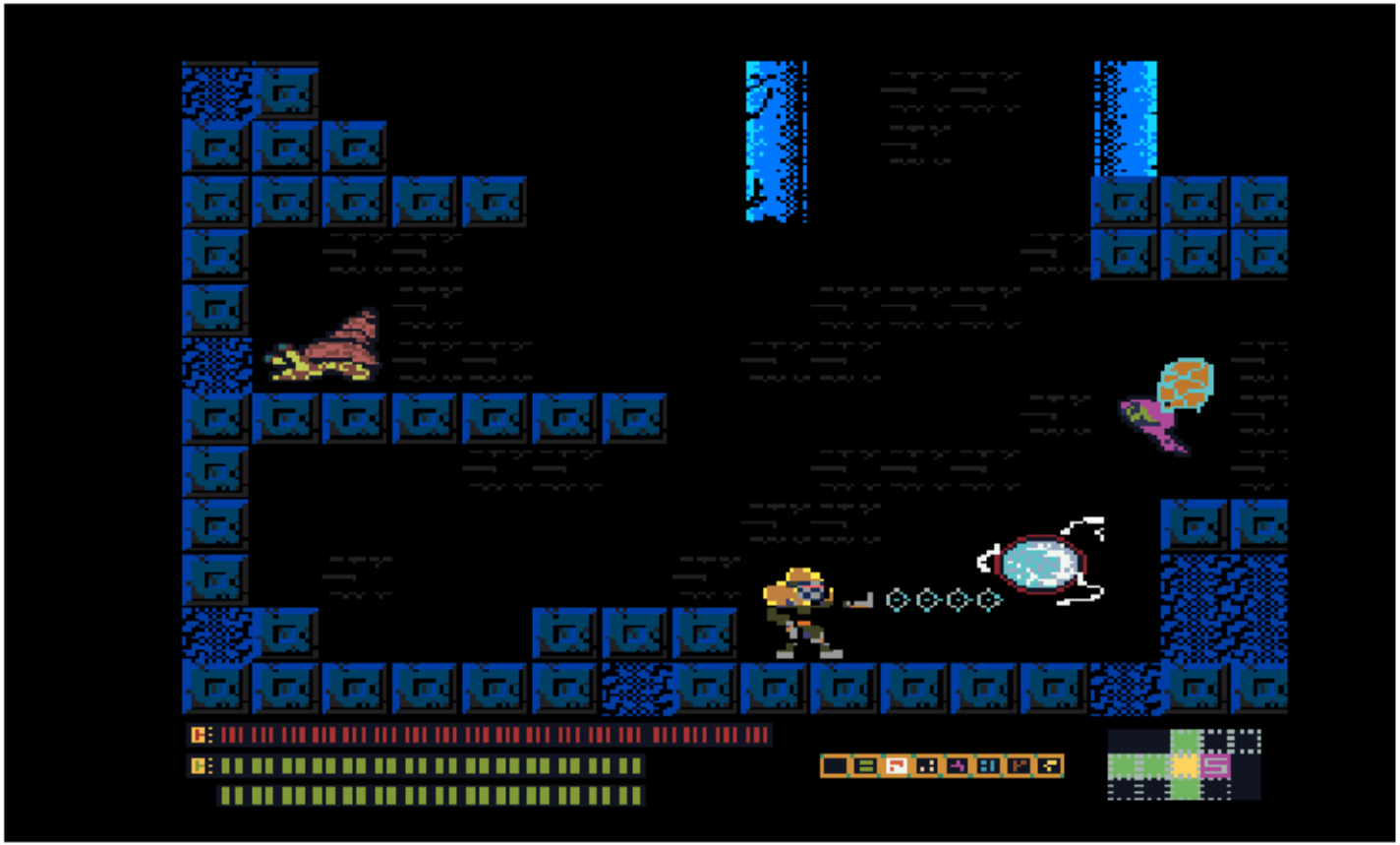
Extreme Pygame Programming

Platformer



python™

pygame

Player Movement

- Installing Pygame and Python on our machine
- Pygame Feature Overview
- Python Programming Concepts
- Platformer Project Resources
- Live Coding Activities

Platformer Concept

Performance Goals

Enemies and rooms are inactive until the player gets there.

“ No reason to shoot at something before I see it on screen.”

Rooms are usually reset after a player leaves that area.

“I return to an area and I still have something to do.”

Smooth and responsive game controls.

“I can jump successfully.”

Quick Menu access.

“ I want to turn the game off now .”

Python Tutorials Online

https://www.w3schools.com/python/

https://www.learnpython.org/

https://www.tutorialspoint.com/python/index.htm

https://realpython.com/

https://www.pythontutorial.net/

Pygame Resources

https://www.pygame.org/wiki/tutorials

https://www.geeksforgeeks.org/pygame-tutorial/

https://realpython.com/pygame-a-primer/

https://pythonprogramming.net/pygame-python-3-part-1-intro/

https://www.techwithtim.net/tutorials/game-development-with-python/pygame-tutorial

Indie Game Resources

https://itch.io/

https://gamejolt.com/

https://opengameart.org/

https://www.1001fonts.com/video-game-fonts.html

https://www.cufonfonts.com/seulgii/collection/game-fonts

Popular Media

https://dafluffypotato.itch.io/gleamshroom

Installing Python and Pygame

Website for Downloading Python

- https://www.python.org/downloads/

Pygame Documentation

- https://www.pygame.org/docs/

Pygame

- https://www.pygame.org/wiki/GettingStarted
- **Windows/ Raspberry PI**

  py -m pip install -U pygame --user
- **Linux /MAC**

  python3 -m pip install -U pygame --user

Pygame

- **Linux**

  sudo apt install python3-pip

  sudo apt-get install python3-pygame

Pygame

- **MAC**

  sudo apt install python3-pip

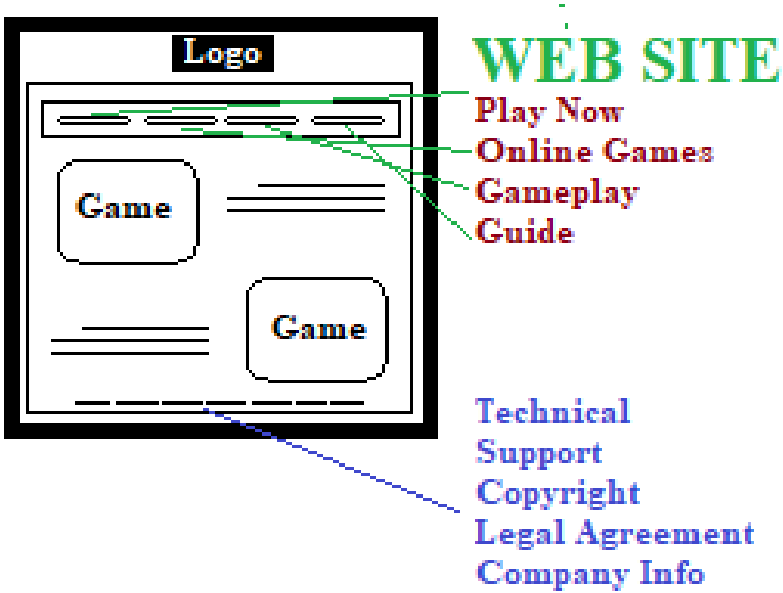
  sudo apt-get install python3-pygame

Where can I display my Projects?

GameJams
Social Media
Video Streaming Services
Portfolio Websites
University Seminars
Online Banners
Commercial Markets
Advertisement Venues
Indie Game Competitions

Marketing Your Game?

- Add a short gameplay trailer.
- Screenshots of action in the game title.
- Not too much text on this page.
- Supported platforms.
- System requirements meaning the hardware and software .
- Clear overview of what they are going to receive when they make a purchase.



Python Programming

- **Hazzard Dude –** **Score based shooting platformer**
  - Hero States
  - Enemy Actions
  - Background Layers
  - Dynamic Projectiles
  - World Tileset
  - HUD Elements

Class Objective

Simple Multi-stage 2D Platformer

Complete Overview of Pygame/Python Programming

Modern Platformer Design

- Player Movement
- Player Controls
- Dynamic Window Resolutions
- Screen Scrolling
- Enemy Behavior
- Shooting Projectiles
- HUD and Mini Maps
- Health and Ammo
- Changing Skills
- Event Loops and State Machines
- Rendering Tilemaps
- Collectable Items
- Collision Detection
- Saving and Loading Game States

How do I Improve my Artwork

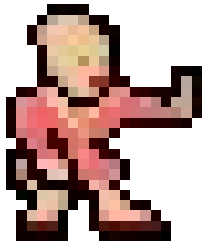
- **What should I do to make better art?**

  Use a template.

