MOOC
Oral presentation

“ Beginning Game Programming with C# ”

Anthony Barbier

Plan

I - Personal project : introduction

II - Personal project : development

III - Let’s play !

I - Personal project : introduction

I - Personal project : introduction

→ Pac-Man-like game

→ Simplified :

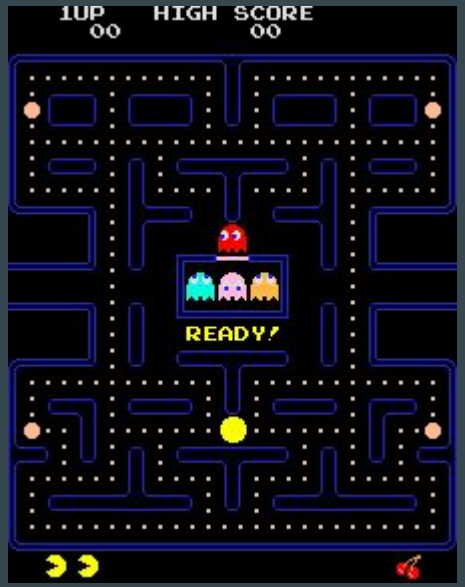
- only one game played
- no candies
- tunnels = dead end
- ghosts start at corners
- ghosts not intelligent
- no animation when Pacman dies

