

A Cautionary Cookie Clicker

Assignment 1 Incremental Clicker

[Research 4](#_Toc51939509)

[Describe the development process of small sized games/applications 4](#_Toc51939510)

[Preproduction 4](#_Toc51939511)

[Production 4](#_Toc51939512)

[Post Production 4](#_Toc51939513)

[Common Game Engines 5](#_Toc51939514)

[Unity 5](#_Toc51939515)

[Unreal 5](#_Toc51939516)

[GameMaker 5](#_Toc51939517)

[CryEngine 5](#_Toc51939518)

[Lumberyard 6](#_Toc51939519)

[Silly Sam 7](#_Toc51939520)

[Basic Design 7](#_Toc51939521)

[Data Types Used 7](#_Toc51939522)

[Float 7](#_Toc51939523)

[Int 7](#_Toc51939524)

[GameObject 7](#_Toc51939525)

[Button 7](#_Toc51939526)

[Animator 7](#_Toc51939527)

[Bool 7](#_Toc51939528)

[Text 7](#_Toc51939529)

[AudioSource 7](#_Toc51939530)

[AudioClip 7](#_Toc51939531)

[Mathematical Operations 8](#_Toc51939532)

[Playa Script 8](#_Toc51939533)

[Button Visual Effects Script 8](#_Toc51939534)

[Item Availability Script 8](#_Toc51939535)

[Attack Scripts 9](#_Toc51939536)

[Kill Scripts 9](#_Toc51939537)

# Research

## Describe the development process of small sized games/applications

### Preproduction

#### Planning

The following questions need to be asked

* What are we trying to achieve?
* Who is our targeted audience?
* What is our timeline
* What is out budget?
* Market research?
* How big is our staff?
* What is the basic concept?

#### Game Design Document

At this stage it is a good idea to create a Game Design Document, this will include the basic pitch, the high concept and detail the core mechanics of how the game works. The Game Design Document will be adapted over time as the game evolves and develops.

#### Prototyping

This is the point where a prototype of the game is developed implementing the core mechanics, basic art direction and basic gameplay. It is the first initial test of whether the game will work or not work and whether there is significant interest enough to further develop the game.

#### Timeline and Budget

Now is a good time to implement a draft timeline and budget

### Production

Once the Preproduction process has completed it is time to begin the production of the game. This includes developing the code, creating the art assets, creating the audio assets, implementing level design, developing the physics and mechanics. The timeline, budget and GDD should be followed and adapted to changing circumstances. Along the way the game should be tested as early as possible. Creating Milestones to reach is a good way to incentivize production and as a check that the timeline is being achieved.

### Post Production

After the game is launched the next process is fixing bug issues, updating content, possible development of DLC as well as adapting the game to be available for multiple platforms. It is also good to debrief and look at the whole process identifying success and failures in order to make the process more streamlined for the next project

## Common Game Engines

### Unity

#### Description

Unity is a Cross Platform 3D game Engine. It is developed by Unity Technologies and can be used to create 3D games, 2D games as well as VR and AR

#### Languages Used

C#,

The main language that is used for Unity is C#. The implementation is Described in the Unity Scripting API <https://docs.unity3d.com/ScriptReference/> . JavaScript can also be used however this has been depreciated in August 2017

Bolt is a visual scripting language available as a package in Unity. It is useful for non programmers as it allows for a node based graph system which does not require any other knowledge of code.

### Unreal

#### Description

Unreal is a game engine initially written for the 1998 game Unreal and first person shooters It is now used for all sorts of games including RPGs, Platformers and fighting games. It is developed by Epic

#### Languages Used

Unreal Engine initially used Uscript which was influenced by C++ and Java. This was removed and now Unreal is scripted using C++

Visual Scripting is done using BluePrints Visual Scripting Sysetm .

### GameMaker

#### Description

GameMaker is developed by YoYoGames. It is primarily used for games with 2D graphics. It also allows for limited use of 3D graphics . It is mainly uses visual programming and is designed for novice gamemakers with limited to no experience in coding.

#### Languages Used

It uses a visual programming language called DragNDrop as well as its own language called GameMakerLanguage similar to C++ and JavaScript

### CryEngine

#### Description

CryEngine is a multiplatform engine developed by Crytek. Amazons Lumberyard is a reworked and extended version of this engine. It is used across all genres, including first person shooter, sports, MMORPG and virtual reality

#### Languages Used

CryEngine uses C++, C# and Lua a multiparadigm language designed for embedded use in applications.

