UI

(in 3D games)

Integrated / In-World UI

(UI uses X, Y, Z and is placed in the world)

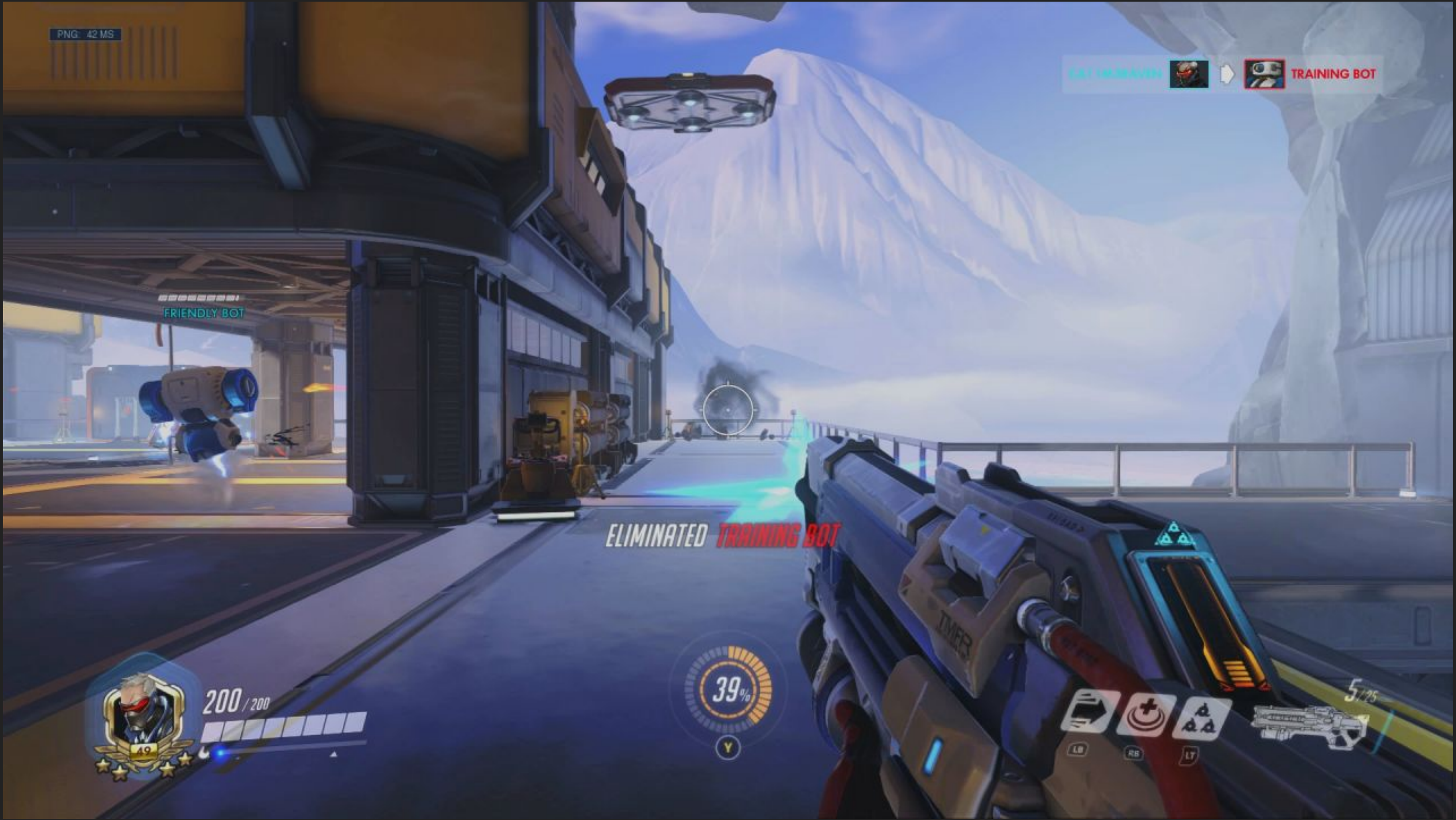
22

36

FLUX DRIVE
ACC CHARGE
THRUST: REQ'D
THRUST: GEN'D
* IMBALANCE *
ACC CHARGE
ACC LEFT LOCK
ACC RIGHT LOCK
FLUX ENABLE
ACC AUTO-BAL
ENG 4
ENG 3
ENG 2
ENG 1
ON
POWER
LOCK
THROT LOCK
LEN REACTOR
CORE TEMP
EM SHIELD LEVEL
BURN RATE
* FUSING *
SYSTEM TEMP
ENGINE SELECTED: 1
PRE-HEAT
FUSE ENABLE
TEMP CO OVRD
LENR ENABLE
FUEL CUT-OFF
FUEL SOURCE
SET
SOURCE TANK: 2
LEVEL
HIGH VOLTAGE BUS
AVAILABLE
FIRE EXTINGUISHER
ENG EJECT
ON
MANEUVERING THRUSTERS
PLASMA P TEMP
EM SHIELD LEVEL
THRUST: GEN'D
POWER: REQ'D
