه آیا پلیمر از بالا روی حلزون پریده یا از بغل بهش برخورد کرده از ساب-ایونتها استفاده کنیم. بیاین اول تست کنیم که آیا پلیمر بالای سر حلزونه. یه ساب-ایونت به ایونت برخورد اضافه کنید:

PlayerBox -> Is falling

ما باید بررسی کنیم که آیا پلیمر واقعاً بالای سر دشمن قرار داره. این کار از کشته شدن تصادفی حلزون جلوگیری می‌کنه، مثلاً اگر از گوشه‌ای پایین بیفتیم حلزون دیگه نمی‌میره. روی شرط Is falling کلیک راست کنید و گزینه‌ی Add another condition انتخاب کنید. حواستون باشه برای اجرا شدن یک ایونت باید همه‌ی شرط‌هاش درست باشن. این شرط رو اضافه کنید:

PlayerBox -> Compare Y -> Less than, SnailEnemy.Y

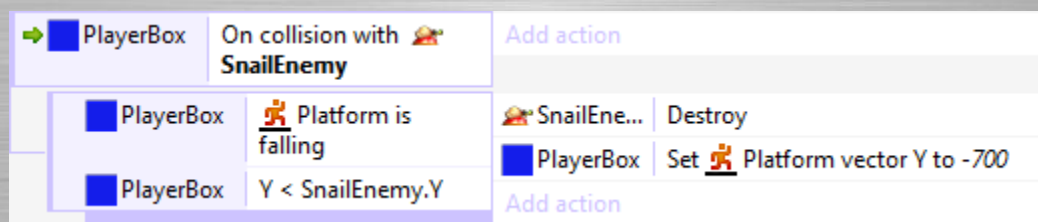
محور Y ها به سمت پایین افزایش پیدا می‌کنه (یعنی همین طور که به طرف پایین می‌ریم Y ها بیشتر می‌شن دقیقاً برعکس ریاضی)، بنابراین اگه Y پلیمر کمتر از مال حلزون باشه، پلیمر بالاتر از اون قرار داره. در این ایونت، این اکشن رو اضافه کنید تا حلزون نابود بشه:

SnailEnemy -> Destroy

ما می‌تونیم با افزودن اکشن دیگری باعث شویم پلیمر به سمت بالا نیز بجهد:

PlayerBox -> Set vector Y -> -700

Set vector Y اساساً فقط سرعت عمودی حرکت پلت فرم رو تنظیم می‌کنه؛ تنظیمش به عددی منفی آن را به سمت بالا تنظیم می‌کند (چون محور Y به سمت پایین افزایش می‌یابد)، و ۷۰۰ کمی از قدرت پرش پلیمر یعنی از ۱۱۰۰ کمتره. پس این کار باعث می‌شه پلیمر بجهد که این جهیدن مثل پرشی ضعیف می‌مونه.



هنوز تموم نشده: روی حاشیه‌ی ایونت Is falling کلیک راست کنید (حاشیه همون جاییه که سمت چپ آیکون PlayerBox هست) و Else -> Add را انتخاب کنید. Else یک شرط مخصوصه که اگر ایونت قبلی اجرا نشه اون اجرا می‌شه. پس اگر ما به حلزون برخورد کنیم اما روی آن نپریده باشیم این ایونت اجرا خواهد شد - یعنی از بغل به حلزون برخورد کردیم. در این ایونت که می‌خواهیم پلیمر آسیب ببیند، این اکشن را اضافه کنید:

(مقادیر پیش فرض را تغییر ندهید و روی Done کلیک کنید) Player -> Flash ->

یادتون باشه که رفتار Flash در شیء Player است نه PlayerBox.

→ PlayerBox	On collision with SnailEnemy	Add action
PlayerBox	Platform is falling	SnailEne... Destroy
PlayerBox	Y < SnailEnemy.Y	PlayerBox Set Platform vector Y to -700
		Add action
System	Else	Player Flash: Flash 0.1 on 0.1 off for 1.0 seconds
		Add action

خوب، چون پلیئر هیچ وقت نمی میره، فقط چشمک می زنه. اما تونستین تشخیص بدیم که آیا از بالا روی آن پریدیم یا از کنار به آن برخورد کردیم. در این آموزش تمام ویژگی های کنستراکت ۲ گفته نشده اند، اما الان فقط می گیم کجا افکت های مردن رو ایجاد کنید (سعی کنید وقتی حلزون نابود شد یک اسپرایت poof روی آن ایجاد کنید، با استفاده از اکشن Spawn object حلزون) و کجا از سلامتی اش کم کنیم (در ایونتی که باعث می شود پلیئر چشمک بزند - می توانید در مورد سلامتی با استفاده از متغیرهای اینستنس یه کارهایی بکنید).

