Camera Battle Transitions

Martí
Davicino
Quevedo

Contents

- What are camera transitions?
- Types of camera transitions
- Camera transitions importance
- Code implementation
- Review exercises

What are camera transitions?

CINEMA VS VIDEO GAMES CAMERA TRANSITIONS

POST PRODUCTION PROCESS

PROGRAMMING PROCESS

VIDEOGAMES ALSO USE CAMERAS TO FILM THE SCENES

Do we need camera transitions?

During the transitions a lot of data is managed

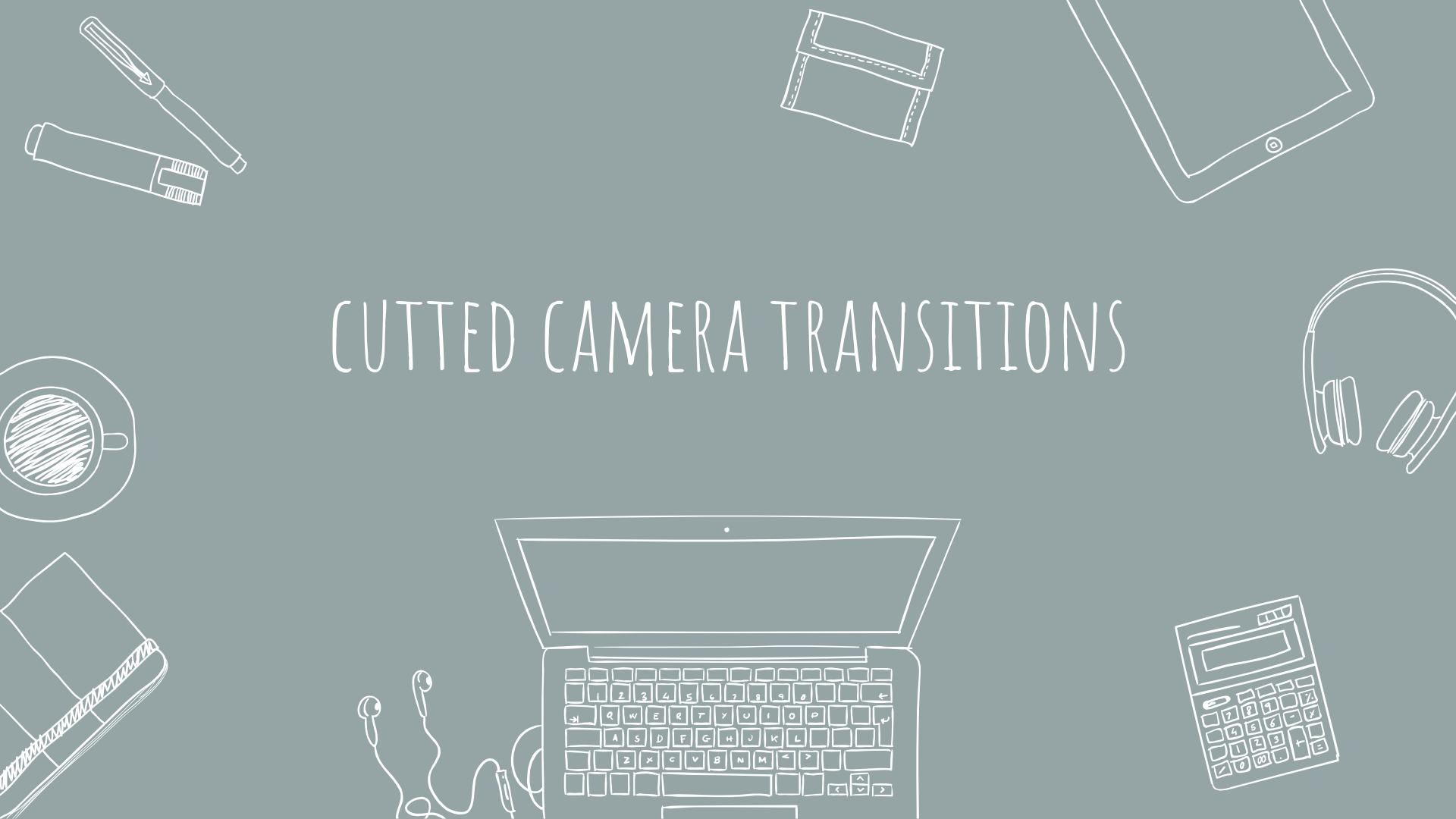
