Morgan Lawrence

Sweets Stacker

A whimsical arcade game

Author: Morgan Lawrence

Date: April 14, 2024

Copyright Information

Version 0.1(draft) October 10, 2005

By Mark Baldwin

Baldwin Consulting

http://baldwinconsulting.org

The Game Design Document (GDD) it the blueprint from which a computer or video game is to be built. As such, every single detail necessary to build the game must be addressed in the document (or support documents). If it’s not in the document, then it probably won’t be in the game.

Below you will find an outline for a generic Game Design Document. The problem is that no generic GDD will be able to address all the various genres for which a game may be created. For example, consider the games PacMan, SimCity and Doom. All three games required detailed design documents, but if you think about it, those documents would be entirely different! As such, when using the outline below you will find sections that will be totally meaningless to your game. But also, there will be sections that your GDD requires to describe the game. Just because it’s not in my outline, it doesn’t mean that it doesn’t belong.

The GDD is a reference document. Members of the development team will constantly be using the document to find specific information for their specific needs. Consider the size such a document may grow to in order to document every piece of the game. We don’t want the GDD to cause information overload and then become a prop under somebody’s wobbly desk. As such it is important that you organize and format the document to make it easy to use. Also note that some of these sections might not appear in the GDD itself but instead would appear in supplemental documents such as an Art Bible or Test Plan. This helps make the overall document more manageable and readable.

One last comment, a game design document is meant to be a living document. Just as when the artist changes the design of his painting every time he takes his brush to the canvas, a computer or video game evolves as code and art are created. The GDD then is the communication tool from which all the members of the team can follow that evolution.

**Document Revision history:**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author(s) | Changes |
| 0.2 | January 16, 2013 | Emile Cormier |  |
| 0.3 | January 05, 2016 | David Burchill |  |
| 0.4 | April 14, 2024 | Morgan Lawrence |  |

*(Use word’s “update table of contents” to generate / regenerate a table of contents. )*

Table of Contents

[1. Section I - Game Overview 7](#_Toc471493942)

[1.1. Game Concept 7](#_Toc471493943)

[1.2. Design History 7](#_Toc471493944)

[1.3. Feature Set 7](#_Toc471493945)

[1.4. Genre 7](#_Toc471493946)

[1.5. Target Audience 7](#_Toc471493947)

[1.6. Game Flow Summary 7](#_Toc471493948)

[1.7. Look and Feel 7](#_Toc471493949)

[1.8. Project Scope 7](#_Toc471493950)

[1.8.1. Number of locations 7](#_Toc471493951)

[1.8.2. Number of levels 7](#_Toc471493952)

[1.8.3. Number of NPC’s 7](#_Toc471493953)

[1.8.4. Number of weapons 7](#_Toc471493954)

[1.8.5. Etc. 7](#_Toc471493955)

[2. Section II - Gameplay and Mechanics 7](#_Toc471493956)

[2.1. Gameplay 7](#_Toc471493957)

[2.1.1. Game Progression 7](#_Toc471493958)

[2.1.2. Mission/challenge Structure 7](#_Toc471493959)

[2.1.3. Puzzle Structure 8](#_Toc471493960)

[2.1.4. Objectives – What are the objectives of the game? 8](#_Toc471493961)

[2.1.5. Play Flow – How does the game flow for the game player 8](#_Toc471493962)

[2.2. Mechanics 8](#_Toc471493963)

[2.2.1. Physics 8](#_Toc471493964)

[2.2.2. Movement 8](#_Toc471493965)

[2.2.3. Objects 8](#_Toc471493966)

[2.2.4. Actions 8](#_Toc471493967)

[2.2.5. Combat 8](#_Toc471493968)

[2.2.6. Economy 8](#_Toc471493969)

[2.3. Screen Flow 8](#_Toc471493970)

[2.3.1. Screen Flow Chart 8](#_Toc471493971)

[2.3.2. Screen Descriptions 8](#_Toc471493972)