Original Pacman

I - Personal project : introduction

→ Goal : collect all the dots

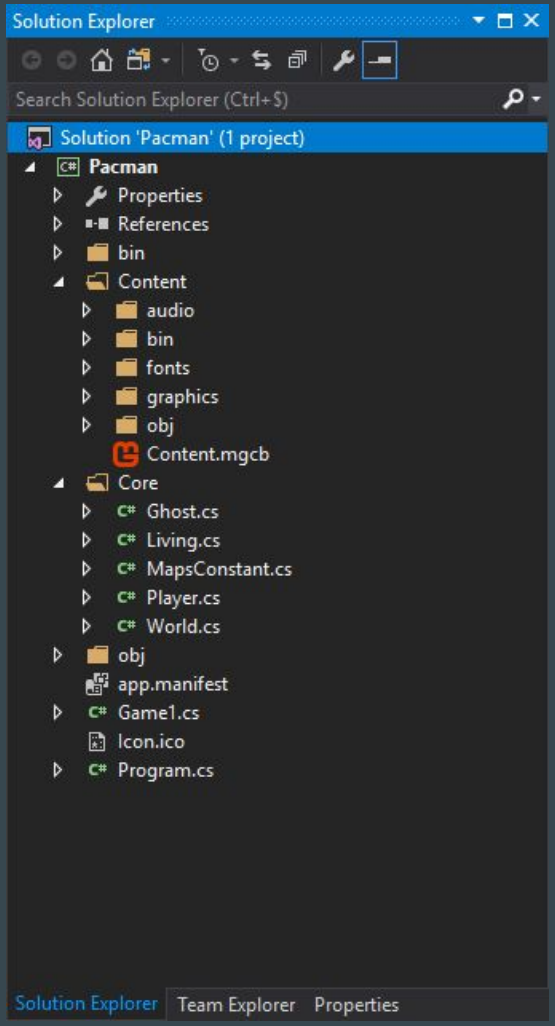
My Pacman



Original Pacman

II - Personal project : development

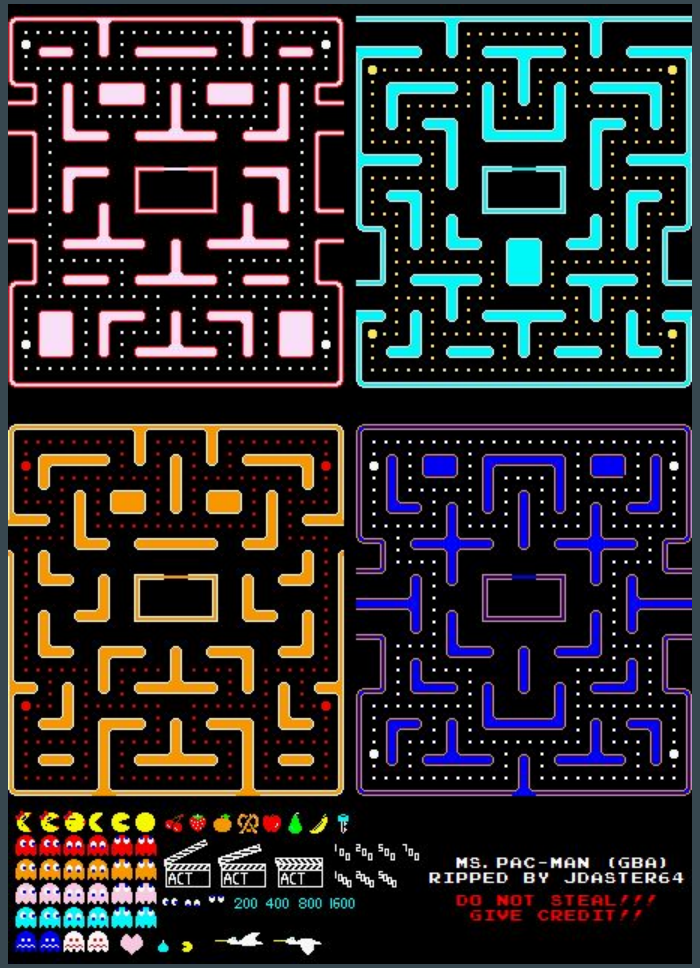
II - Personal project : development



II - Personal project : development

1 - World map

→ Image downloaded from The Spriters Resource

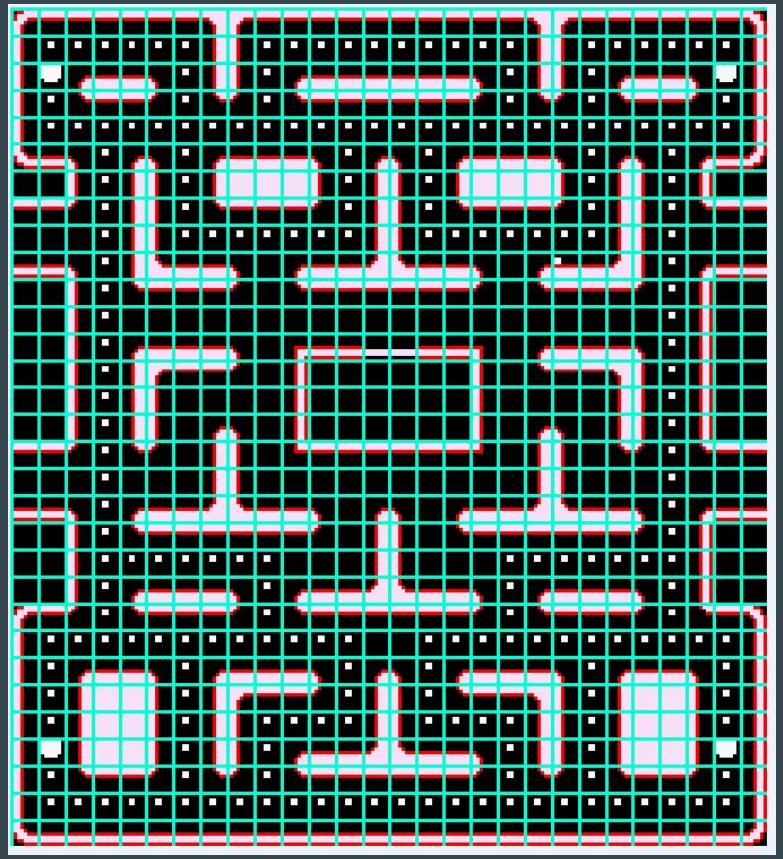