بیا بعد عقب و جلو رفتن حلزون روی زمین رو تنظیم کنیم.

حرکت دشمن

مشکل اصلی حرکت حلزون اینه که چگونه تشخیص دهد که به لبه ی پرتگاه رسیده است. ساده ترین راه انجام این کار نشان گذارهای نامرئی پرتگاه هاست. این ها فقط اسپرایت های نامرئی ای هستند که وقتی حلزون به آن ها برخورد کند جهت حرکتش را قرینه می کنند.

ما می توانیم برای حلزون نیز دوباره از رفتار Platform استفاده کنیم. و این کار خوبی است زیرا:

- می تواند به چپ و راست حرکت کند

- می تواند از سطح شیب دار پایین و بالا برود درست همان طور که پلیئر می تواند

- از پرتگاه ها پایین می افتد اگر بخواهید

- اگر بخواهید دشمنی بسازید که بتواند بپرد، می توانید با اکشن simulate control تنظیم کنید که به صورت خودکار بپرد.

به اسپرایت SnailEnemy رفتار Platform را اضافه کنید. به خاطر اینکه ما از انیمیشن های زیادی روی این شیء استفاده نکردیم، می توانیم با استفاده از رفتار پلت فرم شیء به طور مستقیم بدون شیئی نامرئی رفتار Platform رو مستقیم به خودش بدهیم. توجه داشته باشید اگر بازی پلت فرم دیگری با دشمن هایی که انیمیشن هایی زیاد دارند بسازید، باید از همان تکنیک اسپرایت نامرئی که برای پلیئر استفاده کردیم استفاده کنید.

ما نمی خواهیم پلیئر دشمن را کنترل کند، ما می خواهیم به طور خودکار کنترل شود. ما می توانیم این کار را با تنظیم Default controls به No در نوار ویژگی ها و بعد با استفاده از اکشن Simulate control انجام دهیم. چون حلزون ها نسبتاً آهسته حرکت می کنند، Max speed را به ۵۰ تنظیم کنید و Acceleration و Deceleration را به ۱۰۰ تنظیم کنید!

ما همچنین به نشان گذارهای پرتگاه نیاز داریم. اسپرایتی جدید اضافه کنید و فقط یک مستطیل رنگی بکشید.

آن را EdgeMarker بنامید. سائز شیء را به 40×40 تغییر دهید و Initial visibility آن را به Invisible تغییر دهید تا نتوانیم آن را ببینیم. در آخر هر یک از زمین‌ها یکی از آن‌ها را قرار دهید مثل این: (فراموش نکنید با نگه داشتن کلید کنترل و درگ کردن می‌توانید اینستنس‌هایی^{۲۳} جدید از شیئت‌ان ایجاد کنید)



حلزون باید بداند که الان در چه راهی حرکت می‌کند، چپ یا راست. می‌توانیم این کار را با متغیرهای اینستنس (instance variables) انجام دهیم. این‌ها اعداد یا متن‌هایی هستند که در هر اینستنس شیء ذخیره می‌شوند. اگر چند حلزون داشته باشیم، این متغیرها برای هر کدام از آن‌ها به طور مجزاً ذخیره می‌شوند. این به آن‌ها اجازه می‌دهد که سلامتی، جهت فعلی و چیزهای دیگر منحصر به فرد داشته باشند. ایجاد متغیر اینستنسی که موقعیت فعلی دشمن را نگه می‌دارد تکنیکی ساده برای کنترل شدن خودکار دشمن‌هاست. مثلاً می‌توانید با این روش تنظیم کنید دشمن فرار کند، پلیر را تعقیب کند یا بی‌کار باشد. در این مورد ما فقط می‌خواهیم حلزون به چپ و راست حرکت کند، و انجام این کار با این روش خوبه.