[2.4. Game Options 9](#_Toc471493973)

[2.5. Replaying and Saving 9](#_Toc471493974)

[2.6. Cheats and Easter Eggs 9](#_Toc471493975)

[3. Section III – Story, Setting and Character 9](#_Toc471493976)

[3.1. Story and Narrative 9](#_Toc471493977)

[3.1.1. Back story 9](#_Toc471493978)

[3.1.2. Plot Elements 9](#_Toc471493979)

[3.1.3. Game Progression 9](#_Toc471493980)

[3.1.4. License Considerations 9](#_Toc471493981)

[3.1.5. Cut Scenes 9](#_Toc471493982)

[3.2. Game World 9](#_Toc471493983)

[3.2.1. General look and feel of world 9](#_Toc471493984)

[3.2.2. Area #1 10](#_Toc471493985)

[3.2.3. Area #2 10](#_Toc471493986)

[3.3. Characters 10](#_Toc471493987)

[3.3.1. Character #1 10](#_Toc471493988)

[3.3.2. Character #2 10](#_Toc471493989)

[4. Section IV – Levels 10](#_Toc471493990)

[4.1. Level #1 10](#_Toc471493991)

[4.1.1. Synopsis 10](#_Toc471493992)

[4.1.2. Introductory Material (Cut scene? Mission briefing?) 10](#_Toc471493993)

[4.1.3. Objectives 10](#_Toc471493994)

[4.1.4. Physical Description 11](#_Toc471493995)

[4.1.5. Map 11](#_Toc471493996)

[4.1.6. Critical Path 11](#_Toc471493997)

[4.1.7. Encounters 11](#_Toc471493998)

[4.1.8. Level Walkthrough 11](#_Toc471493999)

[4.1.9. Closing Material 11](#_Toc471494000)

[4.2. Level #2 11](#_Toc471494001)

[4.3. Training Level 11](#_Toc471494002)

[5. Section V - Interface 11](#_Toc471494003)

[5.1. Visual System 11](#_Toc471494004)

[5.1.1. HUD - What controls 11](#_Toc471494005)

[5.1.2. Menus 11](#_Toc471494006)

[5.1.3. Rendering System 11](#_Toc471494007)

[5.1.4. Camera 11](#_Toc471494008)

[5.1.5. Lighting Models 11](#_Toc471494009)

[5.2. Control System 11](#_Toc471494010)

[5.3. Audio 11](#_Toc471494011)

[5.4. Music 11](#_Toc471494012)

[5.5. Sound Effects 11](#_Toc471494013)

[5.6. Help System 11](#_Toc471494014)

[6. Section VI - Artificial Intelligence 11](#_Toc471494015)

[6.1. Opponent AI 11](#_Toc471494016)

[6.2. Enemy AI 12](#_Toc471494017)

[6.3. Non-combat Characters 12](#_Toc471494018)

[6.4. Friendly Characters 12](#_Toc471494019)

[6.5. Support AI 12](#_Toc471494020)

[6.5.1. Player and Collision Detection 12](#_Toc471494021)

[6.5.2. Pathfinding 12](#_Toc471494022)

[7. Section VII – Technical 12](#_Toc471494023)

[7.1. Target Hardware and operating system 12](#_Toc471494024)

[7.2. Supported game controllers and peripherals 12](#_Toc471494025)

[8. Section VIII – Game Art 12](#_Toc471494026)

[8.1. Concept Art 12](#_Toc471494027)

[8.2. Style Guides 12](#_Toc471494028)

[8.3. Characters 12](#_Toc471494029)

[8.4. Environments 12](#_Toc471494030)

[8.5. Equipment 12](#_Toc471494031)

[8.6. Cut scenes 12](#_Toc471494032)

[8.7. Miscellaneous 12](#_Toc471494033)

# Section I - Game Overview

## Game Concept

The player must guide Scoopy the dog to catch as many ice cream scoops as possible on his nose, creating a stack of sweets, while dodging bad ingredients. If Scoopy catches a cone, it completes his sweet stack and clears it for more to catch while speeding up the droppable items, increasing the difficulty. If the player avoids any cones, to not complete the stack, the cones become an obstacle on the ground momentarily for Scoopy to jump over. Once the player has helped Scoopy complete 7 stacks of sweets, a finish line appears that the player must stack enough scoops to reach the line to successfully end the game.