Cutted transitions

Change with two different scenes

Uncutted transitions

Camera relocation to focus the action

CUTTED CAMERA TRANSITIONS



SUDDEN CUT TRANSITION

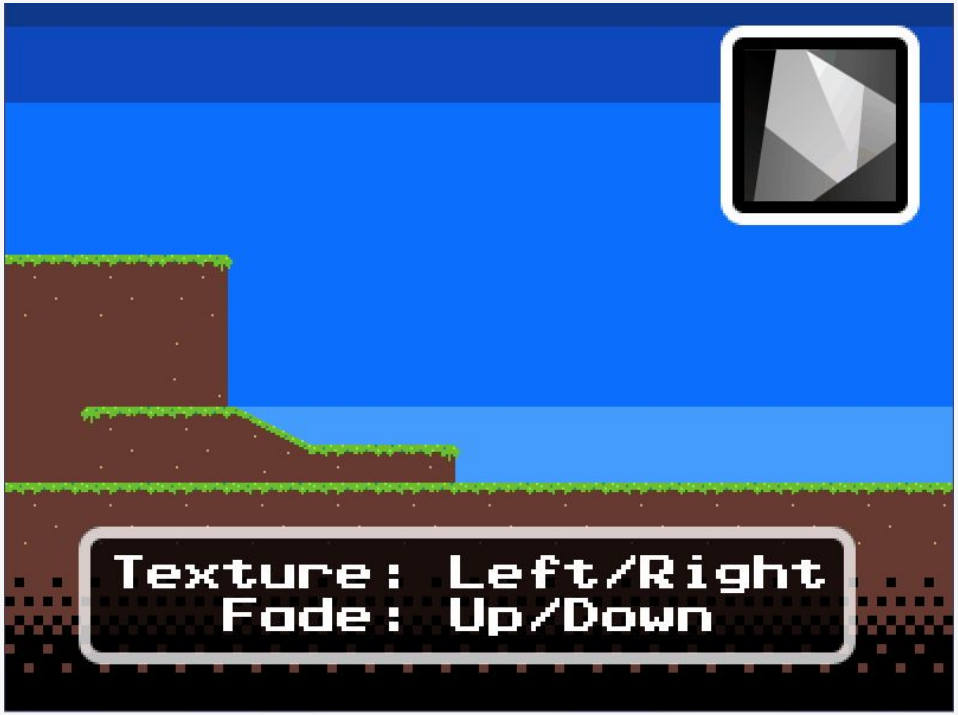
PARTIAL CUT TRANSITION

INVISIBLE CUT TRANSITION

Fade in/out transition

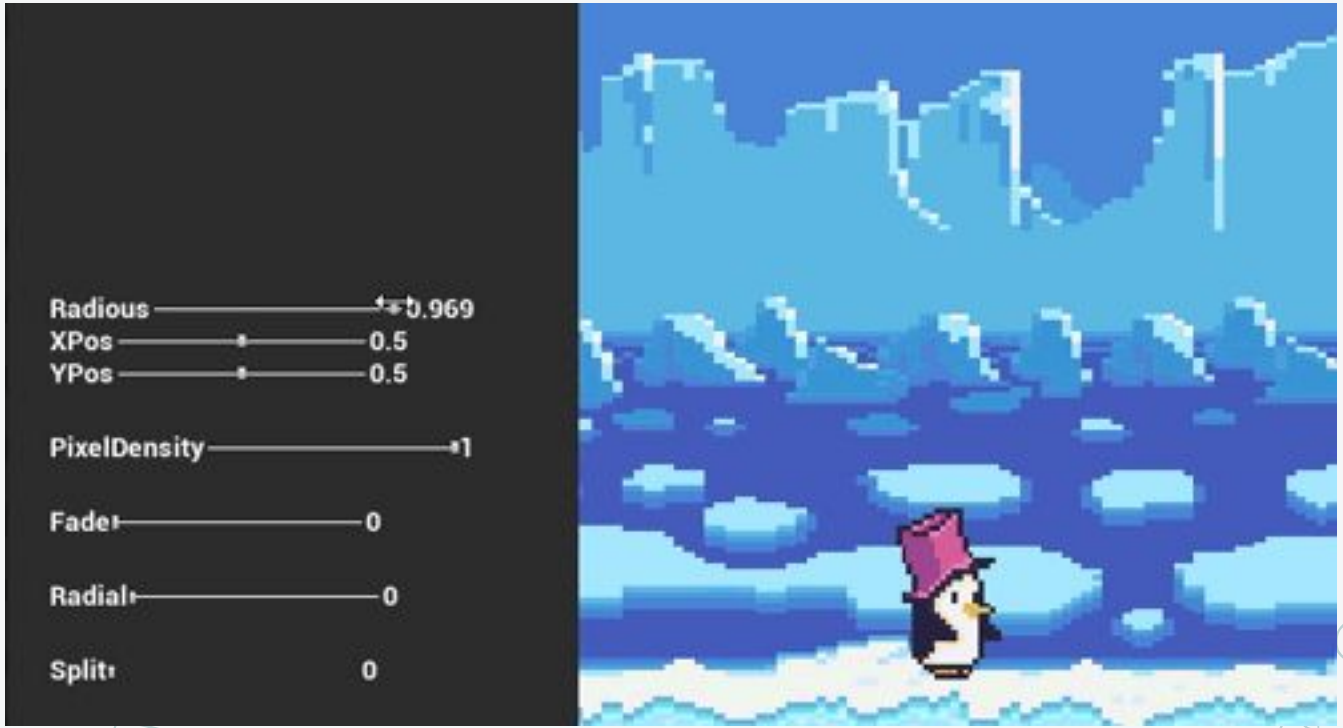
Fade in/out transition

Fade in/out with stencils transition



Fade in/out with stencils transition

Fade in/out with stencils transition

