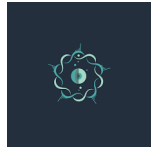


GAME DESIGN DOCUMENT (GDD) TEMPLATE

Title Page

The title page includes general information about the game:

- Game Name :fighter



- Game Logo :
- Game Catch Phrase : "Grad caps in the air, coins to collect, catch them all if you dare!"

Sign-Off

GAME CONCEPT SIGN-OFF

Lead Artist:mahitab mohammed

Lead Designer:mahy moustafa

Lead Programmer:noha ayman ,noha adel

Lead Producer:mostafa gad

Introduction

Game Description

Genre: Adventure

Game Elements:

- Collecting
- Chase
- Combat

Game Content:

- Thriller

Theme:

- Fantasy

Style:

Example:

- Old School

Player:

one

Player Immersion: • This is an attempt to understand what kind of enjoyment the player will receive from the game.

Example:

- Tactical
- Strategy

Reference: "Metal Gear Solid" series

In this game, you play as a clever thief who must sneak past a guard to collect valuable coins. Your goal is to sneak past the guard and collect as many coins as possible without getting caught. As you progress through different levels, the guard will become more vigilant and the obstacles will become more difficult to overcome. Use your skills and power-ups to outsmart the guard and become the ultimate coin collector!

Game Technical

Technical Form: • 3D graphics (Form)

View: • Side-scrolling view: In this view, the camera is positioned to the side of the game world, and the player moves from left to right (or vice versa) across the screen. This view can provide a sense of depth and allow the player to see the environment from different angles.

Platform: • PC

Language: • C#

Device: • PC

Game Sales

Consumer Group: Casual gamers, Hardcore gamers, Fans of stealth games.

Payment:

- This could involve discussions on monetizing the game and receiving payments from customers

Estimated Price:

- This could involve market sizing and market pricing strategies for the game product

Game Atmosphere

Enter a world of shadows and secrets, where every step could mean the difference between success and failure. Sneak past guards, collect coins, and immerse yourself in the thrill of the chase. Can you make it to the end without getting caught?

Game Play

Catch Phrase: "Sneak, collect, and survive! In this thrilling game, your stealth skills will be put to the test as you sneak past guards, collect coins, and avoid being caught. Can you make it to the end with your energy bar intact?"

Gameplay Description: In this game, the player takes on the role of a skilled thief who must navigate through various levels filled with guards and obstacles. The player must use their stealth skills to sneak past guards without being detected, while also collecting coins scattered throughout the levels. If the player is caught by a guard, their energy bar decreases, and they must avoid being caught again or risk losing the game.

To add to the challenge, the guards become more vigilant as the player progresses through the levels, making it more difficult to avoid detection. The player can use power-ups and upgrades to help them in their quest, such as invisibility cloaks or distraction tools.

As the player collects more coins and progresses through the levels, they unlock new challenges and obstacles, making the game increasingly challenging and exciting. With its immersive gameplay and intense action, this game is sure to keep players on the edge of their seats.

Key Features

Key features are a list of game elements that are attractive to the player. It may be a good idea to research the key points below or consult with a professional marketer.

- Number of Levels: 2
- Number of Enemies: 1 / Characters: 1

- Time of Game Play :half hour of fun
- Replayability
- Audio Specifications:coin sound,background music
- Graphic Specifications:Texture quality,Lighting,Character models.
- Device Compatibility: it should be optimized to run on a wide range of hardware configurations, from low-end to high-end systems. The game should be tested on different hardware configurations to ensure that it runs smoothly and without issues.
- Number of Players:1
- Online Activities :high scores
- Number/Type Modes:2
- Marketing Ideas: Social media marketing,Influencer marketing,App store optimization.
- Consumer Group:Casual gamers,Hardcore gamers,Fans of stealth games
- Unique Features:Dynamic environments,Multiple paths
- Merchandising: Create a line of T-shirts,Create a series of collectible figures.

Player Elements

The player elements section lists all the elements that are directly related to the player or serve to benefit the player.

Devise two sets of names for player elements. One set is a generic name (or code) and the other is its game name. Describe the terminology that you use to describe the player's properties.

This is a good place to interact with a graphic designer to ensure the game graphics match the game names. Graphics that will be seen during gameplay should be exhibited here.

Multiplayer issues should also be mentioned here.

- Default (Status): low level,score 0,full energy bar
- Actions: take coins, escape from the guard.
- Information (Status): that he should escape from the guard and collect coin
- Winning: when the player collect all the coins in the game
- Loosing: when the enemy hit him until his energy bar become zero

Player Properties

- Health and lives

- take coins
- escape from the guard

Each property should mention feedback as a result of the property changing!