شیء SnailEnemy را انتخاب کنید. در نوار ویژگی‌ها، روی Instance variables که به رنگ آبی هست کلیک کنید.

Common	
Layer	Main
Angle	0
Opacity	100
Position	1229, 606
Size	101, 67
Instance variables	
Add / edit	Instance variables
Behaviors	
Platform	
Max speed	50
Acceleration	100
Deceleration	100

پنجره‌ای که در آن تمام متغیرهای اینستنس لیست شده ظاهر می‌شود. روی آیکن add کلیک کنید تا یکی جدید اضافه کنید. نام آن را به action تغییر دهید، نوع را به text (متن)، و مقدار اولیه (initial value) را به right تنظیم کنید (برای حرکت به راست).

روی OK کلیک کنید و پنجره‌ی متغیرهای اینستنس را ببندید. به ایونت شیت برگردید.

ما می‌خواهیم طبق الگوی زیر پیش ببریم:

^{۲۳} اینستنس (Instance): وقتی یک شیء رو وارد بازی می‌کنیم از روش چند نمونه در لیوت کپی می‌کنیم یا توی بازی حین بازی کردن ایجاد می‌کنیم مثل گلوله. به هر کدام از نمونه‌های یک شیء در لیوت یک اینستنس گفته می‌شه.

- اگر مقدار action برابر right باشد، برای اینکه حلزون به راست حرکت کند نگه داشته شدن کلید راست شبیه‌سازی شود.

- اگر مقدار action برابر left باشد، برای اینکه حلزون به چپ حرکت کند نگه داشتن کلید چپ شبیه‌سازی شود.

درضمن:

- وقتی حلزون به نشان گذارهای پرتگاه برخورد کرد، مقدار action قرینه شود (اگر left بود right شود و برعکس)

ما با این دو ایونت می‌توانیم حرکت حلزون را راه اندازی کنیم.

Event: SnailEnemy -> Compare instance variable -> action equal to "right"

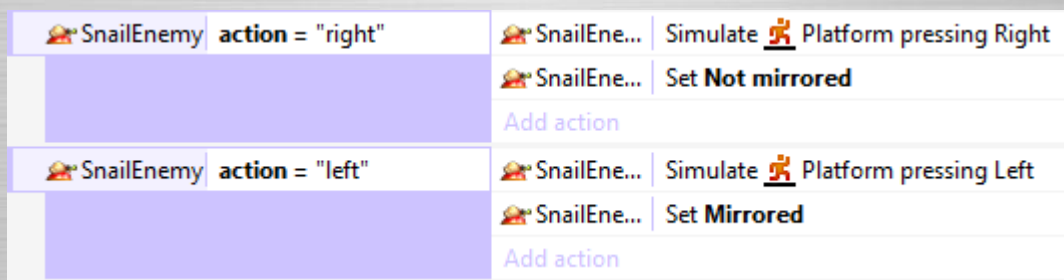
(برای این که نشان دهید متن است باید از دابل کوتیشن (" ") استفاده کنید)

Action: SnailEnemy -> Simulate control -> Right

باید درست همین کار را برای چپ هم انجام دهید. درضمن ما می‌خواهیم روی حلزون به سمت چپ و راست تغییر کند مثل پلیر. بنابراین

در ایونت مربوط به چپ SnailEnemy -> Set mirrored را اضافه کنید، و SnailEnemy -> Set not mirrored را در

ایونت مربوط به راست اضافه کنید. باید در آخر این شکلی شده باشد:



حالا برای این که وقتی حلزون به پرتگاه رسید جهتش قرینه شود:

Event: SnailEnemy -> On collision with another object -> EdgeMarker

Sub-event: SnailEnemy -> Compare instance variable -> action equal to "right"

Action: SnailEnemy -> Set value -> action to "left"

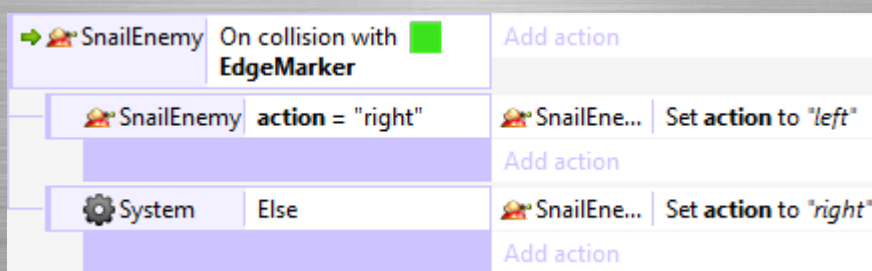
Sub-event: Else

Action: SnailEnemy -> Set value -> action to "right "