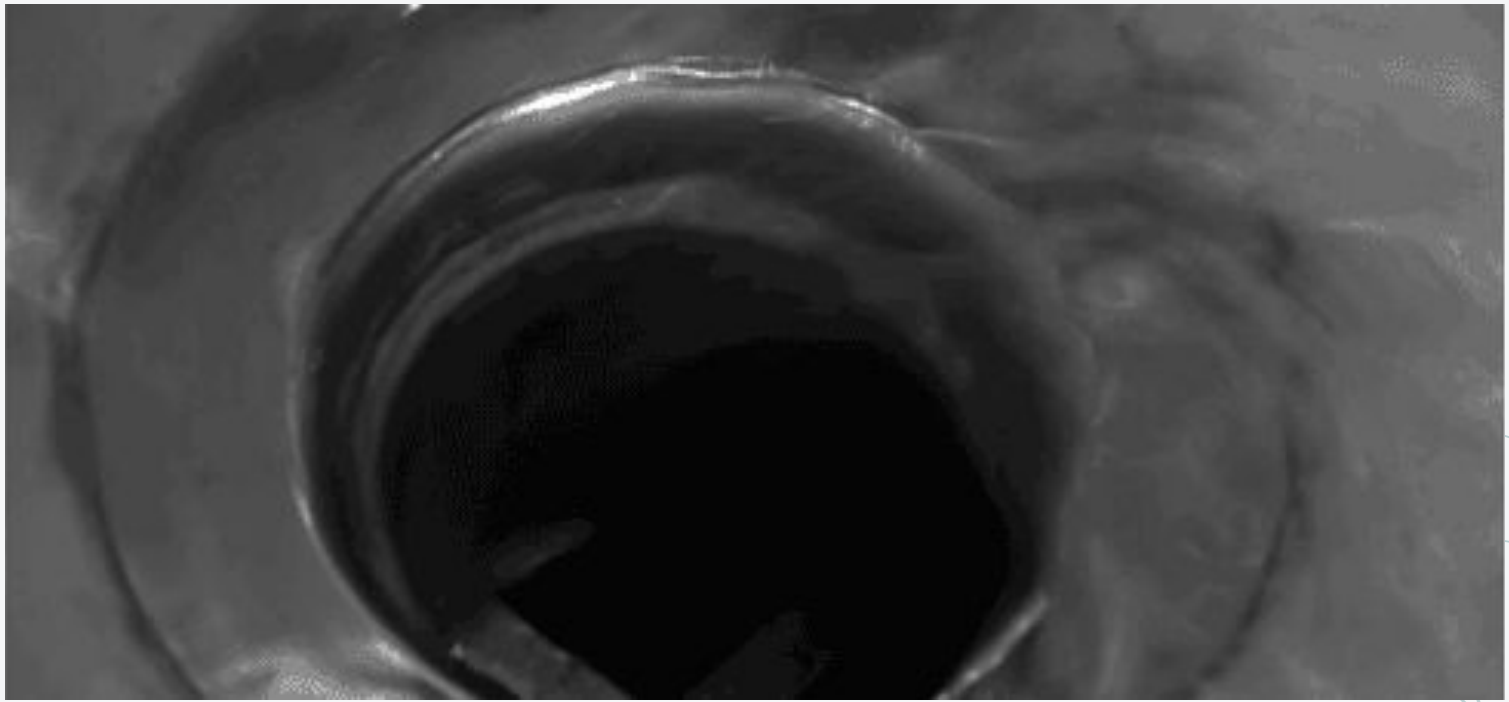


Fade in/out with Animation transition

ZOOM IN/OUT TRANSITION

ZOOM IN/OUT TRANSITION

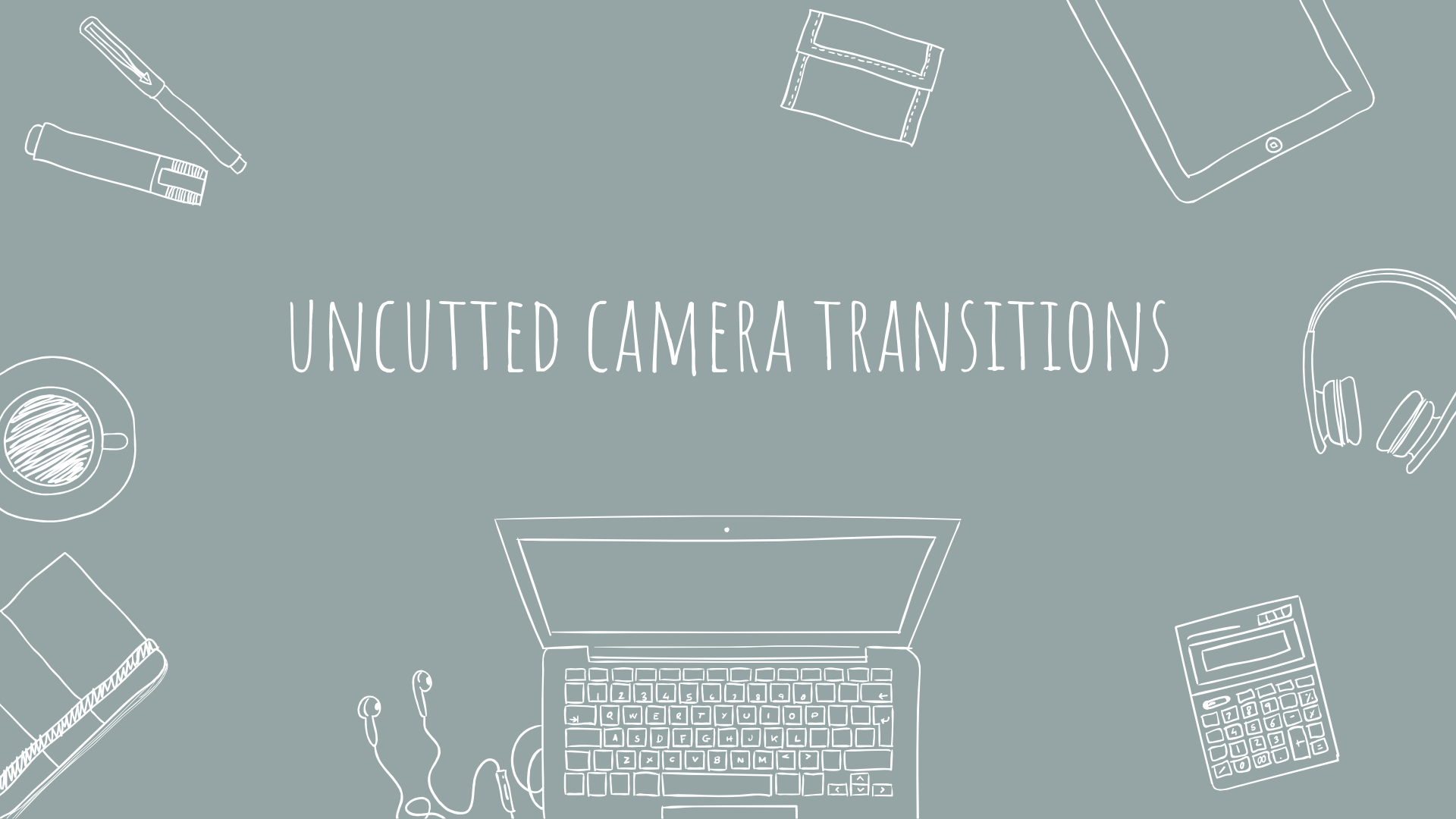
DISSOLVE TRANSITION



WIPE TRANSITION

WIPE TRANSITION

UNCUTTED CAMERA TRANSITIONS



CAMERA LOCKING TRANSITION

Lerp transition

camera translation when player cross the screen boundaries

Lerp transition

The camera is following the action

Lerp transition

The camera is following the action

Lerp transition