## Languages

### C#

#### Functionality

* Compiles to CLR which is interpreted by ASP.NET.
* Object Orientated Programming in Classes
* Runs in a virtual machine which automatically handles memory management
* Does not use pointers
* Can create a console, Windows, ASP.net and mobile applications but cannot create standalone

#### Syntax

* No global functions
* Applies Using Statements
* Supports each loop
* Supports Single Inheritance
* Uses references in codes

### C++

#### Functionality

* Compiles into Machine Code
* Object Orientated Programming in Classes
* Requires manual handling of memory management
* Can Use pointers
* Creates standalone applications

#### Syntax

* Has header files
* Supports Multiple Inheritance
* Uses Pointers

### Lua

#### Functionality

* Designed for extending software applications as a general embeddable extension language
* Fast speed and ease of use, not as much implementation of more complex procedures
* No notion of a main program

#### Syntax

* Simple procedural syntax based on arrays and extensible semantics

### GML

#### Functionality

* Designed for extending software applications as a general embeddable extension language
* Fast speed and ease of use, not as much implementation of more complex procedures

#### Syntax

* Similar to JavaScript in Syntax. Simpler and less functionality than C based languages. API is https://docs.yoyogames.com/source/dadiospice/002\_reference/001\_gml%20language%20overview/

# Silly Sam

## Basic Design

Silly Sam is a Cookie Clicker where you pressure Sam to continually take different substances. Clicking on him pushes him which earns the player more Moolah. Each Substance Sam takes affects him differently. As Sam takes each substance, he starts to earn Moolah Automatically. Once the score reaches 1 million, Sam dies.

## Data Types Used

### Float

Floats are used when using a number that has a decimal point. In the game the floats that are used to control Speed, control Timers, control Volume of Sounds. Float is used to be able to more precise in adjusting these values

### Int

Int is used when whole numbers are more appropriate. In the game Int is used to determine how many times a button has been pressed, to count currency, to add to the score.

### GameObject

GameObjects are the main data types used activating objects in the scene, instantiating objects and destroying objects. The objects vary from empty game objects, audio sources, prefabs, canvases. GameObjects are activated, deactivated and destroyed based on different conditions

### Button

All game icons and Sam are buttons. The Scripts listen for button input and then carry out instructions based on button input. These are defined in the Playa Script and the Button Visual Effects Script.

### Animator

These trigger the different animation conditions upon pushing different buttons. These activate and deactivate triggers and bools. In the prototype only one condition for each animator is active at a time. The three animators are on Sam’s face, the game canvas and the main camera

### Bool

Bools are used in the project to determine when a button has been pressed for the first time, as well as to activate the animations.

### Text

The texts are used to change the score and the money counters. These texts exist on the main score counters as well as on each items price across three different modes, Coming Soon, Sold and Button.

### AudioSource

The Audio Source where the audio will be played from. These are in an Empty Game Object labelled audio. There is a button sound, coin sounds, background music and kill screen options

### AudioClip

These define the audio clips to trigger when different buttons are pressed, and currency is added. All audio clips are in the asset folders and these are manually assigned in Unity.

## Mathematical Operations

### Playa Script

#### Moolah

Every time Sam is pushed 1 moolah is added.

#### Score

Every time Sam is pushed 1 score point is added.

* When the score value is greater than or equal to 15, the second instruction is made inactive and the third instruction appears
* When the score value is greater than or equal to 20, the third instruction is made inactive
* When the score is greater or equal to 1 million, the kill screen is activated

#### Button Counter

Every time Sam is pushed 1 button counter is added.

* When the button count is equal to 1 the game music starts and the first instruction appears.
* When the button count is equal to 10, the first instruction is made inactive and the second instruction appears.

### Button Visual Effects Script

#### First Time Counter

Every time a new button is impressed the first time counter is revaluated for the triggers for the next button. This is used to activate the coming soon button for the next item.

### Item Availability Script

#### Item Availability

The first time moola is greater than the item cost, item activated is triggered and the Button appears. This button will disappear and the sold out button will appear every time the moolah counter gets below the amount after that. his trigger will then destroy the coming soon scribble and activate item availability.

#### Button Press

Every time the button is pressed

* The item count is increased by 1
* The amount the item cost is taken away from the moolah counter in the Play Object

#### Item Timer

Once the item count is greater than one

* A timer counts down. This is calculated at the current time multiplied by time.Deltatime( real time). This timer can be changed for each item
* Once the timer hits zero
  + Moolah is increased by (item amount\*multiplier) Multiplier can be defined for each individual item
  + Score is increased by the same as well

### Attack Scripts

#### Expiry Countdown

The attack scripts have a timer countdown by time.deltatime. When the timer reaches the endtime, the item is destroyed

#### Transform

The attack scripts move the item towards a point. This moves at speed \* time.DeltaTime

### Kill Scripts

#### Final Countdown

The Kill Script activates a final countdown. When the countdown reaches zero the game abruptly ends. This should cut off the audio abruptly as well. The final countdown number each frame is multiplied by time.Deltatime.