II - Personal project : development

1 - World map

→ World map = grid

→ Change candies for dots

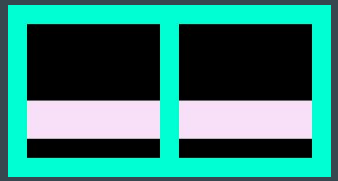


II - Personal project : development

1 - World map

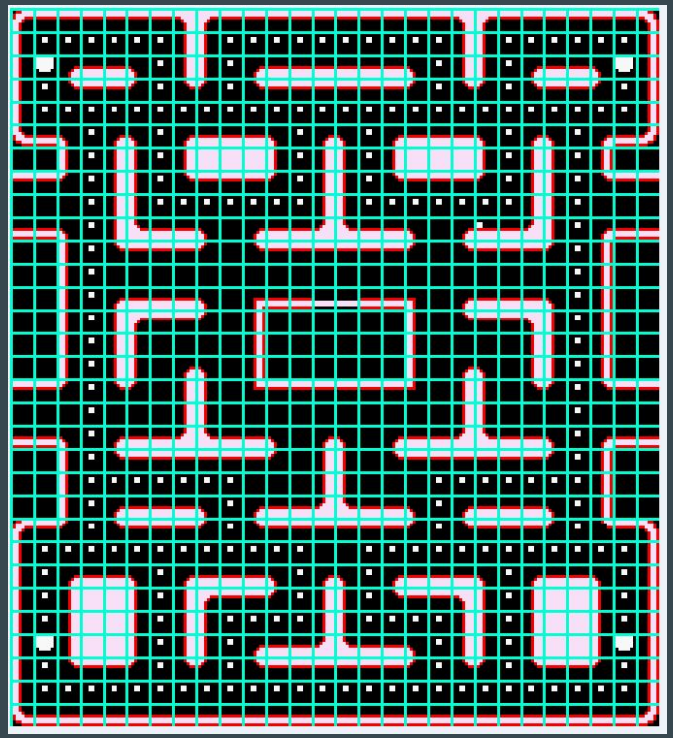
→ Script to convert the image to a C# matrix

→ Almost ok everywhere :



→ A tile :

```
// grid definition
public enum Tile
{
    EMPTY = 0,
    WALL = 1,
    POINT = 2
}
```