Camera rotation according to player actions

INTEGRATED SCENES TRANSITION

INTERFACE RENDERS OVER THE CURRENT SCENE

INTEGRATED SCENES TRANSITION

A DIFFERENT IS CONTAINED IN THE CURRENT SCENE

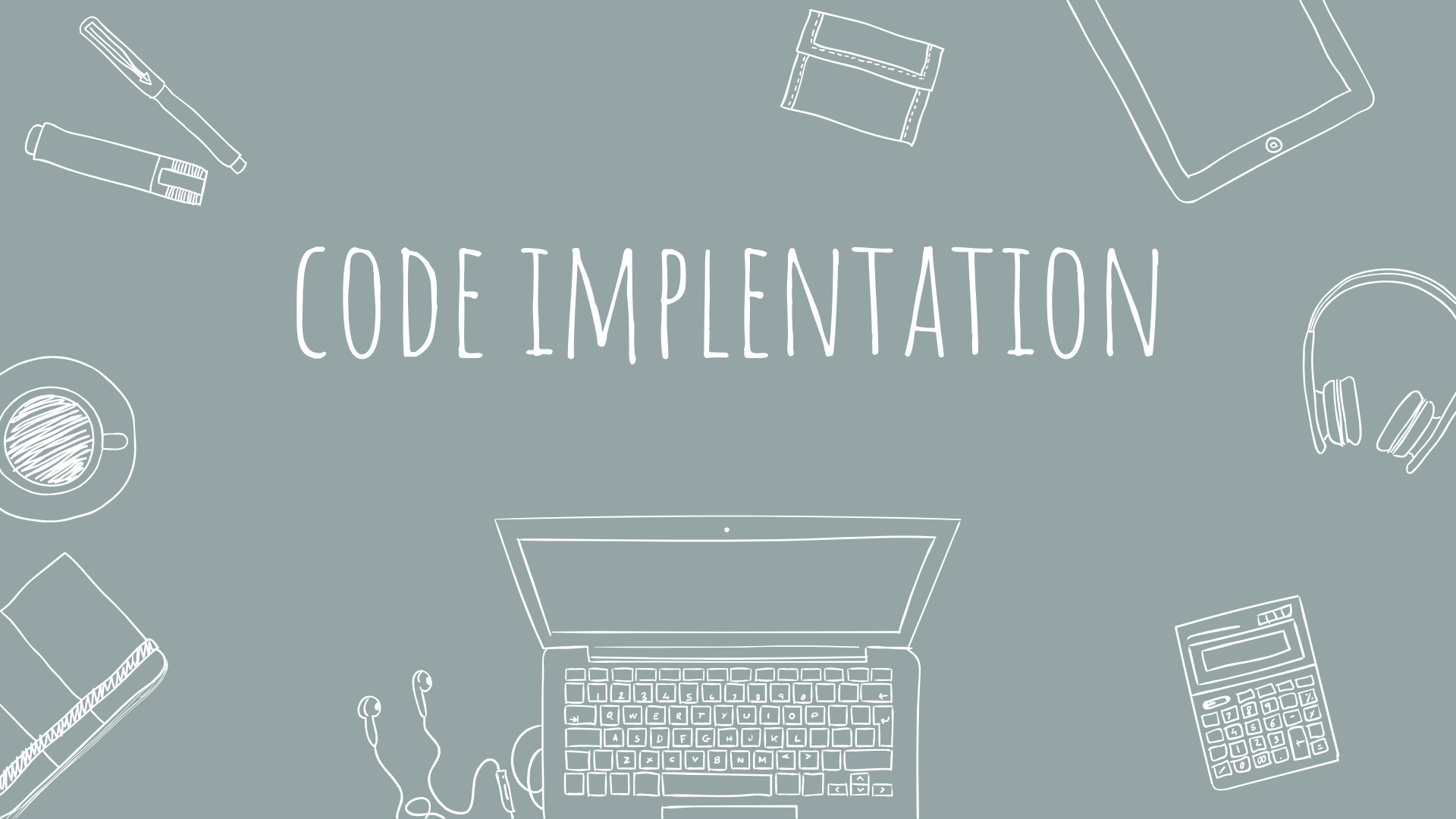
INTEGRATED SCENES TRANSITION

A DIFFERENT IS CONTAINED IN THE CURRENT SCENE

ZOOM AND SHAKE TRANSITION

ALTHOUGH THE ELEMENTS ARE ALREADY SHOWN IN THE SCREEN,
THROUGH A ZOOM AND SHAKE EFFECT WE EMPHASIZE THE ACTION

CODE IMPLENTATION



FADE IN/OUT TRANSITION

```
bool Transition::Update(float dt)
{
    if (currentStep == FadeToBlackStep::NONE)
    {
        return true;
    }

    if (currentStep == FadeToBlackStep::TO_BLACK)
    {
        ++frameCount;
        if (frameCount >= maxFadeFrames)
        {
            currentStep = FadeToBlackStep::FROM_BLACK;
        }
    }
    else
    {
        --frameCount;
        if (frameCount <= 0)
        {
            currentStep = FadeToBlackStep::NONE;
        }
    }
    return true;
}
```

FADE IN/OUT TRANSITION

```cpp
bool Transition::PostUpdate()
{
    if (currentStep == FadeToBlackStep::NONE)
    {
        isFading = false;
        return true;
    }

    float fadeRatio = (float)frameCount / (float)maxFadeFrames;

    // Render the black square with alpha on the screen
    SDL_SetRenderDrawColor(app->render->renderer, 0, 0, 0, (Uint8)(fadeRatio * 255.0f));
    SDL_RenderFillRect(app->render->renderer, &screenRect);

    return true;
}
```