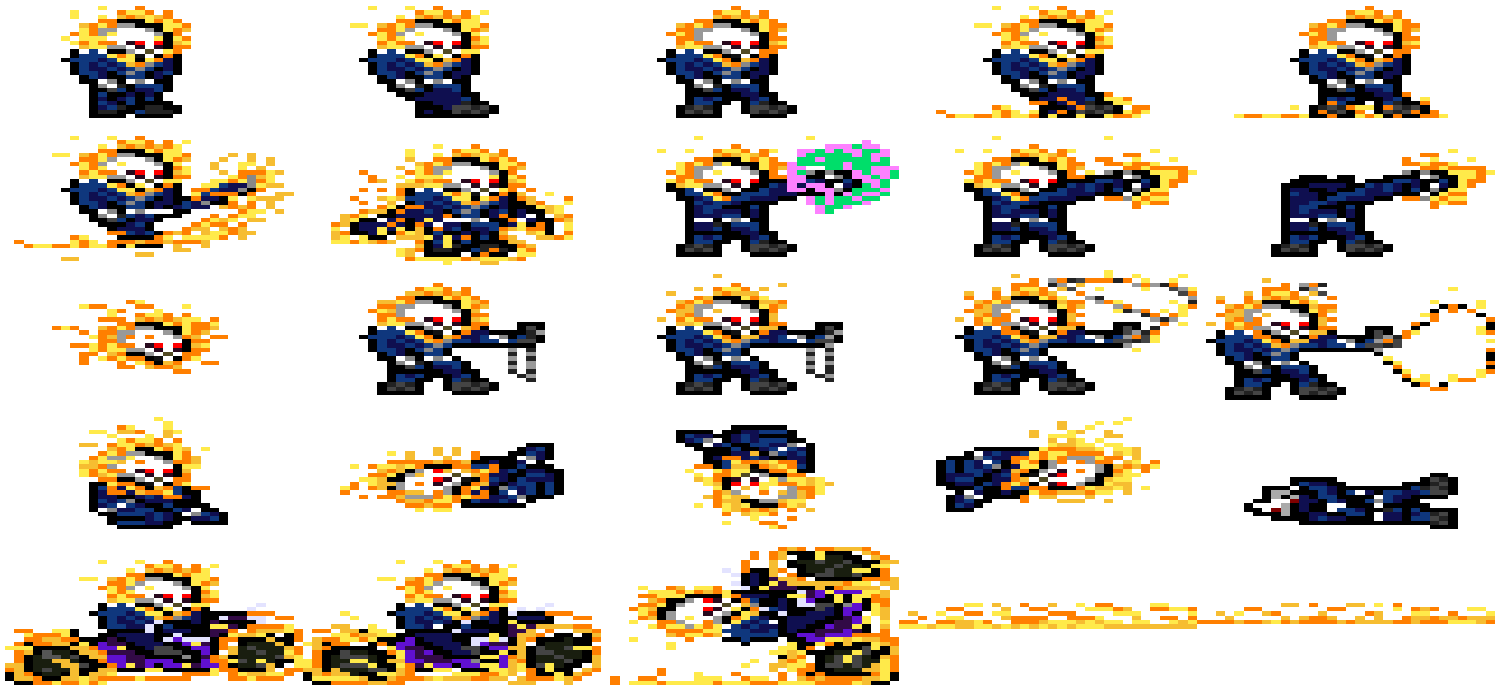
  Practice

  Watch Tutorials

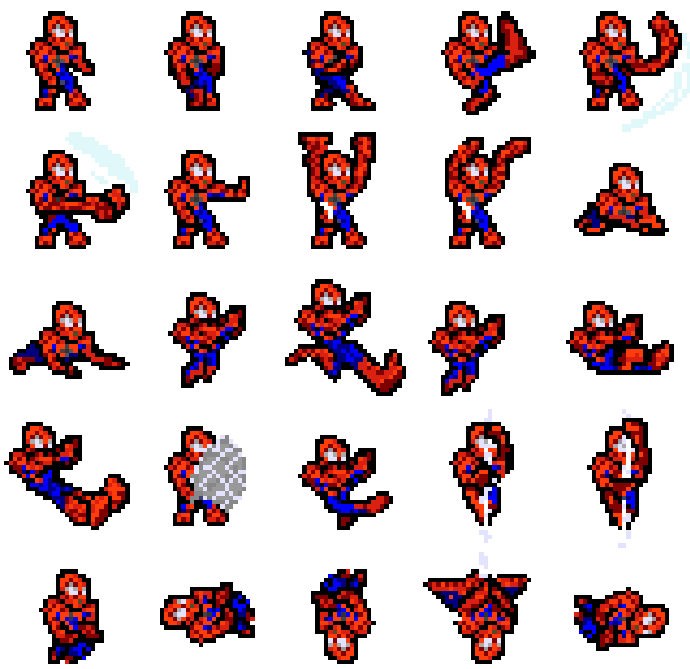
  Pay someone to learn

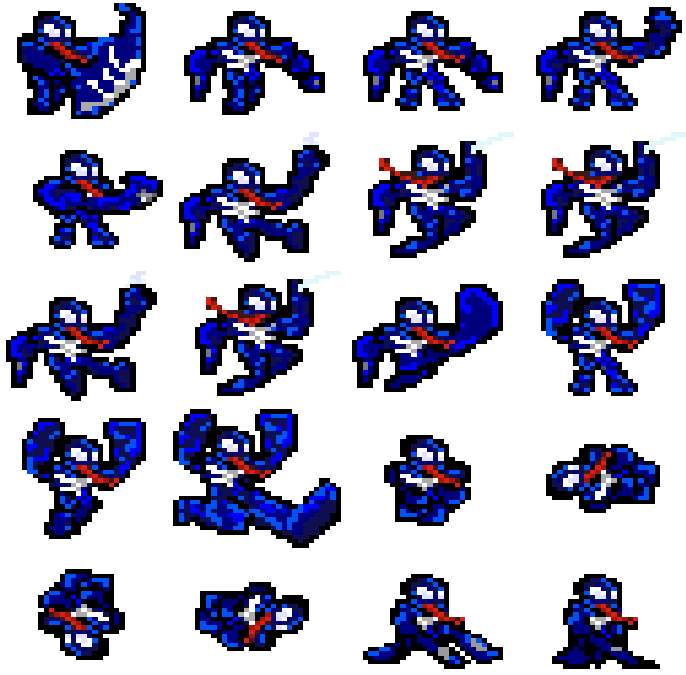


Ghostride Spritesheet



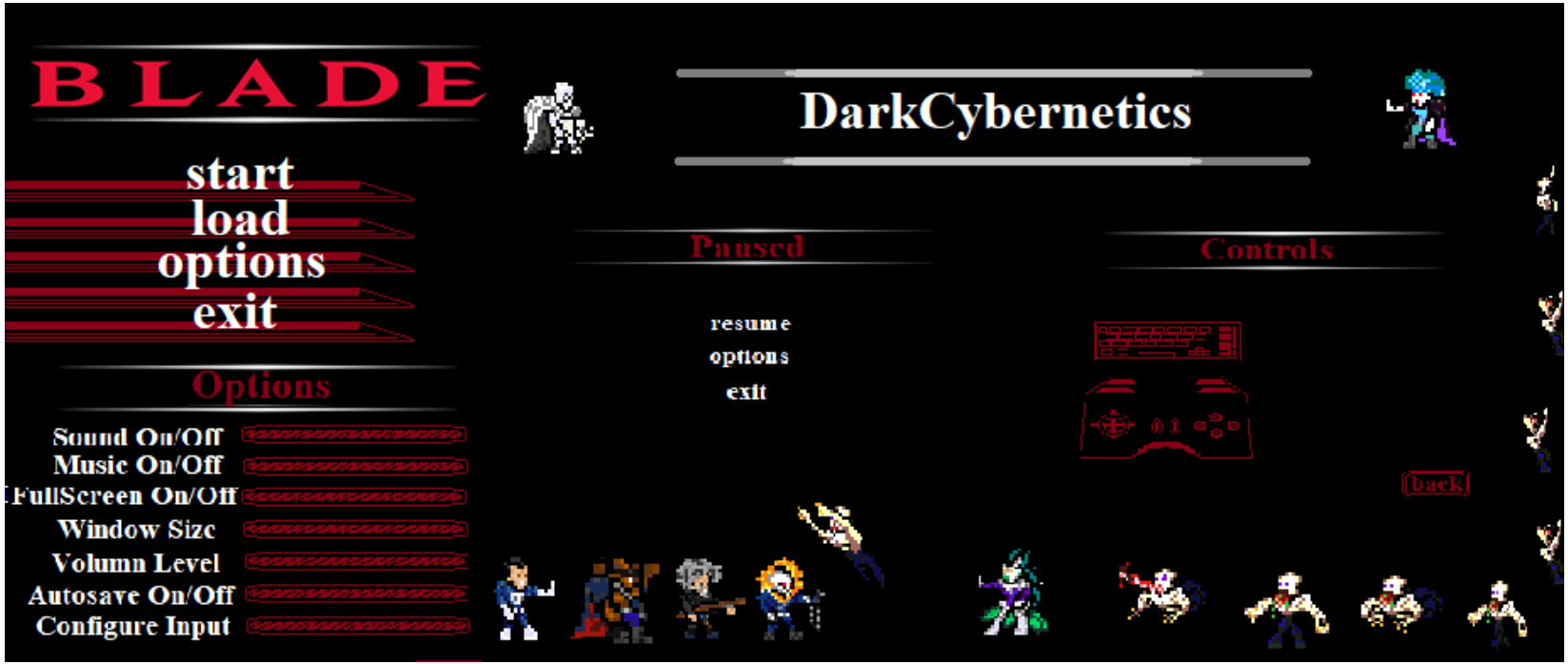
Are you thinking about making an indie Marvel fan game?