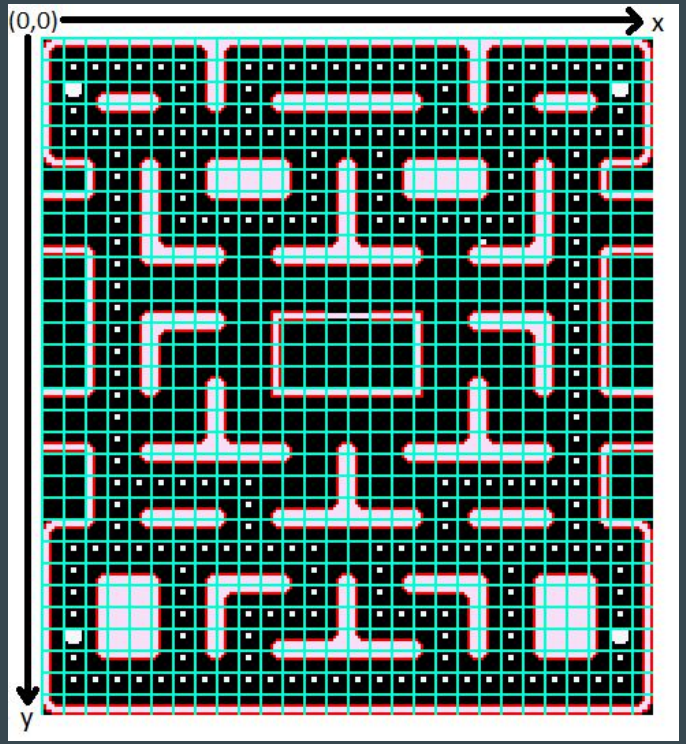


II - Personal project : development

1 - World map

→ Trouble accessing a tile

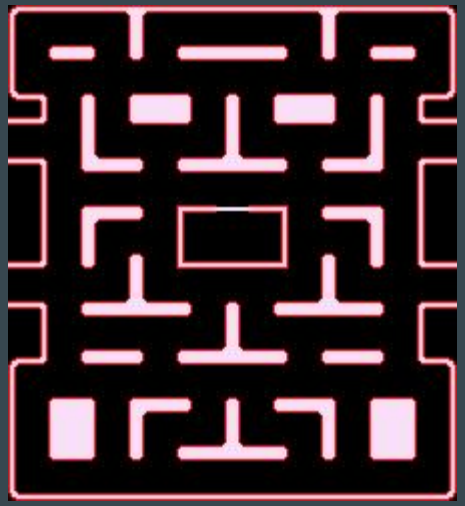
```
/// <summary>
/// Get the value of a tile
/// </summary>
/// <param name="x">the x-th tile from left to right</param>
/// <param name="y">the y-th tile from top to bottom</param>
/// <returns>The value of the tile : Tile.EMPTY, Tile.WALL, or Tile.POINT</returns>
public Tile GetTile(int x, int y)
{
    return matrix[y, x];
}
```



II - Personal project : development

1 - World map

→ Dots removed from the image



II - Personal project : development

2 - Player : Pacman

→ Move Pacman : 4 possible directions

```
public void Move(GameTime gameTime, KeyboardState keyboard, World world)
{
    // if not moving yet
    if (!targetSet)
    {
        if (keyboard.IsKeyDown(Keys.Z))
        {
            // we want to move up
            direction = PossibleDirections.UP;

            // if we can move up
            if (tileY > 0 &&
            world.GetTile(tileX, tileY - 1) != World.Tile.WALL)
            {
                SetTarget(world, tileX, tileY - 1);
            }
        }

        else if (keyboard.IsKeyDown(Keys.Q))...

        else if (keyboard.IsKeyDown(Keys.S))...

        else if (keyboard.IsKeyDown(Keys.D))...
    }
}
```

```
public void UpdatePosition(GameTime gameTime, World world)
{
    // update timer
    msSinceLastUpdate += gameTime.ElapsedGameTime.Milliseconds;

    if (msSinceLastUpdate >= msBetweenTwoUpdates)
    {
        // reset timer
        msSinceLastUpdate = 0;

        if (targetSet)
        {
            // update position
            position.X += (int)velocity.X;
            position.Y += (int)velocity.Y;

            // update collision rectangle
            UpdateCollisionRectanglePosition();

            // if we finished to move
            if (world.GetTileCenterX(tileXTarget, tileYTarget) == getCenterFrameX() &&
                world.GetTileCenterY(tileXTarget, tileYTarget) == getCenterFrameY())
            {
                ClearTarget();

                // we reached the target
                tileX = tileXTarget;
                tileY = tileYTarget;

                // check if it was a point case
                if (world.GetTile(tileX, tileY) == World.Tile.POINT)
                {
                    score += 1;
                    world.SetTile(tileX, tileY, World.Tile.EMPTY);
                    world.NumberPointsLeft -= 1;
                }
            }
        }
    }
}
```

II - Personal project : development

2 - Player : Pacman

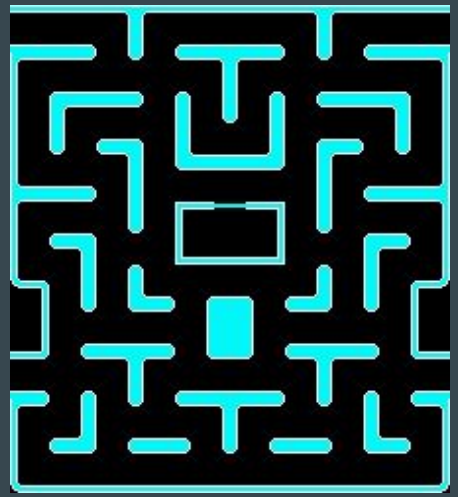
→ Sprite animation :

```
public void UpdateFrame(GameTime gameTime, KeyboardState keyboard)
{
    // update time
    msOnScreen += gameTime.ElapsedGameTime.Milliseconds;

    // update only if the player wants to move
    bool wantsToMove = keyboard.IsKeyDown(Keys.Z) ||
        keyboard.IsKeyDown(Keys.Q) ||
        keyboard.IsKeyDown(Keys.S) ||
        keyboard.IsKeyDown(Keys.D);

    if (targetSet || wantsToMove)
    {
        if (msOnScreen > frameTime)
        {
            // reset timer
            msOnScreen = 0;

            switch (direction)
            {
                case PossibleDirections.UP:
                    if (frameIndex == FramesIndexPlayer.UP_1)
                    {
                        frameIndex = FramesIndexPlayer.UP_2;
                    }
                    else if (frameIndex == FramesIndexPlayer.UP_2)
                        frameIndex = FramesIndexPlayer.UP_3;
                    else
                        frameIndex = FramesIndexPlayer.UP_1;
                    break;

                case PossibleDirections.LEFT:
                    ...

                case PossibleDirections.DOWN:
                    ...

                case PossibleDirections.RIGHT:
                    ...
            }
            // update source rectangle
            sourceRectangle.X = (int)frameIndex * frameWidth;
        }
    }
}
```