## Design History

This is a living document and will change as you develop your game. This section maintains a history of any significant changes to your game.

## Feature Set

Does not apply

## Genre

Arcade

## Target Audience

Children - Young Adults

## Game Flow Summary

Players will load the game to the main menu, where they will be shown the title of the game “Sweets Stacker” and a set of options for user input displaying, “Start Game”, “Controls”, “Credits”, “Exit”. Once the player starts the game by entering the “Start Game” option, the player will be welcomed with their character Scoopy in the middle of the game window to control (using the arrow keys and ‘WAD’ keys) and obstacles falling from above screen. They will also be met with more framing with a score and number of remaining lives on the top of the screen. To progress through the levels, they must catch at least a minimum of one scoop proceeded by a cone which increases the speed of falling items/level. Once seven cones are caught to increase the level to 7, the player must stack scoops to reach this finish line to win the game. Once reached, a winning screen will appear, and they will be sent back to the main menu.

## Look and Feel

Arcade, cartoon, 8-bit low graphic game with bright and vibrant colors for all entities and background. The game appears very fun, both in gameplay and visually, and is enhanced with music and sound effects, making the game feel more immersive.

## Project Scope

Scoopy's main quest centers around stacking as many ice cream scoops on his nose as possible, with each scoop earning the player 50 points. The goal is to dodge falling obstacles and bad ingredients that reduce lives if caught. By catching cones, it earns the player 100 more points, and the end game score is multiplied by the number of cones caught.

### Number of locations

One location – Ice Cream Valley

### Number of levels

Seven levels of progressive difficulty with items falling faster to reach the finish line. Every level after still increases speed difficulty with the reward of an end score multiplier from the number of cones caught.

### Number of NPC’s

Does not apply.

### Number of weapons

Does not apply.

### Etc.

Falling Items include –

Three flavors of ice cream (Strawberry, Chocolate and Vanilla)

Three bad ingredients (Pickle, Hot Chili Pepper and Fish Bones)

One Cone

# Section II - Gameplay and Mechanics

## Gameplay

### Game Progression

The game progression involves guiding your player, Scoopy, across Ice Cream Valley and accumulating scoops of falling ice-cream allows the player to catch a cone and to advance into higher levels of difficulty, until reaching the total seven levels to unlock the finish line. Successful guiding and stacking leads to winning the level when Scoopys stack collides with the finish line, whereas if the player is unsuccessful and loses all three lives, the player loses the game.

### Mission/challenge Structure

Bad items will fall as well as ice-cream and the player must avoid these otherwise, they will lose one of their three lives. When cones are not caught, they fall on the ground and become an obstacle that must be avoid to not lose a life.

### Puzzle Structure

The puzzle of Sweets Stacker is to strategically time and guide Scoopy past different obstacles of varying difficulty.

### Objectives – What are the objectives of the game?

To catch as many scoops as possible, having the highest score possible. Every scoop caught is an added 50 points to the player and every cone caught to complete the stack is an additional 100 points. At the end of the game, all of the player’s accumulated points will be multiplied by the number of cones caught throughout the game.

### Play Flow – How does the game flow for the game player

Falling objects provide the game's pace. The player must adjust their movements to avoid or catch items; when the level is increased, the speed and frequency of the falling items increase too. This causes the user to have to adjust their pace with the game as well. This increases the difficulty of the game and changes the flow for the player to adjust to before being able to reach the finish line after reaching level 7.

## Mechanics

Players can strategically catch ice cream cones to reset the stack and progress to quicker, more rewarding levels. With an additional twist, choosing to stay on slower levels introduces cones as obstacles when they land on the ground that must be jumped over, offering an extra layer of difficulty.

