Technical Design Document

[Incremental Games With Firebase Server]

Content version history

Time	Ver	Author	Changes
22-12-2021	0.1	William	Technical Design
			Document

Table of Content

Template Version history1

Content version history2

Table of Content3

1Introduction4

2Technical Overview4

2.1Target System Requirements4

2.2Tools Used4

2.3Engines & Middleware5

2.4File Format5

2.5Technical Code5

3Technical Feature5

3.1Feature A5

3.2Feature B6

4Technical Design6

4.1System Architecture6

4.2Class Diagram7

4.3Database Diagram8

5Administrative Data8

5.1Server8

5.2Database8

5.30ther8

6Reusable Codes8

6.1FB Module8

6.2Blablabla Module9

70ther Notes9

OBJ

1 Introduction

Incremental Games merupakan sebuah Clicker Game dimana Player diharuskan untuk klik layar mereka sebanyak mungkin agar mereka mendapatkan Gold yang akan digunakan untuk mengupgrade job miliknya sehingga Player mendapatkan lebih banyak Gold. Game ini akan mensimulasikan ilmu matematika di dalamnya, dimana pendapatan Player akan semakin tinggi jika mereka mengklik dengan cepat dan juga mengupgrade job mereka.

1.1Purpose & Objective

Incremental Games ditargetkan untuk pemain casual, terutama penggemar Clicker Game agar dapat dimainkan di sela-sela aktivitas para pemain. Target utama dalam game ini adalah untuk memiliki Gold yang banyak. Pemain akan menggunakan klik pada layar mereka untuk mendapatkan gold, semakin cepat di klik maka semakin cepat pula gold yang didapat, selain itu juga ada fitur upgrade job agar Player dapat meningkatkan value dari Gold ketika diklik.

1.2Project Scope

Pada game ini akan diimplementasikan fitur-fitur berikut:

- Gold bertambah ketika layar di klik
- Animasi yang menarik
- Upgrade Job melalui ScrollView di kanan layar
- Upgrade Job bervariasi dengan harga gold yang setiap level lebih tinggi
- Upgrade gold yang didapatkan saat klik ketika upgrade job lebih tinggi
- Achievement ketika Player unlock job yang baru
- Data akan di Save di Firebase Storage sehingga tidak akan hilang file milik Player

2 Technical Overview

2.1Target System Requirements

Platform/OS: Android 4.1 API level 16

RAM: 512MB

2.2Tools Used

Users	Tool	Use
Prog	Unity	2018.3.3.f1
Prog	Java SDK	jdk_v1.8.0 and jre_v1.8.0

2.3Engines & Middleware

Game ini juga menggunakan Firebase bagian Storage untuk menyimpan data berupa JSON file.

Game ini akan dibuat dengan menggunakan Unity 2D game engine. Physics dalam game pun akan menggunakan fitur Physics2D dari Unity untuk mempermudah pengerjaan.

2.4File Format

Save	.json	Menyimpan Data seperti Gold
Code Script	.cs	Game Script

2.5 Technical Code

2.5.1 Technical specification

Platform	Android Mobile
Programming Language	C#

3 Technical Feature

3.1 Gold bertambah jika Coin di klik

Penjelasan fitur dan problem yang dihadapi, dan solusinya Pada game, ada coin yang besar yang jika di klik maka akan bisa mendapatkan gold.

3.1.1 Solution 1

Komponen UI pada Unity yaitu Panel pada Canvas sebagai Tap Area yang akan di klik.

3.1.2 Solution 2

Komponen Script PointerEventData pada Unity sebagai trigger ketika Tap Area di klik.

3.1.3 Selected Implementation

Solusi yang dipilih dan alasannya

Daerah yang akan dipakai sebagai tempat untuk klik coin adalah separuh layar. Nah dalam hal ini akan menggunakan komponen *UI* seperti *Panel* pada *Canvas* sebesar separuh layar yang akan digunakan sebagai *Tap Area*. Agar Tap Area bisa dikenali maka diperlukan *Script* yang mengandung fungsi *PointerEventData* yang digunakan sebagai *trigger* ketika *Tap Area* di klik.

3.2 Fitur B

Penjelasan fitur dan problem yang dihadapi, dan solusinya

Pada game, akan ada sebuah *ScrollView* di kanan layar dimana isinya akan berupa item-item yang dipergunakan sebagai fitur Upgrade untuk meningkatkan penghasilan gold.