FADE IN/OUT TRANSITION

```cpp
bool Transition::FadeToBlackEffect(bool fadeIn, float frames)
{
    // If we are already in a fade process, ignore this call
    if (currentStep == FadeToBlackStep::NONE)
    {
        if (fadeIn == false)
        {
            isFading = true;
            currentStep = FadeToBlackStep::TO_BLACK;
            frameCount = 0;
            maxFadeFrames = frames;
            return true;
        }
        else
        {
            currentStep = FadeToBlackStep::FROM_BLACK;
            frameCount = frames;
            maxFadeFrames = frames;
            return true;
        }
    }

    return false;
}
```

FADE IN/OUT TRANSITION

```
void Scene::UpdateLogoScene()
{
    if (timer >= 50) app->render->DrawTexture(logo, 0, 0);

    if (timer < 50)
    {
        timer++;
    }
    else if (timer < 80)
    {
        app->transition->FadeToBlackEffect(true, 30.0f);
        timer++;
    }
    else if (timer < 250)
    {
        timer++;
    }
    else if (timer >= 250 && timer < 280)
    {
        app->transition->FadeToBlackEffect(false, 30.0f);
        timer++;
    }
    else
    {
        SetScene(MAIN_MENU);
    }
}
```

Lerp (2d translation) transition

```
CameraTranslation::CameraTranslation(float transition_time, iPoint destination) : Transition(transition_time)
{
	uint w, h;
	App->win->GetWindowSize(w, h);

	origin = { App->render->camera.x, App->render->camera.y };
	this->destination = {(int)(-destination.x + w * 0.5), (int)(-destination.y + h * 0.5)};
}
```

Lerp (2D translation) transition

```
void CameraTranslation::Entering()
{
        Transition::Entering();

        float step_x = LerpValue(percent, origin.x, destination.x);
        float step_y = LerpValue(percent, origin.y, destination.y);


        App->render->camera.x = step_x;
        App->render->camera.y = step_y;
}

void CameraTranslation::SetOriginAndDestination(iPoint origin, iPoint destination)
{
        this->origin = origin;
        this->destination = destination;
}
```

WIPE TRANSITION

```
Wipe::Wipe(float transition_time, bool is_scene_change, int scene_to_change, Color color) : Transition(transition_time)
{
        this->is_scene_change = is_scene_change;
        this->scene_to_change = scene_to_change;
        this->color = color;

        App->win->GetWindowSize(width, height);
        rect = { -(int)width, 0, (int)width, (int)height };
        SDL_SetRenderDrawBlendMode(App->render->renderer, SDL_BLENDMODE_BLEND);
}
```

WIPE TRANSITION

```
void Wipe::Entering()
{
        Transition::Entering();

        float normalized_x_position = LerpValue(percent, -(int)width, 0);

        if (normalized_x_position >= 0)
                rect.x = 0;
        else rect.x = normalized_x_position;

        DrawRect();
}
```

WIPE TRANSITION

```
void Wipe::Action()
{
	Transition::Action();

	DrawRect();

	if (is_scene_change)
	{
		App->scene_manager->ChangeScene(scene_to_change);
	}
}
```