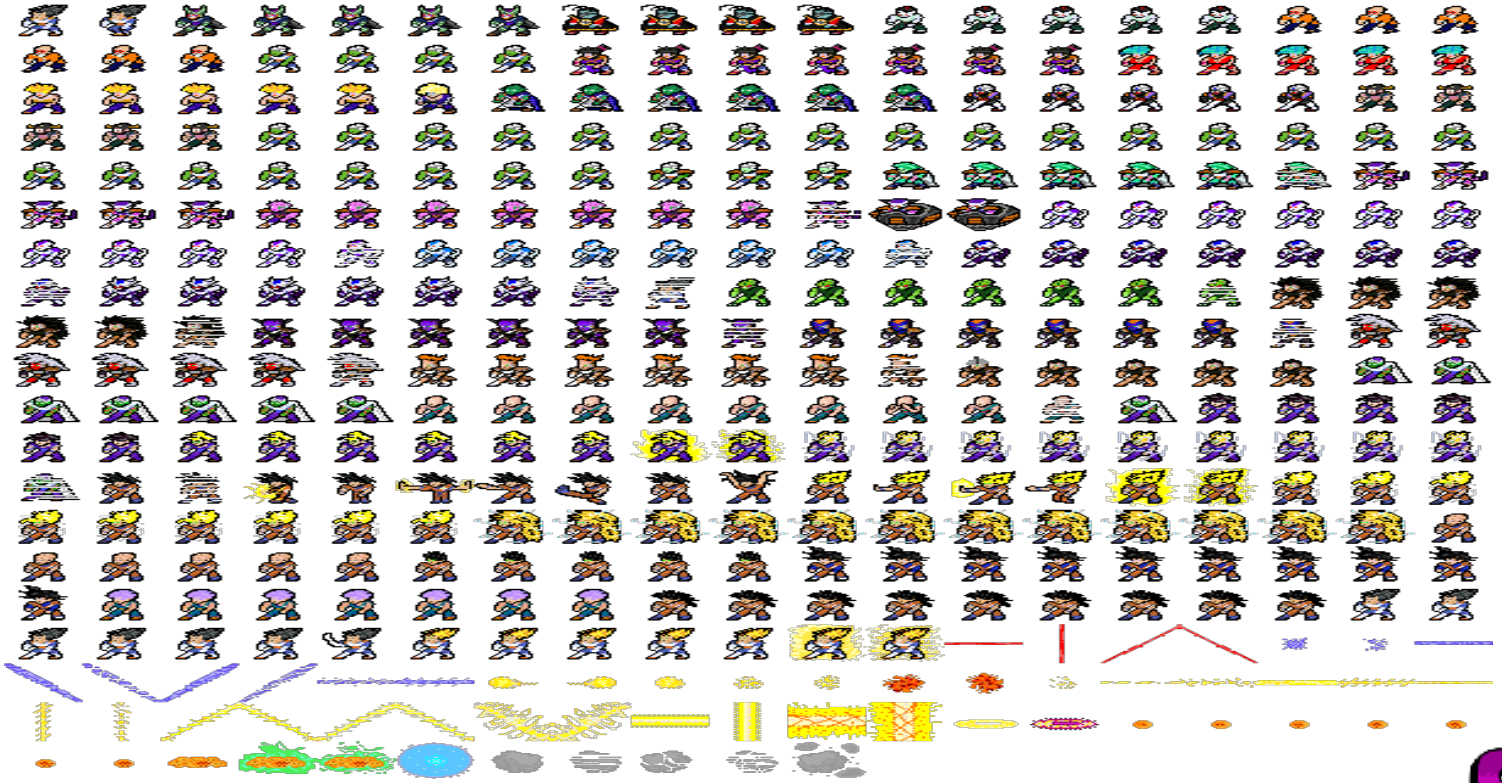


Do you like the X-men?

What about Blade?

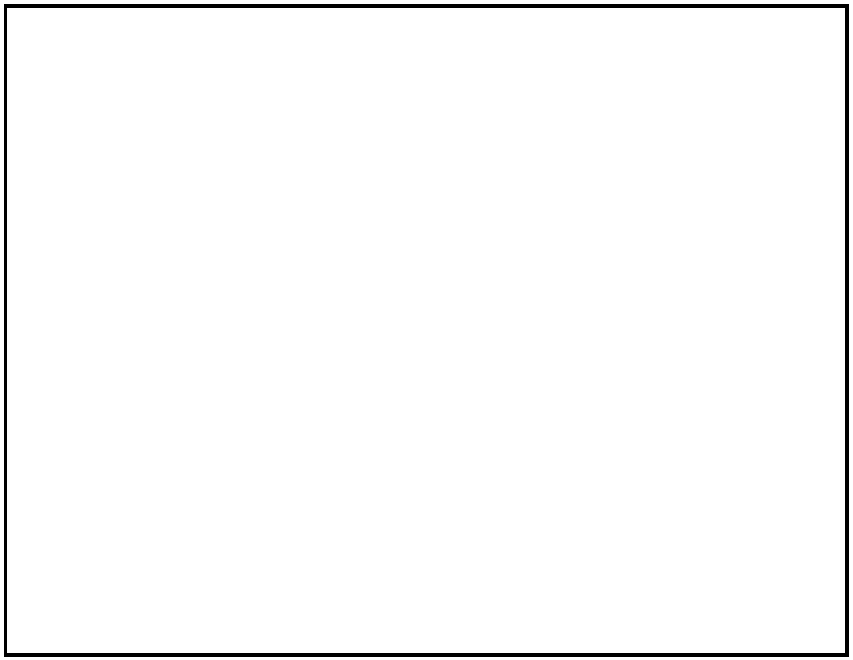


Do you like Dragonball Z?



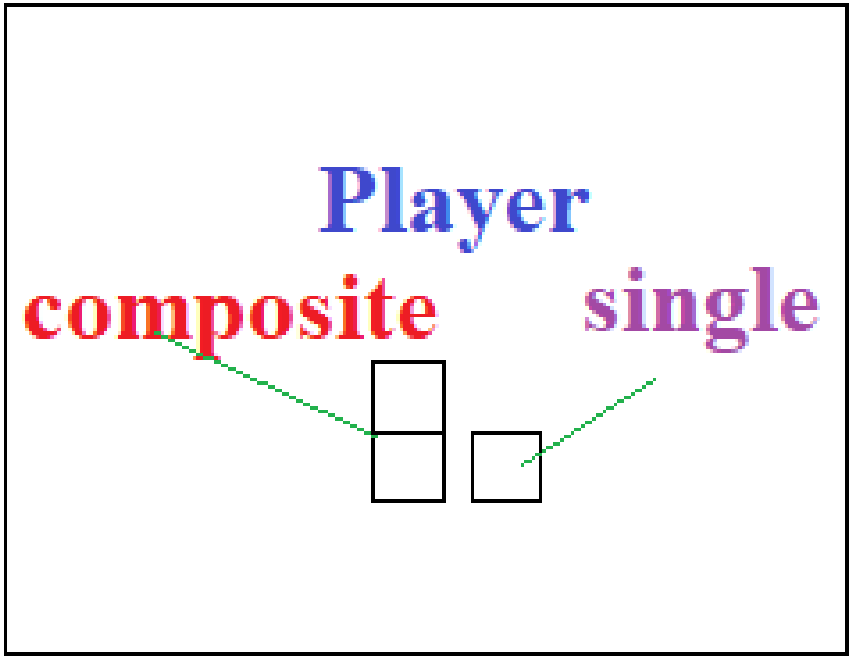
Platformer Concept

Canvas – area of pixels.



