

# Procedure and Work Instruction

*Game - Nightmare Seekers*

**Disusun guna memenuhi Kualitas Perangkat Lunak**



Oleh:

**Sion Saut Parulian Pardosi**

# Prosedur dan Instruksi Kerja Pengembangan Game - *Nightmare Seekers*

## 1. Introduction

Dokumen ini bertujuan untuk menjelaskan prosedur dan instruksi kerja dalam mengembangkan game "**Nightmare Seekers**" sebagai bagian dari pengalaman di **Apple Developer Academy** menggunakan pendekatan **Challenge Based Learning (CBL)**. Pengembangan game "Nightmare Seekers" bertujuan untuk memberikan pengalaman imersif bagi pemain melalui lingkungan bertema horor. Proyek ini mengimplementasikan penggunaan Swift, SpriteKit, dan UIKit untuk membuat game berbasis tilt dengan elemen emosional yang berfokus pada ketenangan pemain.

## 2. Purpose

Tujuan dari prosedur ini adalah memastikan pendekatan yang terstruktur dan efisien dalam pengembangan game "Nightmare Seekers," termasuk penerapan Challenge Based Learning (CBL) untuk mengasah keterampilan kolaborasi, pemecahan masalah, dan penggunaan teknologi terbaru selama proses pengembangan. Ini mencakup penyiapan lingkungan, integrasi aset, dan pemeriksaan fungsionalitas guna memastikan bahwa mekanika gameplay, kontrol karakter, dan elemen horor berfungsi dengan lancar.

## 3. Terms and Abbreviation

- **CBL** : Challenge Based Learning, yaitu pendekatan pembelajaran yang menantang developer untuk memecahkan masalah nyata secara kreatif dan kolaboratif.
- **Swift**: Bahasa pemrograman yang digunakan untuk mengembangkan aplikasi iOS

- **SpriteKit:** Framework yang digunakan untuk pengembangan game 2D di iOS.
- **UIKit:** Framework untuk pengembangan antarmuka pengguna dan animasi.
- **Jumpscare:** Teknik untuk mengejutkan pemain dengan elemen yang tiba-tiba muncul di layar.

#### 4. Applicable Documents

- Spesifikasi dan persyaratan proyek game
- Dokumentasi tech stack (Swift, SpriteKit, UIKit)
- Panduan Apple Developer Academy tentang CBL
- Dokumentasi GitHub untuk kontrol versi

#### 5. Method

##### 1. Inisialisasi Proyek:

- Persiapkan perangkat komputer dengan kapasitas RAM minimal 8 GB untuk kinerja optimal.
- Lakukan analisis spesifikasi game, seperti mekanik jumpscare, kontrol karakter, dan elemen horor.
- Siapkan lingkungan pengembangan dengan memasang Swift, SpriteKit, Xcode menggunakan template SpriteKit dan alat pendukung lainnya.
- Tetapkan nama proyek sebagai "Nightmare Seekers".

##### 2. Tech Stack

- Pilih teknologi yang relevan untuk pengembangan game, yaitu Swift sebagai bahasa pemrograman, SpriteKit untuk pengembangan game 2D.

- UI-Kit sebagai bagian dari framework yang digunakan untuk animasi dan interface untuk elemen antarmuka pengguna seperti menu dan tombol di IOS
- Tentukan elemen pendukung lainnya, seperti pustaka grafis dan audio untuk efek horor.

### **3. Environment of Development**

- Pasang perangkat lunak yang diperlukan, termasuk Xcode sebagai IDE utama untuk pengembangan Swift dan SpriteKit.
- Siapkan proyek baru di Xcode dengan menggunakan template game SpriteKit.

### **4. Pembuatan Repositori GitHub**

- Inisialisasi repositori baru di GitHub untuk menyimpan dan mengelola kode game "Nightmare Seekers."
- Tambahkan file README dan file konfigurasi yang diperlukan untuk memudahkan kolaborasi dan dokumentasi.

### **5. Integrasi Aset**

- Impor aset karakter, latar belakang, dan objek (seperti kursi dan hantu).
- Pastikan semua aset disesuaikan dengan ukuran yang sesuai di SpriteKit.

### **6. Desain Konsep Game:**

- Kembangkan ide dasar game yang berfokus pada tema ketenangan di tengah tantangan.
- Buat sketsa awal dari antarmuka pengguna dan elemen gameplay.