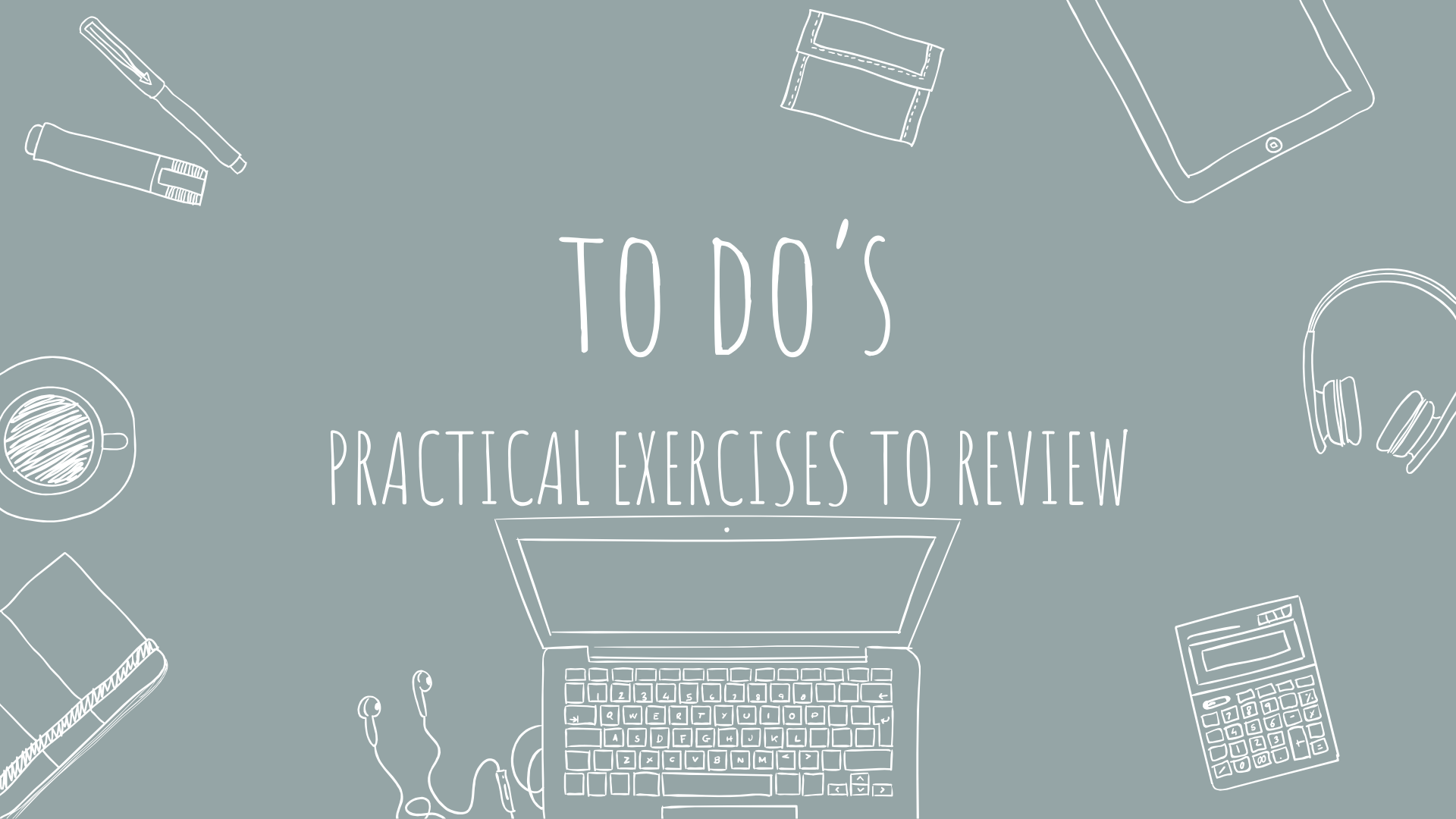
WIPE TRANSITION

```
void Wipe::Exiting()
{
        Transition::Exiting();

        float normalized_x_position = LerpValue(percent, 0, -(int)width);

        if (normalized_x_position <= -(int)width)
                rect.x = -(int)width;
        else rect.x = normalized_x_position;

        DrawRect();
}
```

WIPE TRANSITION

TO DO'S

PRACTICAL EXERCISES TO REVIEW



Exercises

Identify camera transitions

hint: some of them use more than one

What kind of transition is this?

WIPE + FADE OUT TRANSITION

What kind of transition is this?

What kind of transition is this?

ZOOM IN TRANSITION

What kind of transition is this?

What kind of transition is this?

SUDDEN CUT TRANSITION

What kind of transition is this?

INTEGRATED SCENE (UI) TRANSITION

What kind of transition is this?

ZOOM IN/OUT + DISTORT EFFECT TRANSITION

What kind of transition is this?

Lerp transition: camera rotation

What kind of transition is this?

CAMERA LOCKING PLAYER POSITION TRANSITION

What kind of transition is this?

WIPE OUT TRANSITIONS

What kind of transition is this?

LERP TRANSITION: CAMERA TRANSLATION

What kind of transition is this?

LERP (CAMERA TRANSLATION) + ANIMATED WIPE TRANSITION


THE END