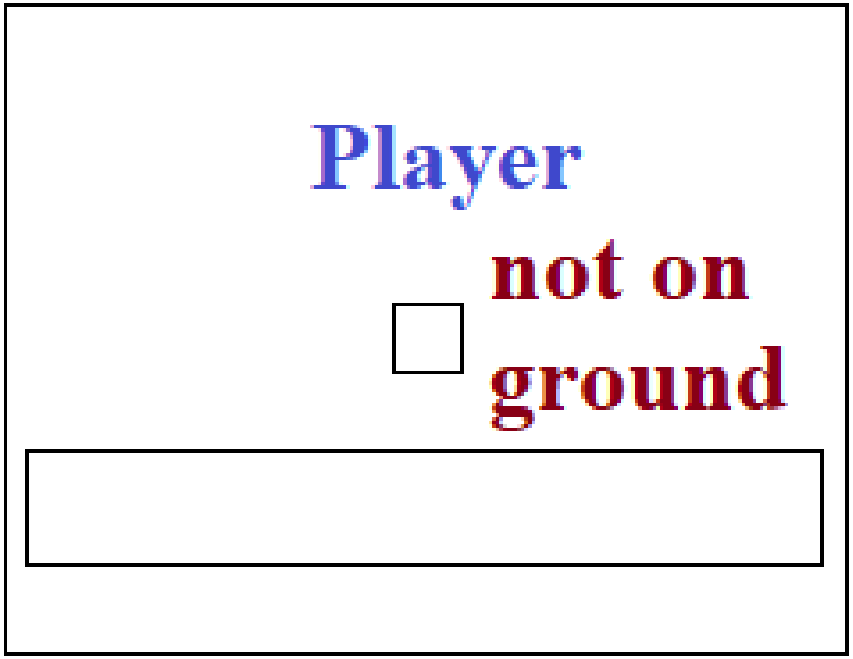
Platformer Concept

Game sprites are either made from a combination of smaller images(composite) or a single image.



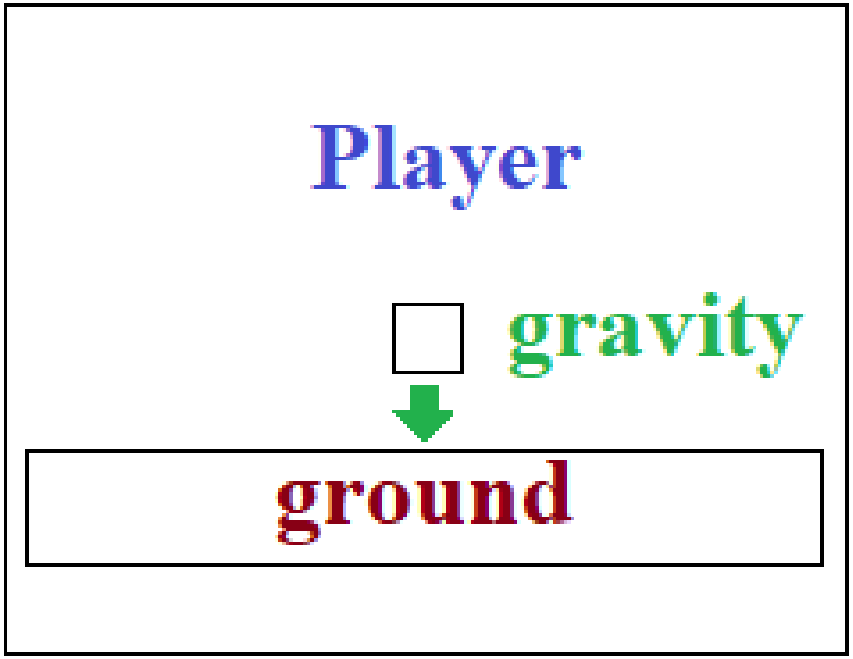
Platformer Concept

Platformers have some real world concepts such as gravity and motion.



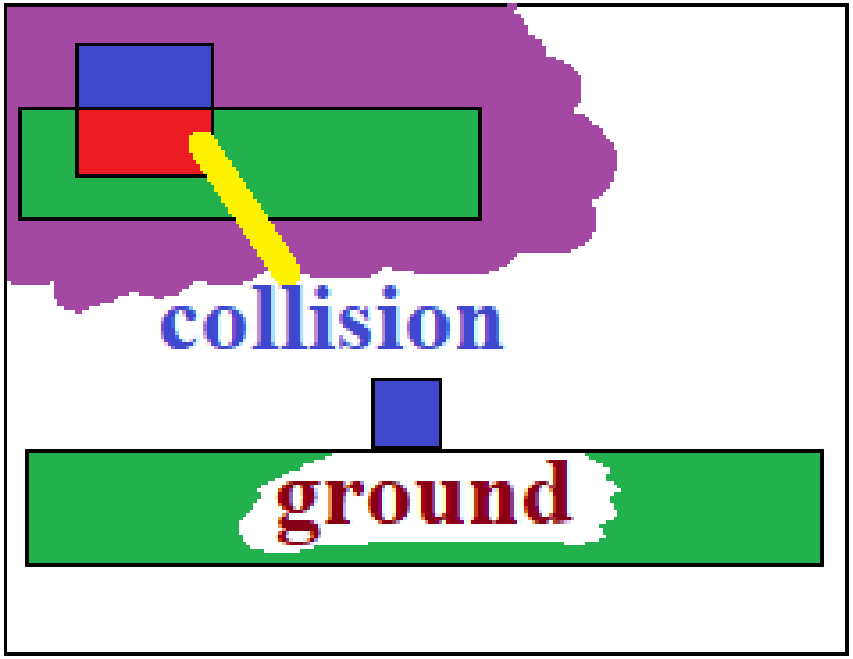
Platformer Concept

Gravity is a strong negative constant in a game world.



Platformer Concept

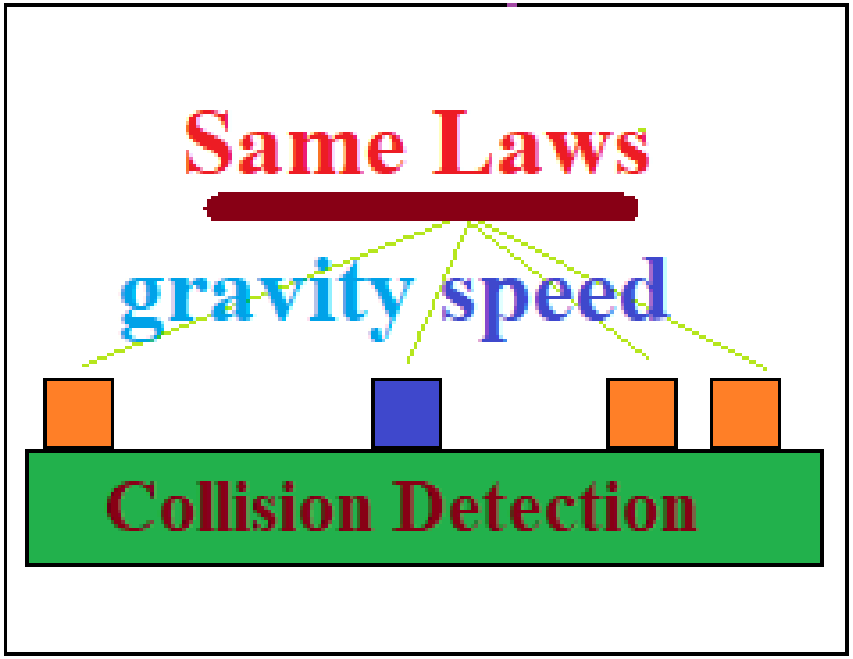
This strong constant called gravity is reduced to zero once a player collides with a solid object.



Platformer Concept

In Platformers most object have these following attributes.

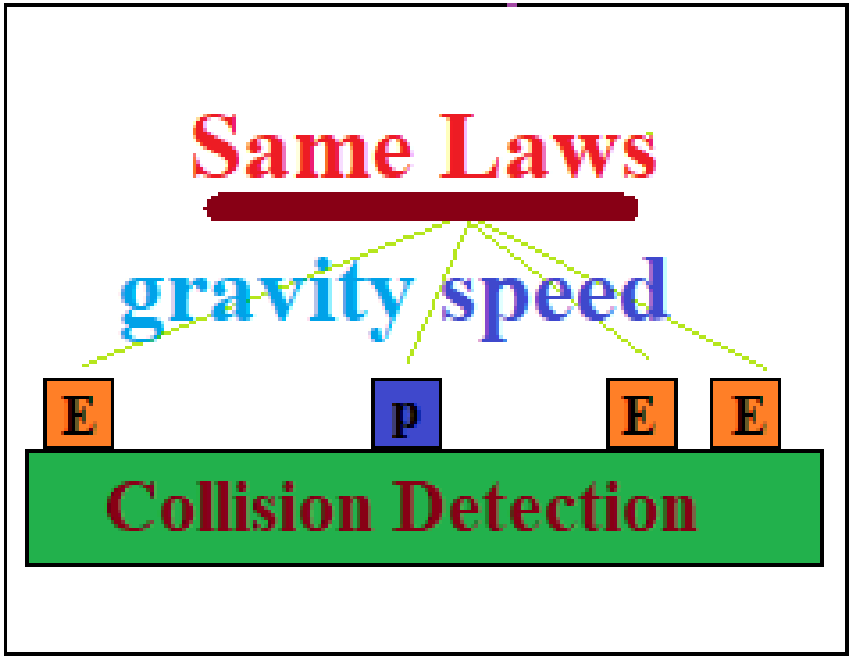
- **Rate of Speed**
- **X and Y positions**
- **Gravity**
- **Life Amount**
- **Behavior**
- **Solid or Not**