Player Rewards (Power-ups & Pick-ups):Health Power-ups

User Interface (UI):

Health and energy bars

Coin counter

Levels

Pause menu

Heads up Display (HUD)

Health and energy bars

Coin counter

Mini-map

Player View:



Antagonistic Elements

Guards, Environmental hazards.

Antagonistic Properties

This is a list of properties that antagonistic elements have in common.

Antagonistic List

Detection: The antagonistic elements could have the ability to detect the player's presence

Strength: The antagonistic elements could be stronger than the player

Intelligence: if the guard see the player he hit him.

Artificial Intelligence (AI)

- Normal State: he search for the player
- Detection State: if the player in his line of sight he detect him
- Reaction State: he hit the player until the player died or escape

- End State: the player win or loss

Global Game Elements

.

The Story

In this game, you play as a clever thief who must sneak past a guard to collect valuable coins. Your goal is to sneak past the guard and collect as many coins as possible without getting caught. As you progress through different levels, the guard will become more vigilant and the obstacles will become more difficult to overcome. Use your skills and power-ups to outsmart the guard and become the ultimate coin collector!

Concept Art

Sketches that are used for the concept can go into this section as visual reference. In the case of a brand, certain creative restrictions should be noted here. This is a good place to collaborate with a graphic designer to ensure game graphics match game names.

Audio & Sound F/X

Coin audio

Background audio

Hit audio

Game Architecture

- Title Screen
- Option Screens
- Game level
- End Screen

Technical Document

Technical Document: Fighter

Overview

Fighter is a 3D platformer game where the player must sneak past guards, collect coins, and avoid being caught. The game is designed for mobile devices and is built using the Unity game engine.

Architecture

The game is built using C# programming language and follows an object-oriented programming design pattern. The codebase is organized into several modules, including the game engine, level design, player controls, AI,

sound and music, user interface, saving and loading, and upgrades and progression.

User Interface

The game's user interface includes a HUD that displays the player's health, coins collected, and other relevant information. The main menu includes options to start a new game, continue a previous game, or adjust the game's settings.

Gameplay Mechanics

The player can move left, right, jump, and sneak. The player can also use gadgets such as smoke bombs and distraction items to distract guards and avoid detection. The player must collect coins to unlock new gadgets and abilities. If they are caught by a guard, their energy bar decreases. If the energy bar reaches zero, the player loses the game.

Antagonistic Elements

The game includes guards, dogs, security cameras, and other obstacles that the player must avoid or overcome. The guards have different levels of alertness, and the dogs can detect the player's scent. The security cameras can detect the player's movement and trigger alarms if they detect the player.

Levels and Environments

The game includes several levels, each with a unique environment and layout. The levels include hazards such as electrified floors, poisonous gas, and collapsing ceilings. The environments include different types of buildings, such as museums, banks, and government buildings.

Audio and Visuals

The game's art style is a 3D cartoonish style, and the sound effects and music are designed to create a sense of tension and excitement. The sound effects include footsteps, guard chatter, and alarm sounds. The background music includes orchestral and electronic elements.

Networking and Multiplayer

The game does not include networking or multiplayer features.

Audio Content

Antagonistic Elements

- Type of Sound f/x
- Device Vibration

Global Elements

- Ambient Music

Splash Screens

- Ambient Music

Menus

- Type of Sound f/x

Programming Content

- File Size Restrictions
- File Format Type
- Coding Conventions
- Language/Device Restrictions

Player Elements

- Player
- coins

Antagonistic Elements

- guard

Global Elements

- Global upgrades
- Collectibles

Splash Screens

- Introduction to the game world

Menus

- Level
- Start
- pause

Code Structure

1. Game engine: The game engine would be the core component of the game, providing the framework for the game's mechanics and rendering. This could include features such as physics, collision detection, and AI.

2. Level design: The level design code would be responsible for creating the game's levels and environments, including the placement of guards, coins, and obstacles.
3. Player controls: The player control code would handle the input from the player, such as movement, sneaking, and using gadgets.
4. AI: The AI code would be responsible for the behavior of the antagonistic elements, such as guards and dogs. This could include pathfinding, detection, and pursuit.
5. Sound and music: The sound and music code would handle the game's audio, including sound effects and background music.
6. User interface: The user interface code would handle the game's HUD, menus, and other UI elements.
7. Saving and loading: The saving and loading code would handle the game's save system, allowing players to save their progress and resume at a later time.
8. Upgrades and progression: The upgrades and progression code would handle the player's progression through the game, including the collection of coins and the unlocking of new gadgets and abilities.

Resources

Unity assets store