اینجا استفاده از Else مهم است، چون ایونت‌ها از بالا به پایین اجرا می‌شوند. اگر به جای else (به معنی وگرنه) بگوییم اگر متغیر

action برابر left بود اکشن‌ها اجرا شوند، در هر صورت برابر left است چون ایونت قبلی آن را برابر left قرار می‌دهد. بنابراین در هر

صورت به راست حرکت می‌کند. با استفاده از else، ما از اجرای ایونت دوم در صورتی که اولی درست باشد جلوگیری می‌کنیم.



بازی‌تان را اجرا کنید. می‌بینید که حلزون دارد به عقب و جلو حرکت می‌کند. این باعث می‌شود که پریدن روی حلزون کمی سخت‌تر شود! این یک سیستم هوش مصنوعی بسیار ابتدایی است، اما تصور کنید که می‌توانید دشمنانی باهوش‌تر که حرکتشان با ایونت‌های بیشتر کنترل می‌شود بسازید. ممکن است حتی به آن‌ها اجازه دهید از پرتگاه بیفتند، یا از نشان‌گذارهای دیگری استفاده کنید تا با رسیدن به لبه‌ها بپرد تا این‌طور به نظر برسد که دشمن می‌داند باید از لبه‌ها بپرد.

یه زمین بسازید و دو تا حلزون روش بذارید. توجه کنید که آن‌ها شخصاً خودشان را کنترل می‌کنند، چون هر یک از آن‌ها متغیر اینستنس action مربوط به خود را دارند که وضعیتشان را مشخص می‌کند. امیدوارم ببینید که چه قدر متغیرهای اینستنس برای حرکت به طور مستقل مهم هستند. همه‌ی حلزون‌ها مجبور نیستند دقیقاً همان کاری را انجام دهند که دیگری انجام می‌دهد.



اشیاء جامپ-ثرو^{۲۴}

اگر در اطراف پلیئر تعدادی تایل قرار دهید، ممکن است متوجه شوید که اگر بخواهید از زیر یک تایل به رویش بپرید سرتان به آن برخورد می‌کند و پایین می‌افتید. اگر بخواهید تایل‌هایی داشته باشید که بتوانید از زیرشان به رویشان بپرید چه؟

این کاری است که رفتار Jump-thru انجام می‌دهد. مثل رفتار Solid، شیء دارای رفتار Platform می‌تواند روی آن بایستد، با این تفاوت که این بار پلیئر می‌تواند از زیر آن هم به رویش بپرد.

بیا یک نسخه‌ای از شیء تایل بسازیم که رفتار Jump-thru داشته باشد. روی شیء Tile کلیک راست کنید و Clone object type را انتخاب کنید. با این کار یک شیء جدا می‌سازید، به اسم Tile2، (از توی نوار پروژه درگ کنیدش توی لیوت) که می‌تواند رفتارهای متفاوتی داشته باشد. رفتار Solid را حذف کنید و رفتار Jump-thru را اضافه کنید.

از تایل شماره‌ی ۱۳۴ و ۱۳۵ استفاده کنید تا یک جامپ-ثرو مثل زیر بسازید:



بازی را اجرا کنید. توجه کنید که چگونه می‌توانید از زیر روی آن بپرید.

امیدوارم تا به حال به اندازه‌ی کافی در مورد طراحی یک مرحله‌ی کامل اطلاعات به دست آورده باشید! اینجا مرحله‌ای را می‌بینید که به عقب زوم شده است و من آن را به سرعت در حین نوشتن آموزش طراحی کردم. حواستان که در دو طرف زمین‌هایی که می‌گذارید از نشان‌گذارها استفاده کنید تا حلزون‌ها از رویشان نیفتند.