Platformer Concept

E - Enemy

P - Player



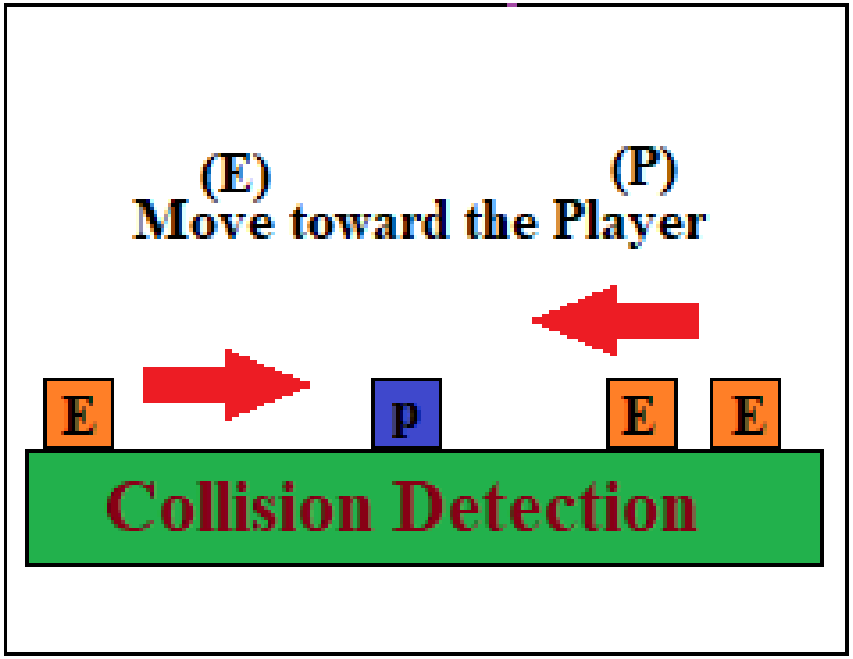
Platformer Concept

Natural Motions

Simple AI behavior for agents in a game world.

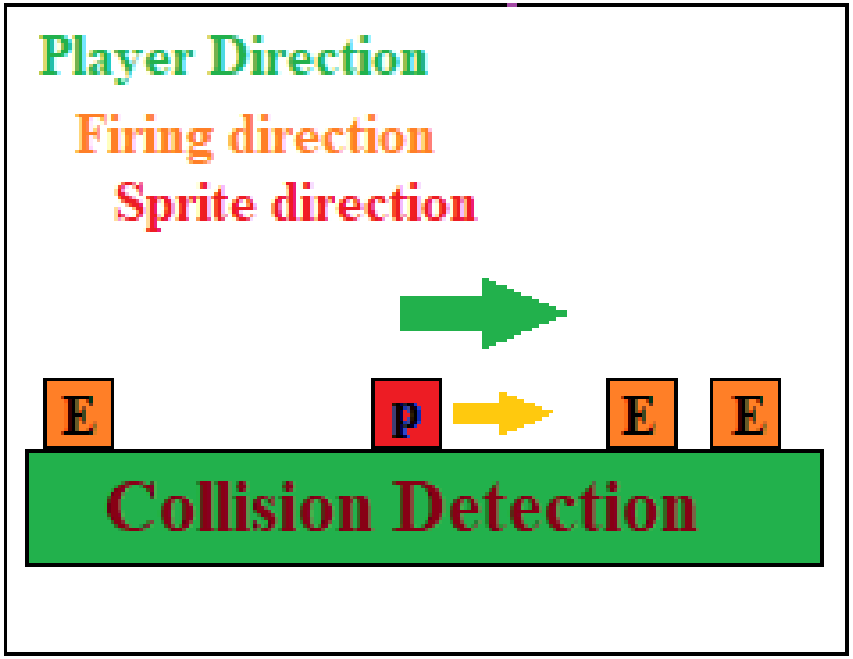
Moving left to right.

Facing and following a player.



Platformer Concept

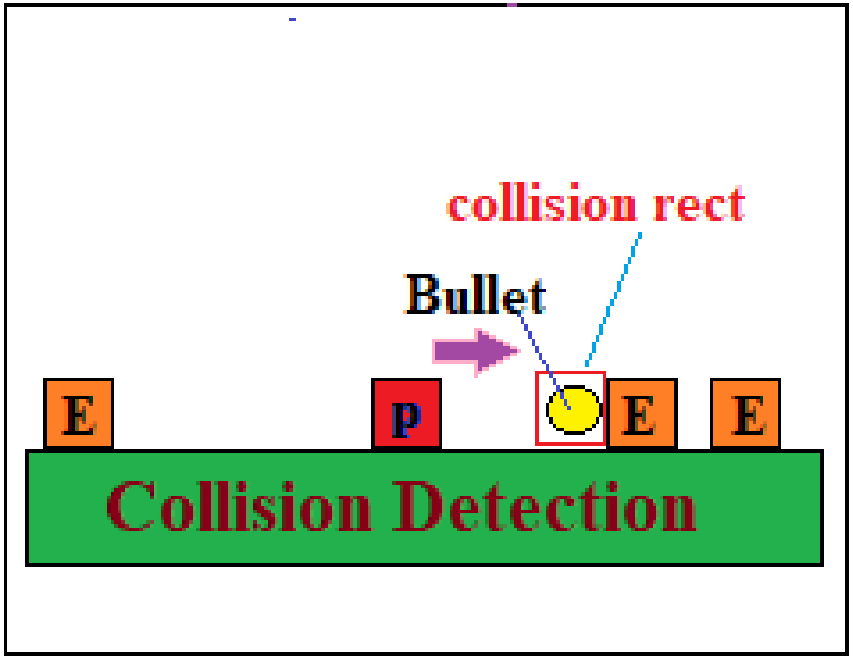
The player's direction determines the facing direction for attacks and jumping angle.



Platformer Concept

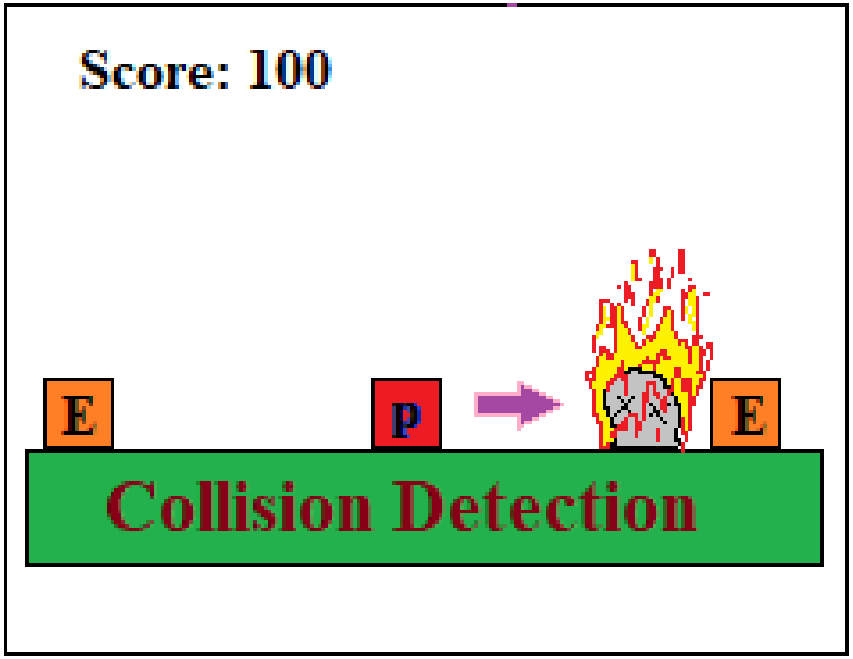
Bullets are solid objects as well.

A collision rect is a square region around game objects. This rectangle is usually an approximated size used for AABB collision detection strategies.



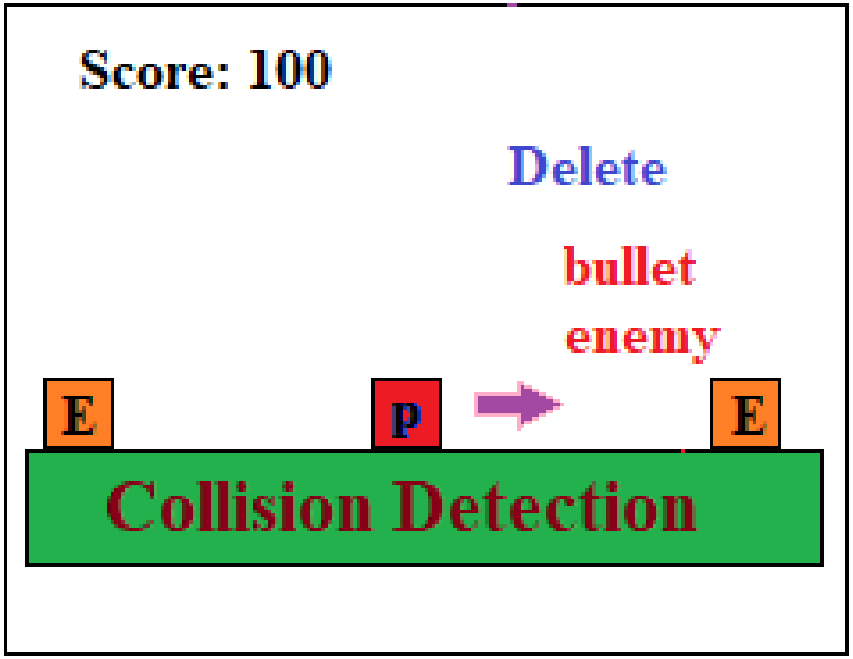
Platformer Concept

In some games a death animation is used to indicate to the player the object has been defeated.