FUEL: REQ'D
SYSTEM TEMP
INJCTR ENABLE
ICH COUPLER
FUEL CUT-OFF
FUEL SOURCE
ON
POWER
XMIT MAX
XMIT HIGH
XMIT NORM
XMIT MIN
XMTR ENABLE
ON
A/P ALLOW
ON
POWER
SELF
TGT
ATT REF
R-IN
R-OUT
CBT MODE
EBP ENBL
BSTR ENBL
RCS ONLY
BOOST DEFL ENABLE
COLD GAS OVRD
IGNITE/ PROVIDE
HOLD

UI Overlay

(what we originally did in 2D)

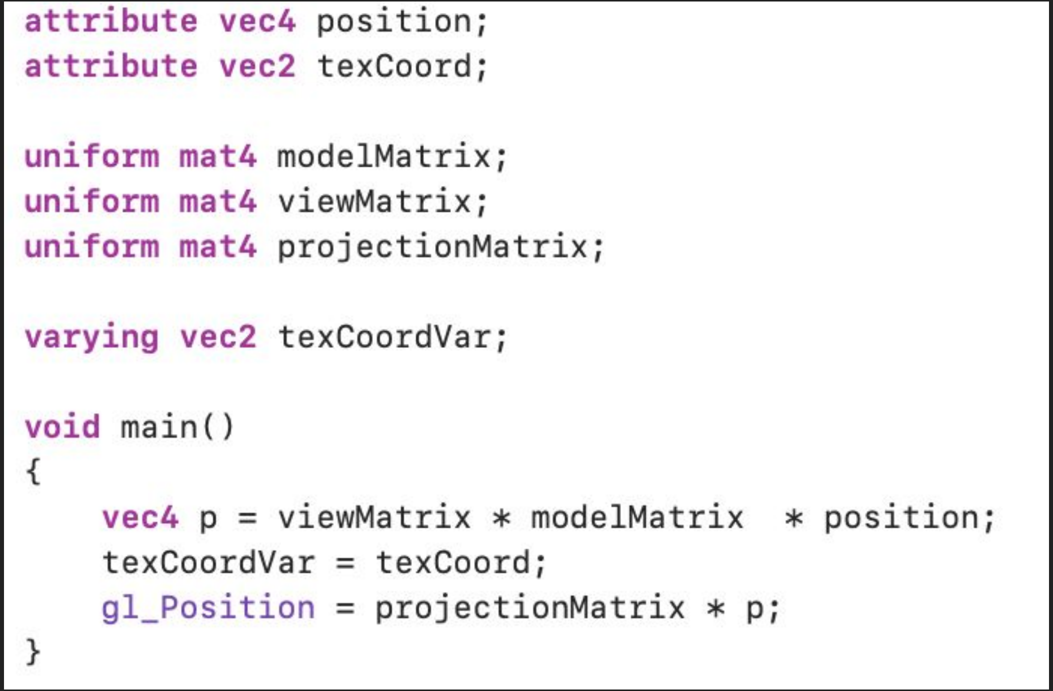

PNG: 42 MS
TRAINING BOT
FRIENDLY BOT
ELIMINATED TRAINING BOT
39%
Y
200 / 200
49
LB
RB
LT
5/25

1st
TIME 01:07:771
1
2
3
4
P1
LAP 2/3
43
mph

Sankar
16
Saldeans Farm
28 min
29

You may have noticed when drawing text in our newer projects, the UI was placed “in world.”