II - Personal project : development

2 - Player : Pacman

→ Score and lives : displayed at the bottom of the window



add margin at the bottom

II - Personal project : development

3 - Ghosts

→ Can't die

→ Move randomly

II - Personal project : development

4 - Collision between Pacman and ghosts

→ With collision rectangles

→ On collision : reset positions,

and -1 life for the player

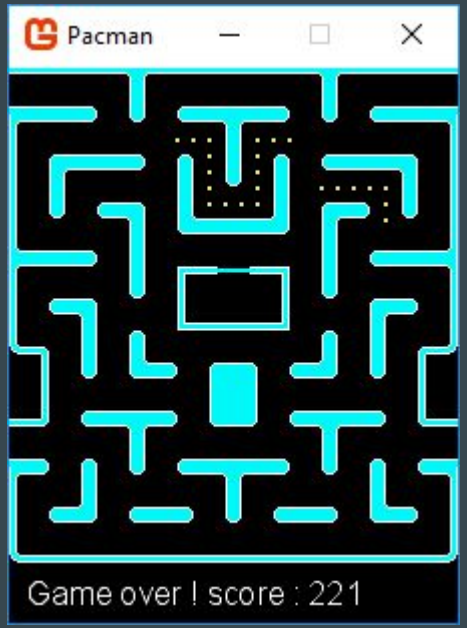
```
// check for collisions
bool hasCollided = false;
foreach (Ghost ghost in ghosts)
{
    if (player.CollisionRectangle.Intersects(ghost.CollisionRectangle))
    {
        hasCollided = true;
    }
}
if (hasCollided)
{
    // reset timer
    msCountDownStartEllapsed = 0;

    // reset player
    player.Reset(world);

    // reset ghosts
    foreach (Ghost ghost in ghosts)
    {
        ghost.Reset(world);
    }

    // play death music
    death.Play();
}
```

After updating positions of Pacman and ghosts (Game1 Update method)

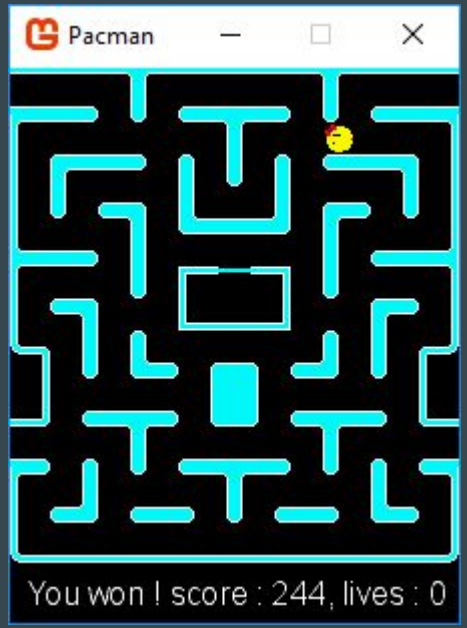
II - Personal project : development

5 - End of the game

→ End when game over or when the player won



Player lost



Player won

III - Let’s play !