2021-07-22 Start

[2021 Update!] Make Games with MonoGame - Installation and Development Fundamentals

<https://www.youtube.com/watch?v=sPH-sNTSrhw&t=0s>

1. Introduction
2. Installing visual Studio and Monogame
3. Creating new projects and code organization
   1. Windows CrossPlatform
   2. Constructor: public game1()

* We can change windows size or
  1. Initialize method
* When game starts it is executed, setting
  1. LoadConent method
  2. Update method
* Gameloop => 60times in a second
* Game frame
  1. Draw method
* E.g. show target
* Runs in every frame

1. Importing assets with the Pipeline tool
   1. Content Pipeline: The MonoGame Pipeline Tool is used to create and manage MonoGame content projects. The files in content projects are processed by the MonoGame Pipeline tool and outputted as . xnb files for use in CocosSharp and MonoGame applications.

Graphical user interface, application

Description automatically generated

* Add Existing Item - Build
* .xnb files

1. Drawing graphics #1
   1. Sprites: you take your individual animation frames (of characters, objects, etc.), and drag and drop them into TexturePacker. Then TexturePacker combines them into **sprite sheets**, which you can then load into your game.
   2. Load Content
   3. // Assign Imported assets(.xnb files = stream target) to targetSprite

targetSprite = Content.Load<Texture2D>("target");

* 1. Term – SpriteBatch: A SpriteBatch is used to draw a whole bunch of sprites all at one time. ... We will then draw our sprites, and when we're done, we will indicate that we are done drawing our batch of sprites. Go down to the Draw() method and add the following three lines of code:

1. Drawing graphics #2
   1. \_spriteBatch.Draw(targetSprite, new Vector2(0,0), Color.White);

* Three parameters
* Vector: position (Origin: (0, 0)), unit pixel, increase y 🡪 downer
* Color: tint
  1. Drawing Order

1. Printing Text(SpriteFonts): Blue print of fonts in Monogame
   1. SolutionExploer – Content – GalleryFont.sprite

* In draw method:
* \_spriteBatch.DrawString(gameFont, "Test Message", new Vector2(100, 100), Color.White);

2021-07-23

1. Variables
   1. Declare variable

* Vector2 targetPosition = new Vector2(300, 300);
  1. Using variable
* \_spriteBatch.Draw(targetSprite, targetPosition, Color.White);
  1. Declare constant variable
* Doesn’t change and cannot be changed
* Using const keyword

1. Mouse Input
   1. State: What mouse do particular moment(click, moving, etc…)

* Because update method is executed in every frame, score increases more than 1.
  1. Use mRelased that indicates mouse button released state

1. Shooting the Target
   1. How can we tell that the mouse clicks the target?

* Calculate the distance between the mouse click point and center of the target
* If the distance is less than target’s radius, it’s hit
* The origin point of sprite is upper left corner
* Offset: \_spriteBatch.Draw(targetSprite, new Vector2(targetPosition.X - targetRadius, targetPosition.Y - targetRadius), Color.White);

1. Randomness
   1. Change Target Location Randomly after hit

* // Change target Position randomly
* // Range is exclusive
* Random random = new Random();
* targetPosition.X = random.Next(0, \_graphics.PreferredBackBufferWidth);
* targetPosition.Y = random.Next(0, \_graphics.PreferredBackBufferHeight);