3.2.1 Solution 1

Komponen *UI* pada *Unity* yaitu *Scroll Rect* pada *Canvas* sebagai *Scroll Area* yang memiliki item-item yang akan bisa di klik.

3.2.2 Solution 2

Komponen *Script Serializable* pada *Unity* yang isinya berupa *Name, UnlockCost, UpgradeCost, Output* digunakan sebagai *mockup* item dalam *ScrollView* nantinya.

3.2.3 Selected Implementation

Solusi yang dipilih dan alasannya

Komponen *UI* pada *Unity* yaitu *Scroll Rect* pada *Canvas* sebagai *Scroll Area* yang memiliki item-item yang akan bisa di klik. Komponen *Script Serializable* pada *Unity* yang isinya berupa *Name, UnlockCost, UpgradeCost, Output* digunakan sebagai *mockup* item dalam *ScrollView*. Nantinya item ini akan bisa di edit isi dari itemnya ketika *Serializable* didefinisikan sebagai *List* seperti name Internship, unlock cost 100 upgrade cost 100, output 200.

4 Technical Design

4.1 Class Diagram

AnalyticsManager

LogEvent(string,params Parameter[]) : void LogUpgradeEvent(int,int) : void LogUnlockEvent (int) : void SetUserProperties (string,string) : void

LoadingController

_localButton : Button _cloudButton : Button

Start() · void

SetButtonInteractable(bool) : void

TapText

SpawnTime : float Text : Text

spawnTime : float OnEnable() : void Update() : void

TapArea

IPointerDownHandler : Interfaces OnPointerDown(PointerEventData) : void

Game Manager

AutoCollectPercentage : float

SaveDelay : float ResourcesConfigs : Arrav<ResourceConfig> ResourcesSprites : List<Sprite>

Transform ResourcesParent : Transform ResourcePrefab : ResourceController TapTextPrefab : TapText

Coinlcon : Transform Text GoldInfo : Text Text AutoCollectInfo : Text

activeResources : List<ResourceController>

_tapTextPool : List<TapText> _collectSecond : float saveDelayCounter : float

Start() : void

Update() : void AddAllResources() : void ShowNextResource(): void CheckResourceCost(): void CollectPerSecond(): void AddGold(double) : void CollectByTap(Vector3,Transform) : void

GetOrCreateTapText(): TapText

ResourceConfig : struct

UserDataManager

PROGRESS_KEY : string Progress: UserProgressData

oadFromLocal() · void

LoadFromCloud(System.Action) : IEnumerator

Save(bool) : void GetTargetCloudStorage() : StorageReference HasResources(int) : bool

UserProgessData

Gold : double

ResourcesLevels : List<int>

AchievementController

_popUpTransform : Transform popUpText : Text

_popUpShowDuration : float _achievementList : List<AchievementData>

_popUpShowDurationCounter : float

Update(): void UnlockAchievement(AchievementType,string): void ShowAchivementPopUp(AchievementData): void

AchievementData : class

AchievementType : enum

ResourceController

ResourceButton : Button ResourceImage : Image ResourceDescription : Text ResourceUpgradeCost : Text ResourceUnlockCost: Text

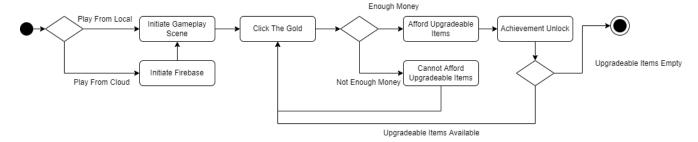
ResourceConfig _config : ResourceConfig

index : int

IsUnlocked : bool

Start(): void SetConfig(int,ResourceConfig): void GetOutput() : double GetUpgradeCost() : double GetUnlockCost : double UpgradeLevel(): void UnlockResource(): void SetUnlocked(): bool

4.2Activity Diagram



5 Administrative Data

5.1Server

Item	Value	Note
Host IP	-	-
Domain name	-	-
Username	-	-
Password	-	-

5.2Database

Item	Value	Note
Host IP	-	-
Domain name	-	-
Server username	-	-
Server password	-	-
DB username	-	-
DB password	-	-

5.3Other

Item	Value	Note
Google store account	=	-
Apple developer id	=	-
Username	-	-
Password	-	

6 Reusable Codes

Dokumentasikan seluruh reusable code dalam project bila ada.

6.1 Modul A

6.1.1 Description

Modul ini digunakan untuk mengimplementasikan dsb

6.1.2 How to Use

_

7 Other Notes

Notes lain yang perlu dicatat dalam project yang ada.

_