Platformer Concept

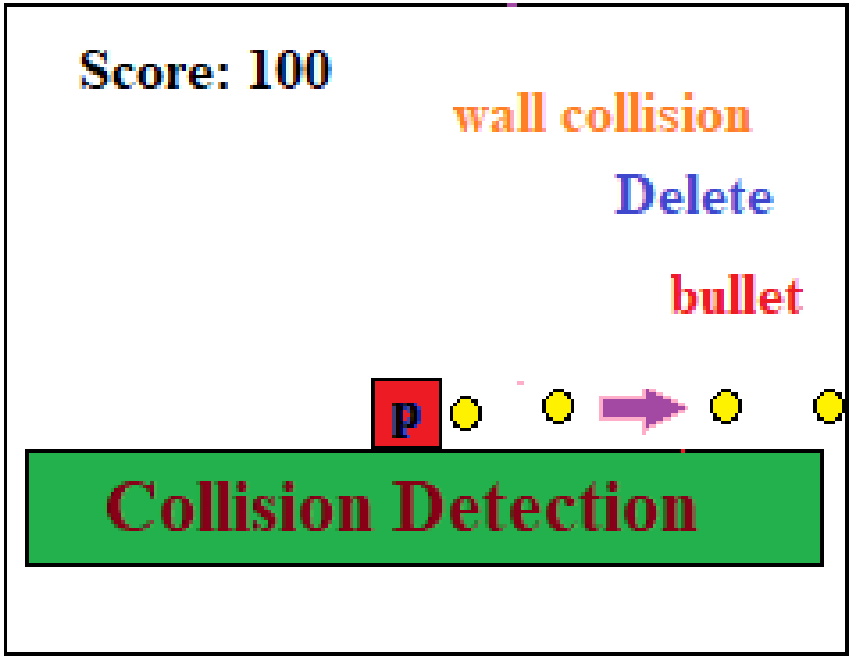
The destruction of an enemy usually results in the destruction of the entity being shot and the object responsible for the shooing the bullet.



Platformer Concept

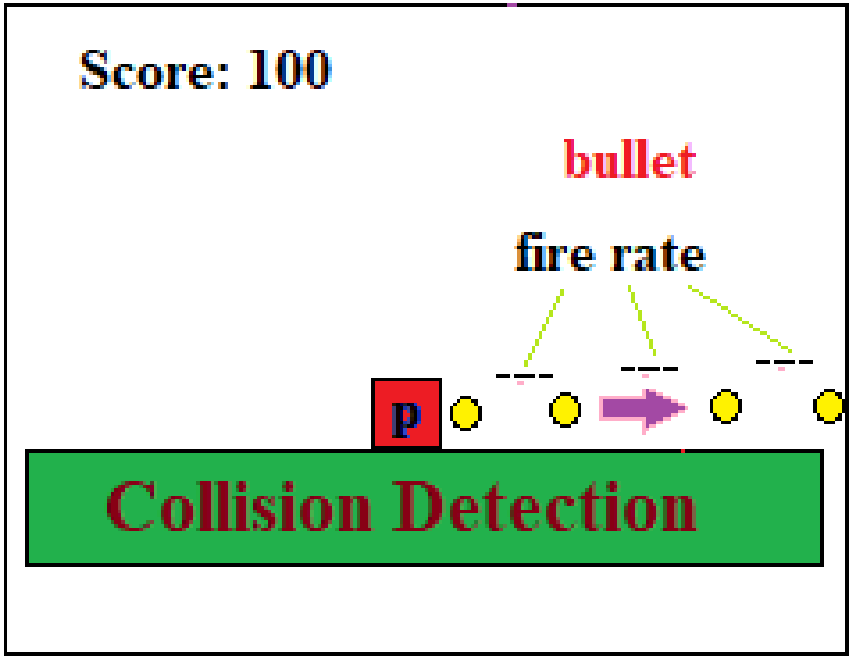
The bullet only travels as far as the width or length of the screen.

On collision with the edge of the screen it is also destroyed to save memory.



Platformer Concept

A firing rate can be used to regulate the amount of time between each weapon's attack. Also this can be used to simulate a variety of different attacks for the player or enemy.



Platformer Concept

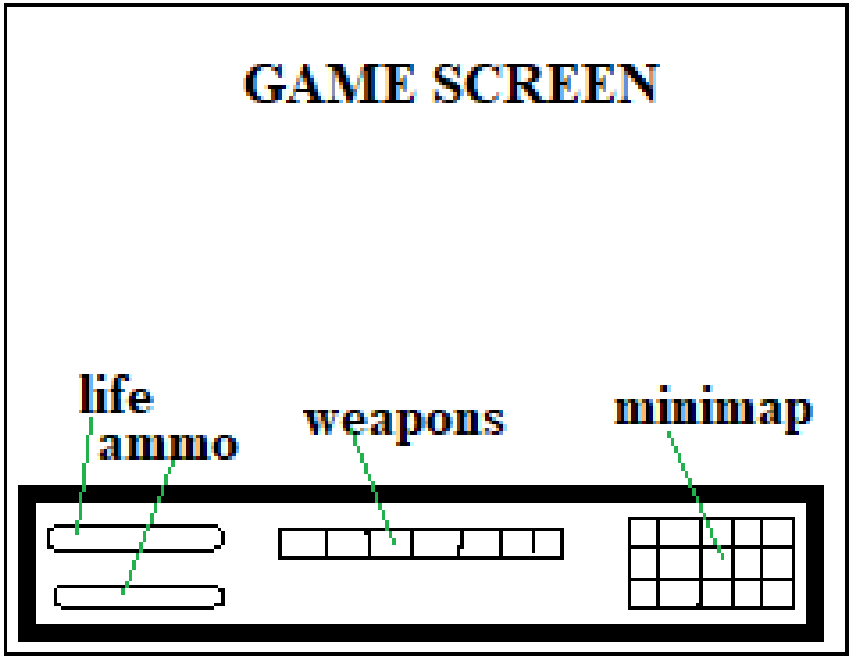
Our game screen.

Life

Ammo

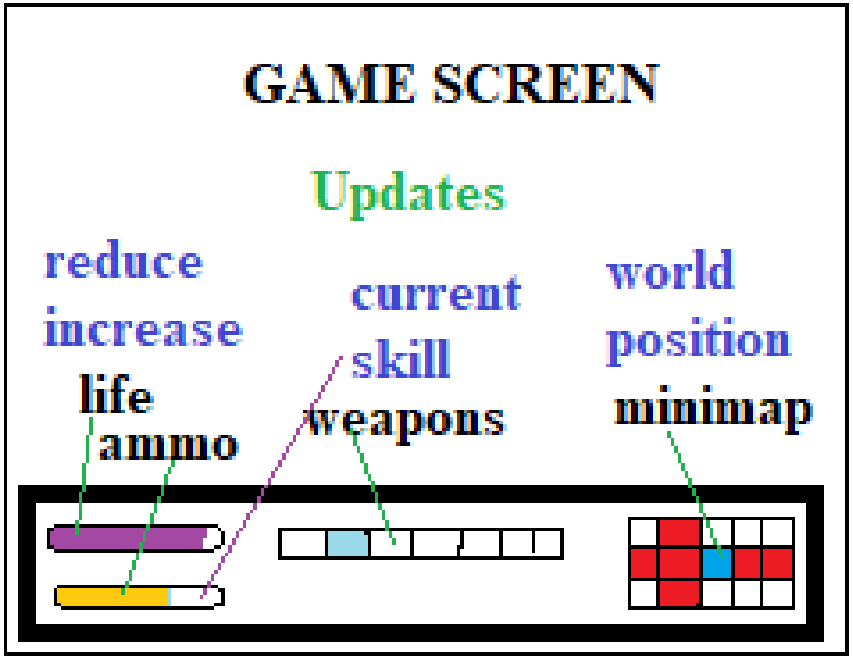
Weapon

Minimap



Platformer Concept

The last thing that is updated is the game display region because it is the top most layer of the game world. This region can also include text of custom font or life bar items. "Last thing drawn to screen."