This is due to everything being affected by the projection and view matrices.

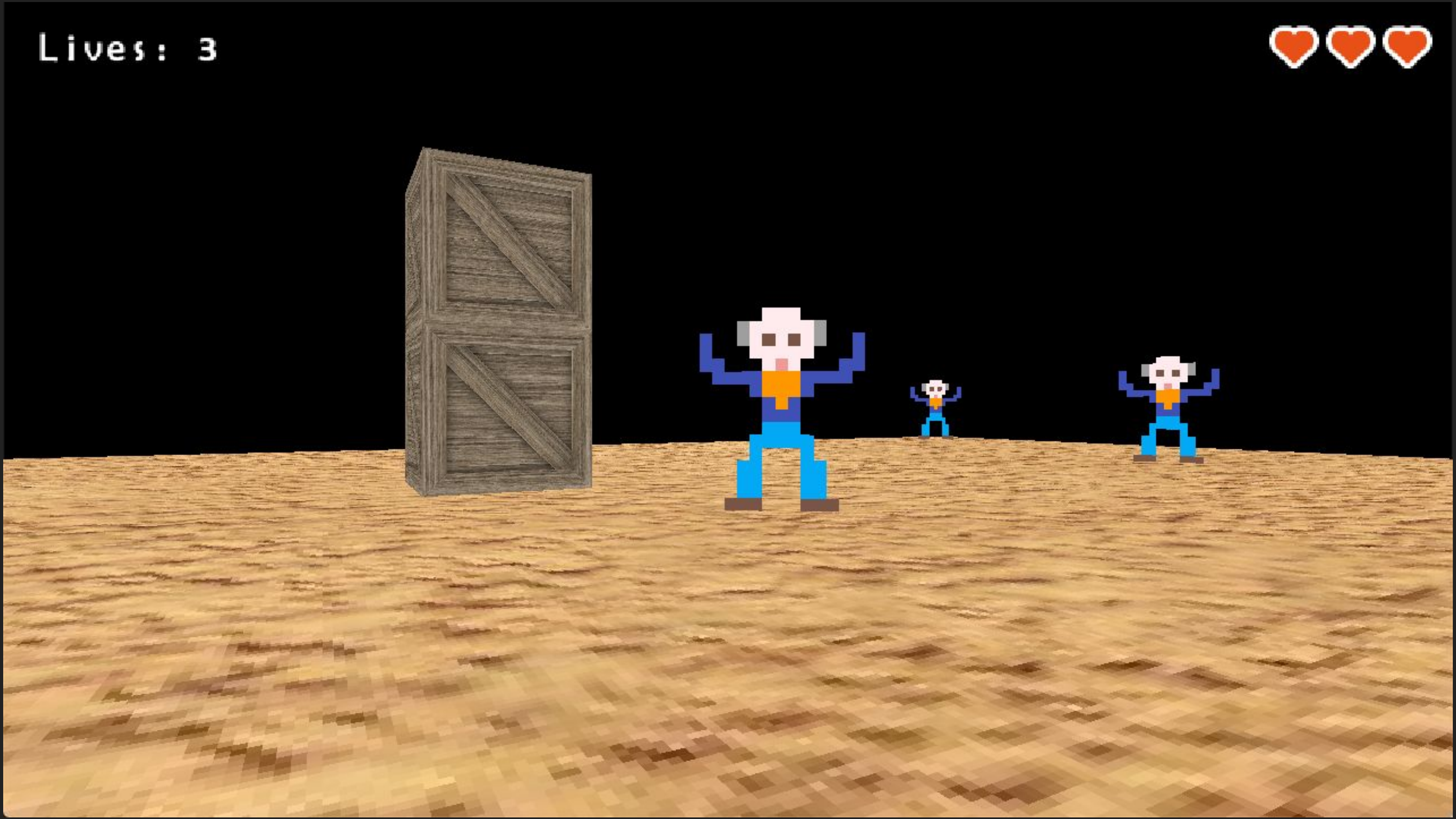


```
attribute vec4 position;
attribute vec2 texCoord;

uniform mat4 modelMatrix;
uniform mat4 viewMatrix;
uniform mat4 projectionMatrix;

varying vec2 texCoordVar;

void main()
{
    vec4 p = viewMatrix * modelMatrix  * position;
    texCoordVar = texCoord;
    gl_Position = projectionMatrix * p;
}
```