## 7. Implementasi Game "Nightmare Seekers"

- Kembangkan logika permainan menggunakan SpriteKit dan Swift, sesuai dengan spesifikasi gameplay tilting.
- Implementasikan kontrol tilt-based untuk pergerakan karakter, di mana pemain dapat mengontrol karakter dengan memiringkan perangkat.
- Tambahkan elemen permainan seperti tantangan, rintangan, dan efek jumpscare.

## 8. Pengujian dan Debugging

- Uji mekanik permainan untuk memastikan kelancaran kontrol dan respons terhadap interaksi pengguna.
- Gunakan simulator iOS dan perangkat nyata untuk menguji pengalaman pengguna dan performa game.

## 9. Dokumentasi Proses:

- Simpan versi kode yang diperbarui secara teratur di repositori GitHub.
- Buat dokumentasi teknis untuk fitur dan fungsionalitas game.
- Catat hasil pengujian, termasuk pengujian perangkat dan hasil observasi.

## 10. Pelaporan dan Tindak Lanjut

- Adakan pertemuan tim untuk membahas perkembangan, tantangan, dan langkah selanjutnya dalam pengembangan game.
- Gunakan alat manajemen proyek untuk memantau tugas dan kemajuan proyek.

---

## WORK INSTRUCTION

### Persiapan:

1. Analisis Proyek dengan memahami spesifikasi dan konsep gameplay "Nightmare Seekers" dengan mengacu pada CBL untuk memaksimalkan kolaborasi dan kreativitas.
2. Pastikan perangkat yang digunakan memenuhi spesifikasi minimum untuk pengembangan.
3. Instalasi Alat Pengembangan:
  - Instal Xcode, pastikan SpriteKit dan UIKit siap digunakan.
  - Siapkan akun GitHub untuk kontrol versi dan kolaborasi tim.

### Pengembangan:

1. Buat Repositori GitHub:
  - Buat repositori baru di GitHub untuk menyimpan seluruh kode dan dokumentasi game.
  - Tambahkan file README dan file konfigurasi dasar, seperti .gitignore.
2. Buat Proyek di Xcode:
  - Buat proyek baru di Xcode menggunakan template SpriteKit
  - Atur modul proyek dan jalankan perintah swift package init jika diperlukan.
3. Impor aset game (karakter, kursi, hantu, dll) ke UI-Kit
4. Implementasi Kontrol Permainan:
  - Implementasikan kontrol tilt-based untuk pergerakan karakter menggunakan sensor akselerometer.
  - Implementasikan kontrol tilt-based untuk pergerakan karakter menggunakan sensor akselerometer.

### 5. Mekanisme Tantangan:

- Tambahkan elemen game seperti kursi dan hantu untuk efek jumpscare.
- Gunakan SpriteKit untuk animasi, suara, dan interaksi elemen-elemen ini sesuai dengan konsep game.

### 6. Kontrol Versi dengan GitHub:

- Commit kode setiap perubahan signifikan untuk menjaga kontrol versi.
- Gunakan branch untuk pengembangan fitur baru, dan buat pull request untuk review sebelum integrasi.

## Pengujian:

### 1. Simulasi Permainan di iOS:

- Uji di simulator iOS untuk memastikan respons kontrol tilt dan animasi berjalan lancar.

### 2. Pengujian di Perangkat Nyata:

- Lakukan uji coba pada perangkat iOS asli untuk validasi interaksi tilt dan mekanik permainan.

### 3. Perbaikan dan Debugging:

- Lakukan debugging di Xcode dan perbaiki jika terdapat bug atau masalah performa.
- Gunakan laporan dari pengujian untuk perbaikan berkelanjutan.

## Dokumentasi:

### 1. Buat Dokumentasi Fitur dan Mekanik Permainan:

- Dokumentasikan setiap fitur gameplay, mekanik tilt, dan elemen jumpscare yang digunakan.

### 2. Catat Hasil Pengujian:

- Simpan hasil pengujian, kesalahan yang ditemukan, dan tindakan korektif di dalam repositori atau alat manajemen proyek.

### 3. Pembuatan Panduan Pengguna:

- Buat panduan pengguna dasar mengenai kontrol permainan dan mekanik gameplay untuk pemain.

### Pengembangan Berkelanjutan:

1. Terapkan CI/CD untuk alur kerja pengembangan yang efisien.
2. Pantau kinerja game untuk memastikan stabilitas dan kualitas pengalaman bermain dan selama tahap Pameran Project.
3. Evaluasi dan Pembaruan Rutin:
  - Adakan evaluasi rutin untuk menilai pengalaman pengguna dan memperbarui fitur gameplay jika diperlukan.