Creating Free Art Assets

Paint **< Windows Operating System >**

https://www.piskelapp.com/

https://www.gimp.org/

https://orama-interactive.itch.io/pixelorama

https://www.mapeditor.org/

Let's Pretend

We want to create a game that we can play on the Gameboy Color.

Let's Pretend

Gameboy Color

Screen Resolution -160 x 144 Pixels
32,000 different colors displayable, of which 56 simultaneously

Screen size = 4,4 x 4 cm

Game Story

Some dude ran into a group hostile alien lifeforms in an abandoned complex. Explore the abandonment space station and unlock secrets and gain new abilities. Use whatever means necessary to survive.

How do I handle Keyboard Input?

```
import pygame,sys
from pygame.locals import *
pygame.init()
canvas=pygame.display.set_mode((400,300))
pygame.display.set_caption("Hello")
canvas.fill((0,0,0))
while True:
   for event in pygame.event.get():
      if(event.type == QUIT):
         pygame.quit()
            sys.exit(1)
```

Player Movement

- **Left(-) and Right(+) make the player move left or right.**
- Player_Speed = 5

  Check is the player on the ground?

  Change player Direction

  Animate player (walking)

    else

  Player_Speed = 0

  Animation player(idle)

Pygame

```
if event.type == KEYDOWN:
    if event.key == K_RIGHT:
      x= x+5
    if event.key == K_LEFT:
      x=x-5
    if event.key == K_UP:
      y=y-5
    if event.key == K_DOWN:
      y=y+5
    pygame.display.update()
```

- event.type
- event.key
- pygame.display.update()
- K_DOWN
- x=x+5

Player Movement

Hazzard Dude

UP Down Left Right
Player Movement

Select
Toggle Skill / Hold for Menu

A
Jump

B
Shoot / Accept

Start
Pause / Options Menu

PC Controls

- Movement (Directions)
- A – A (Jump)
- B – S (Shoot)
- Start – Enter(Pause)
- Select -Space Bar (Menu)

What about Controller Input?

Why user testing is important in games.

- Device Dependent (phone?)
- Active Driver Support (Window 7?)
- Operating System Input

  (MAC/Ubuntu?)
- Library Support(New Version of Pygame?)

Hazzard Dude Assets

- *Game Projects Folder Names*

  res – short for resources(.png)

  sfx – short for sound effects(.ogg)

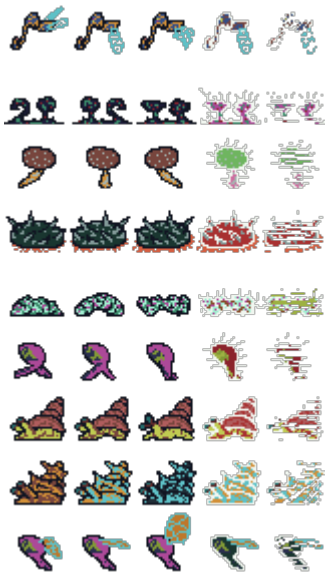
  src – short for source code(.py)

Color in Game Design

- **Separate Pallets into two groups**

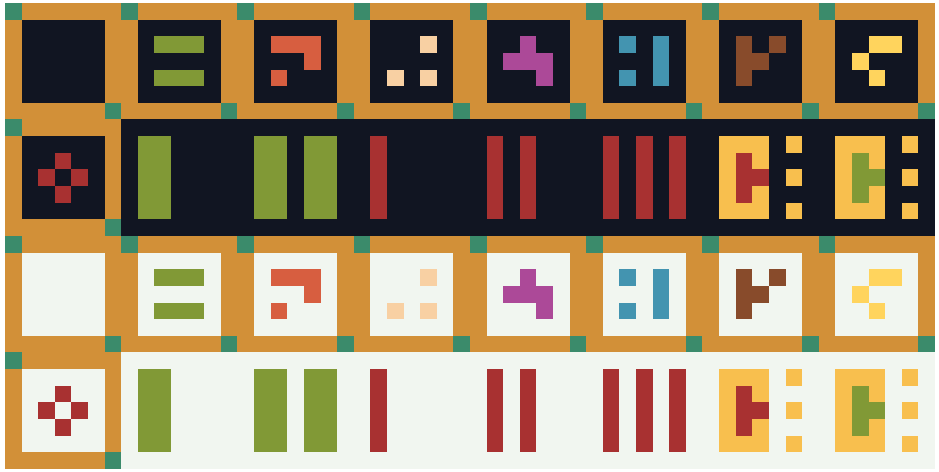
  Player Enemies Items Coins

  Backgrounds Level HUD



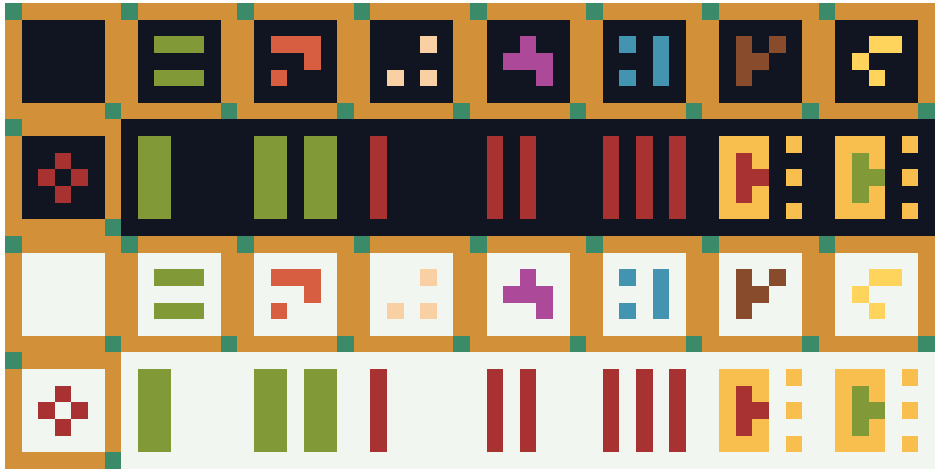
Game Resources

- Sound Effects
- Game Graphics
- Videos
- Menu and Buttons
- Font



Weapon Select

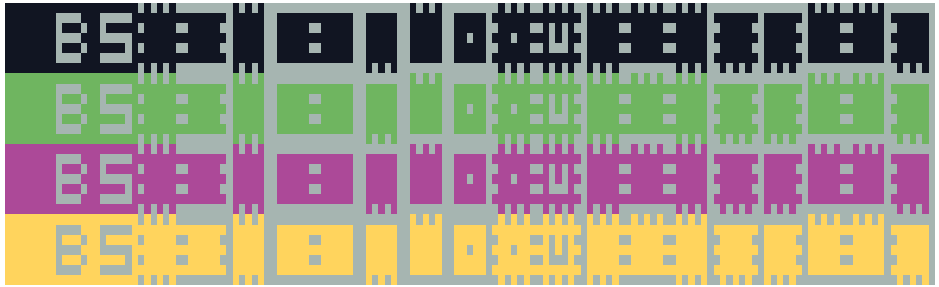
- Has a state of unlocked ,inactive , and occupied.