پارالاکس

می‌خواهیم یک افکت پارالاکس اضافه کنیم تا پس‌زمینه دورتر به نظر برسد. در نوار لایه‌ها لایه‌ی Background را انتخاب کنید. در نوار ویژگی‌ها، ویژگی Parallax را به ۵۰، ۵۰ تنظیم کنید (که یعنی نصف سرعت پیمایش). می‌بینید که چگونه پس‌زمینه آهسته‌تر حرکت می‌کند و باعث می‌شود احساس دور بودن به انسان بدهد. استفاده از چندین لایه که همگی دارای پارالاکس‌هایی با مقادیر متفاوت هستند به بازی دو بعدی تقریباً احساس سه بعدی بودن می‌دهد.

جمع بندی و پایان

در این آموزش چیزهای زیادی گفته شد. ممکنه بخواید دوباره این را بخوانید. اینجا تعدادی از نکات کلیدی آموزش رو دوباره تکرار می‌کنم:

- برای رفتار Platform پلیئر، همیشه حرکت را به مستطیلی نامرئی که انیمیشنی ندارد بدهید. اسپرایت پلیئر که دارای انیمیشن است و دیده می‌شود را روی آن مستطیل قرار دهید. این کار از اشکالاتی که انیمیشن‌ها برای حرکت پلت فرم ایجاد می‌کنند جلوگیری می‌کند.
- مرحله‌ها می‌توانند از تایل‌ها ساخته شوند.

- اکشن Set mirrored شما را از ساختن کپی‌هایی از تمام انیمیشن‌ها که قرینه‌ی هم هستند بی‌نیاز می‌کند.
- گاهی اوقات برای اینکه انیمیشن‌ها را وارد کنیم چند کار باید انجام دهیم، تکرار شدن از اول و سرعت درست را تنظیم کنیم، محل صحیح نقطه‌ی مبدأ را تعیین کنیم، و قسمت‌های اضافی همه‌ی فریم‌ها را برش دهیم. حتی شاید بخواهید کالیژن ماسک‌ها را تغییر دهید، اگرچه این جزء آموزش ما نبود.
- دشمن‌ها نیز با رفتار پلت فرم قابل کنترل هستند. Default Controls را به No تنظیم کنید و برای کنترل خودکار حرکت از اکشن Simulate control استفاده کنید.
- شما می‌توانید تشخیص دهید که آیا پلیر از بالا روی دشمن پریده‌است و یا در حال راه رفتن به آن برخورد نموده است. با بررسی این که آیا پلیر هنگام برخورد هم در حال سقوط بوده و هم در محور Yها بالای دشمن قرار داشته است می‌توانیم بفهمیم پلیر بالای سر حلزون بوده در غیر این صورت (با استفاده از Else)، در حال راه رفتن به دشمن برخورد کرده است، و باید آسیب ببیند.
- متغیرهای اینستنس می‌توانند اعداد یا متن‌هایی را برای هر اینستنس شیء به صورت جداگانه در خود ذخیره کنند. این به کنترل اشیاء به صورت جداگانه کمک می‌کند، که برای هوش مصنوعی خیلی به درد می‌خورد.
- نشان‌گذارهای پرتگاه راهی سریع و آسان برای این هستند که دشمن‌ها روی زمین به عقب و جلو حرکت کنند.
- جامپ-ثروها (Jump-thru) پلت فرم‌هایی هستند که می‌توانید از زیرشان به رویشان بپرید.
- پارالاکس افکتی جذاب و ساده است که به بازی‌های پلت فرم اضافه می‌شود.

واضح است که ما یک بازی پلت فرم کامل نساخته ایم. اما، این آموزش اغلب مشکلات، و چیزهای مهمی را که هر سازنده‌ی بازی‌های پلت فرم در سطح حرفه‌ای باید بداند گفته است. در اینجا امیدوارم که شما ایده‌ای داشته باشید که چگونه بازی پلت فرم بسازید. وقت زیادی می‌برد که با ابزار پیچیده‌ی توسعه در کنستراکت ۲ آشنا شوید. اما، آزمون و خطا می‌تواند یک سرگرمی باشد، و چیزهای زیادی به شما بیاموزد! بازی پلت فرمتان مبارک!

به مناسبت ولادت پیامبر اکرم، حضرت محمد (ص)

مجتبی قاسم زاده تهرانی

moitaba.mihanblog.com

زمستان ۱۳۹۲