How do we break free of the 3D projection matrix and the moving view matrix to draw our UI in 2D?

We add an orthographic projection matrix and a non moving view matrix!

(just like we did way back)


Lives: 3

Update main.cpp

```
// Add to the top
glm::mat4 uiViewMatrix, uiProjectionMatrix;
GLuint fontTextureID;


// Add inside of initialize
uiViewMatrix = glm::mat4(1.0);
uiProjectionMatrix = glm::ortho(-6.4f, 6.4f, -3.6f, 3.6f, -1.0f, 1.0f);



// Make sure you have a font in your project.
fontTextureID = Util::LoadTexture("font.png");
```

Update main.cpp

```
// Top of Render
program.SetProjectionMatrix(projectionMatrix);
program.SetViewMatrix(viewMatrix);


// Draw 3D objects here…


// Once we are done drawing 3D objects...switch!
program.SetProjectionMatrix(uiProjectionMatrix);
program.SetViewMatrix(uiViewMatrix);

Util::DrawText(&program, fontTextureID, "Lives: 3", 0.5, -0.3f,
                                               glm::vec3(-6, 3.2, 0));
```

Let’s Code!

Update main.cpp

Add the UI Matrices
Update Render
Draw Text

What about icons?

❤️ ❤️ ❤️

We can use our usual 2D drawing code!

Update Util.h

```
// Add this definition
static void DrawIcon(ShaderProgram *program, int iconTexture, glm::vec3 position);
```

Update Util.cpp

```
void Util::DrawIcon(ShaderProgram *program, GLuint iconTexture, glm::vec3 position)
{
    glm::mat4 modelMatrix = glm::mat4(1.0f);
    modelMatrix = glm::translate(modelMatrix, position);
    program->SetModelMatrix(modelMatrix);

    float vertices[] = { -0.5, -0.5, 0.5, -0.5, 0.5, 0.5, -0.5, -0.5, 0.5, 0.5, -0.5, 0.5 };
    float texCoords[] = { 0.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0, 0.0 };

    glVertexAttribPointer(program->positionAttribute, 2, GL_FLOAT, false, 0, vertices);
    glEnableVertexAttribArray(program->positionAttribute);

    glVertexAttribPointer(program->texCoordAttribute, 2, GL_FLOAT, false, 0, texCoords);
    glEnableVertexAttribArray(program->texCoordAttribute);

    glBindTexture(GL_TEXTURE_2D, iconTexture);
    glDrawArrays(GL_TRIANGLES, 0, 6);

    glDisableVertexAttribArray(program->positionAttribute);
    glDisableVertexAttribArray(program->texCoordAttribute);
}
```

Update main.cpp

```
// Add to the top
GLuint heartTextureID;


// Add inside of initialize
heartTextureID = Util::LoadTexture("platformPack_item017.png");



// Add inside of Render
for (int i = 0; i < 3; i++)
{
     // These icons are small, so just move 0.5 to the right for each one.
     Util::DrawIcon(&program, heartTextureID, glm::vec3(5 + (i * 0.5f), 3.2, 0));
}
```

Let's Code!

Update main.cpp

Update Util.h and Util.cpp

Add the heart to your project

You can find it here (item 17):

https://github.com/carmineguida/CS3113/blob/master/Assets/2D%20Sprites/Kenney_SimplifiedPlatformer.zip