- Weapon Skill
- Health
- Energy

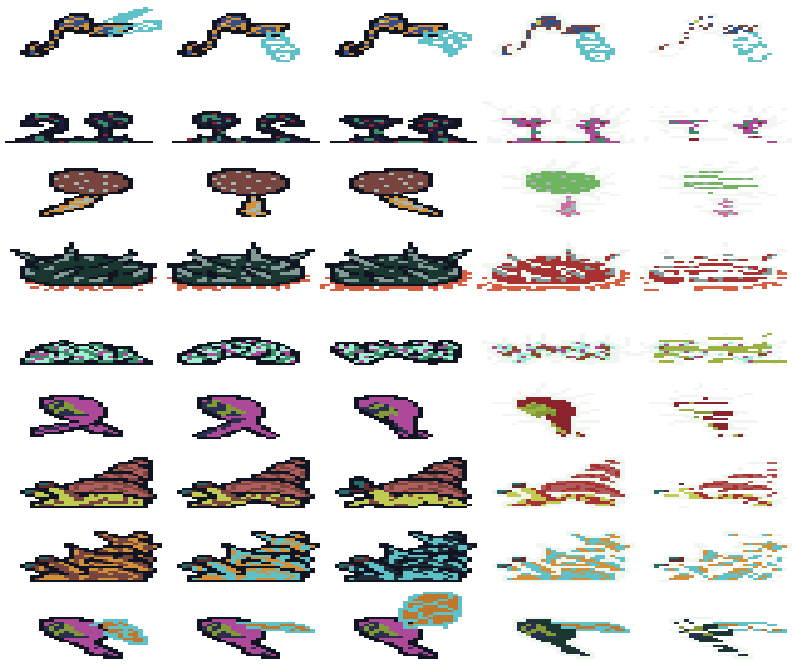
Minimap System

- Current Room
- Explored
- Unexplored
- Save Rooms
- Boss Areas



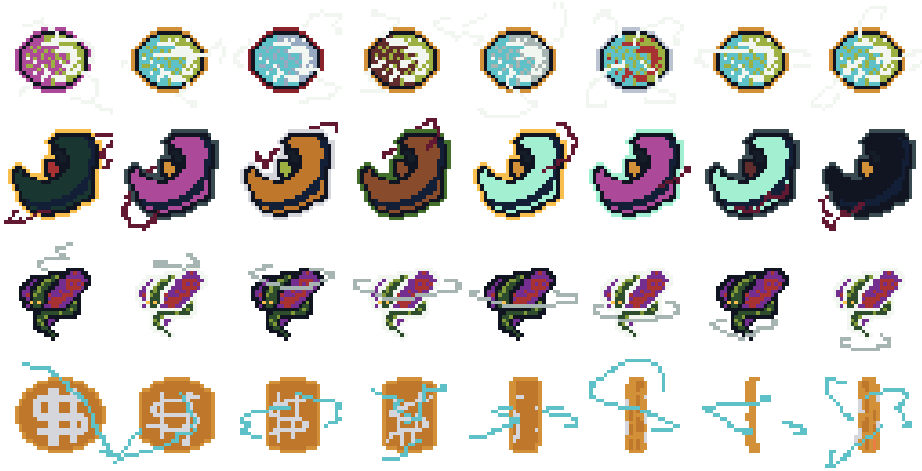
Tileset

- Enemies have an idle moving hit and dead state.



PowerUps and Collectables

- These items just animated and disappear once the player collides with the item.
- Coins , Ammo , etc..

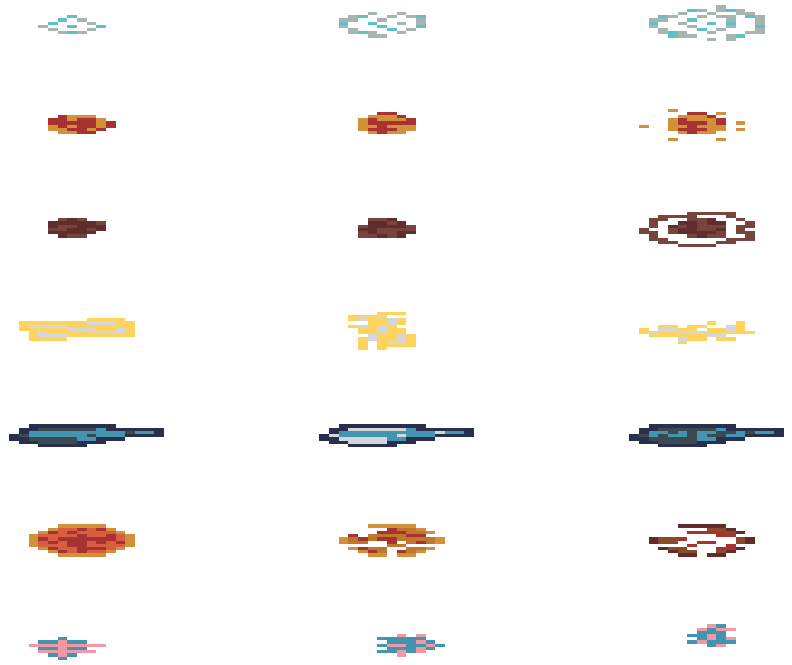


Tileset

- The tileset is the level and background held in row and column format this one also includes the base color pallet.

Bullets or Projectiles

- Hazardous dart like weaponry created by the player or the enemy.
- Items have a constant positive direction and are destroyed on impact.



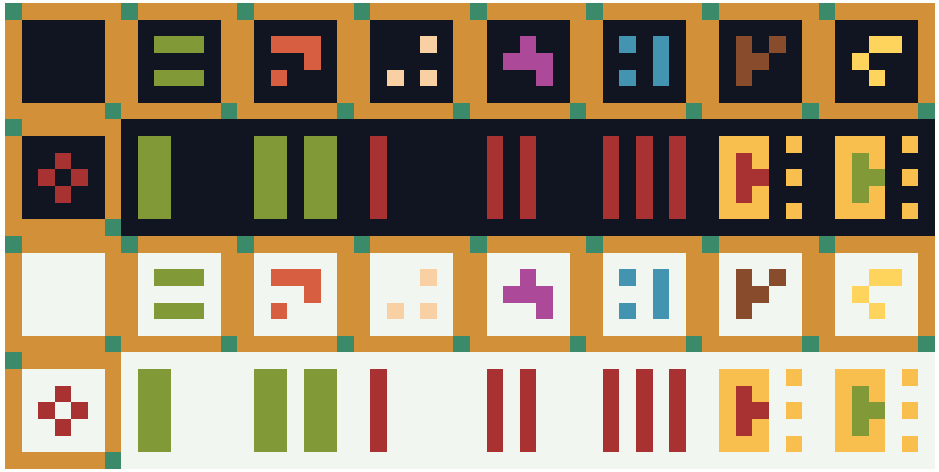
The Player Dude

- A Player can have various states. These playable characters often have changeable attire or weapons in most games.

Color Pallet

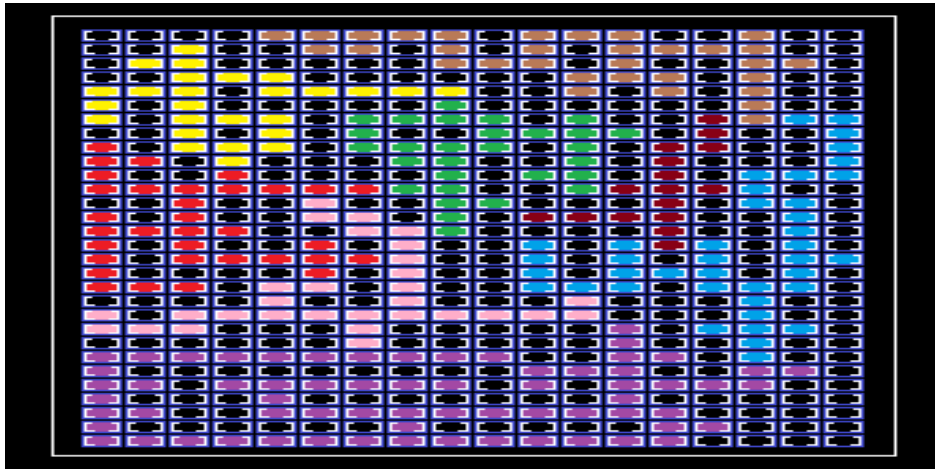
- Using independent color sets is a graphical way to distinguish the foreground from the background.
- Enemies /Solid tile
- World / effects

Tileset



Level Design

- Each game area is denoted by an individual color.
- Some games use sectors.
- Each squre is a room which may be individually loaded into memory to save CPU cycles.



Making the Minimap Work

- The player is inside a room cell.
- That date corresponds to a region of squres inside the minimap array.

- 5 by 4 cells(Sprites)
- Some games use a pixel array like a form of primitive radar to save memory.

  Ex: RTS Games

Basic Game Event Loop

- While (game not over){

  Check for user input : mouse press or buttons on keyboard

  Update game actions : player shoots and animations

  Check for game conditions : Player on the ground? Alive? Gun can shoot? Can Jump? Game not paused? Can collection item?

  Draw to the screen : background , sprites , objects ,HUD

  } #end of while loop

Dynamic Window Resolutions

- The window fits the desired device appropriately.
- Full-screen mode is a selectable option.
- I can change the dimension of the x and y for different size montior displays.
- No forced vision modes not supported or no longer supported by GPUs or monitors .

HUD and Mini Maps

- The Minimap can be an array which tells the player where to go next. This menu can include the location of boss fights and collectible objects. Trigger events also can be placed on this grid such as a save location of a level transition region.

Enemy Behavior

Desired enemy behavior depends on game performance specifications. Usually the enemy has the x and y position of the player at all times.The enemy only moves when it is in the radius of the player. A path from the player to the enemy is developed by a graph search algorithm.

Shooting Projectiles

Constant applied to a created object.

Health and Ammo

Increases a variable for a fixed value.

Changing Skills

Changing weapon types.

Event Loops and State Machines

Start screen to game screen and onto the play state.

Rendering Tilemaps

Crop out squares and place them in a grid in desired intervals.

Collectable Items

Spawn items which aid the player.

Collision Detection

Axis Aligned Bounding Box Collision Detection. Determine the distance between two object and see if they intersect.

Saving and Loading Game States

Store a code in memory which is used to later determine the position and HUD condition of our player.

Player Movement

Jumping , Shooting , Being Hurt , Death

Regulate the Frames Per Second

Smooth Animation

Platformer Game Project

- Operating system determines what should be in the game folders.

  Python requirements

  SDL2 requirements

  Graphical and Audio resources

  Font

Homework

Design **Characters** , **Enemies** , and the **Game Stages** on paper and convert them into **pixel art**.

Homework

Make Game Folders

Homework

Create Game Sprites

Homework

Install Python and Pygame