### Physics

There is a gravity system that adjusts and keeps the player downwards if the player has jumped. Player and objects collide on with the ground, causing the player and cones to stay in place and scoops and bad ingredients to disappear.

### Movement

#### General Movement

The player will be able to navigate the world with their character by using their choice of either keys 'W,' 'A, and 'D,' or the left, up and right arrows to move left to right and jump. Additionally, the inclusion of the 'Space Bar' also allows for jumping.

#### Other Movement

Does not apply.

### Objects

#### Picking Up Objects

All objects can be caught with the character’s nose, with the exception of cones unless there’s a scoop already stacked on the character’s nose.

#### Moving Objects

All objects move downwards until collision with the ground or the player’s nose. All moving objects increase in speed and frequency of spawn rate based on the player's level.

### Actions

#### Switches and Buttons

Does not apply.

#### Picking Up, Carrying and Dropping

Falling items are caught when they land on Scoopys nose, this applies to all objects but some must be avoid to avoid losing a life while others are encouraged to be caught like Scoops and cones to earn points.

#### Talking

Does not apply.

#### Reading

Does not apply.

### Combat

Catching bad food items will cause the player to lose a single life on collision with the players nose. Player colliding with cone that have collided with the ground will also cause the player to lose a life.

### Economy

Does not apply.

## Screen Flow

### Screen Flow Chart

The first is the Main Menu Screen, which allows access to the Game, Controls Screen, and Credits Screen. Access to the game loads into the game level. The second screen is accessed during gameplay in the game level which is the Pause Screen. The final set of screens are the Game Over Screen and the Winning Screen on games end.

### Screen Descriptions

#### Main Menu Screen

Allows the user to access the game, the credits screen, the controls screen or quit the game. Footer displays the menu controls.

#### Credits Screen

Displays the creator's name and contact info with a footer to display option to go back to main menu.

#### Controls Screen

Displays the controls and game guide with a footer to display option to go back to main menu.

#### Winning Screen

Displays the players score then proceeds to send them to the main menu.

#### Pause Screen

Displays the option Resume or quit the current game session. Controlled using mouse input.

#### Game Over Screen

Displays the players score and proceeds to send them to the main menu.

#### Etc.

## Game Options

Does not apply.

## Replaying and Saving

Player may replay as many times desired by going back to the main menu and choosing the option to play. Saving does not apply.

## Cheats and Easter Eggs

Does not apply.

# Section III – Story, Setting and Character

## Story and Narrative

### Back story

This game portrays a simple adventure for a dog. The dog, Scoopy, uses his love for ice cream to complete the challenge of stacking as many scoops as possible using only his nose. Sweet Stacker tells the story of Scoopy's sweet tooth adventures for the tastiest ice cream tower.

### Plot Elements

Scoopy likes ice-cream but is not a fan of toppings, like fish bones, hot chili peppers and pickles so he avoids them at all cost! Otherwise, they cause Scoopy stomach aches, but he can’t eat ice-cream without a cone, he loves the crunchy shells!

### Game Progression

As the game progresses Scoopy tries his best to catch more ice-cream otherwise they will be wasted when they fall on the floor.

### License Considerations

Does not apply.

### Cut Scenes

#### Cut scene #1

##### Actors

Does not apply.

##### Description

Does not apply.

##### Storyboard

Does not apply.

##### Script

Does not apply.

#### Cut scene #2

etc.

## Game World

### General look and feel of world

The environment of Sweet Stacker radiates bright and vibrant colours while maintaining its cartoonish look. The initial feel of the game is like a retro video game, simple yet enjoyable

### Area #1

#### General Description

The backdrop consists of vibrant mountains of past ice cream scoops that have fallen and accumulated behind the player. This environment not only captivates visual appeal but sets the player in the world our main character would live in.

#### Physical Characteristics

Ice-cream valley consists of mountains of caramel ice-cream, covered in chocolate fudge rivers and sprinkles throughout the land. Large scoops of ice-cream cover the mountains with varying flavors.

#### Levels that use area

This area encompasses every level of Sweets Stacker. If you win or lose you will always play in this area.

#### Connections to other areas

Does not apply.

### Area #2

Does not apply.

## Characters

### Character #1

Scoopy

#### Back story

Scoopy is a young puppy who followed his sweet tooth all the way to Ice Cream Valley! Scoopy’s journey consists of fulfilling his sweet tooth by catching as many ice cream scoops as he can every day while trying to avoid a stomachache from any bad ingredients as toppings.

#### Personality

Scoopy’s personality is a cheerful young dog who follows his heart. He keeps his head up no matter what falls at him!

#### Look

Scoopy is a white dog with a back full of strawberry syrup covered in rainbow sprinkles.

##### Physical characteristics

Scoopy is a large dog, similar to a husky, with four legs, a big bushy tail and a big nose.

##### Animations

Scoopy’s animations are associated with his ability to run left to right. When the game is over, scoopy lays down on the floor.

#### Special Abilities

Scoopy can jump, allowing him to jump over fallen cones.

#### Relevance to game story

Scoopy holds the game's story with his love for sweets.

#### Relationship to other characters

Does not apply.

#### Statistics

Scoopy’s success is measured in receiving points for successful catches of sweets and cones and a multiplier of points by the number of cones he has caught in the current game.

### Character #2

Does not apply.

# Section IV – Levels

## Level #1 - 7

### Synopsis

Scoopy must catch as many scoops as possible without losing all three lives.

### Introductory Material (Cut scene? Mission briefing?)

Does not apply.

### Objectives

Catch as many scoops as you want before catching a cone, once a cone is caught then the level progresses and the items fall at increasing speeds.

### Physical Description

Scoops, bad items and cones falling from the sky. When cones hit the ground, they become obstacles.

### Map

Ice-cream Valley in background, ground where player walks along.

### Critical Path

Must collect scoops, avoid bad items and cones on the ground.

### Encounters

Bad items will hurt the player if caught in Scoopy’s stack and so will cones on the ground if collided with player. Cones must be jumped over if they’re on the ground.

### Level Walkthrough

Does not apply.

### Closing Material

Does not apply.

# Section V - Interface

## Visual System

### HUD - What controls

The user is provided with a HUD on the top of the game screen to show the current score of the player as well as three health icons in the form of normal dog treats inside square HUD boxes for visibility.

### Menus

Main menu which the player can play, quit or access the credits and controls using the keyboard to select and option. The pause menu allows the user to resume play or quit back to the main menu using the mouse click to select an option.

### Rendering System

2D pixel art rendered using SFML library.

### Camera

Camera is stationary throughout the whole game and displays the full view of the area.

Falling items are dropped from above the cameras view.

### Lighting Models

Does not apply.

## Control System

In Sweets Stacker the input relies on the player’s keyboard to allow movement to Scoopy. The Up Arrow, W and space move Scoopy up to jump, the Left Arrow and D move Scoopy to the left, and finally the Right Arrow and A move Scoopy to the right. Also, the player can pause the game by entering P on their keyboard during gameplay.

## Audio

The game implements spatial audio for dynamically adjusting sound distribution based on the player's position on the screen. Audio is added to enhance the player's experience and outline when positive or negative actions are made.

## Music

Sweets Stacker features background music in both the menu and in-game, which depicts the feel of the game. The music is cheerful to fit the lighthearted atmosphere of the gameplay.

## Sound Effects

Sound effects will vary depending on the actions preceding it. Scoopy will whimper or bark when he eats bad food or hits cones on the floor. Splat noises are made when the scoops are correctly caught on his nose while catching a cone after a scoop will provide the player with a lick sound for Scoopy eating.

## Help System

Does not apply.

# Section VI - Artificial Intelligence

## Opponent AI

Does not apply.

## Enemy AI

Does not apply.

## Non-combat Characters

None for this level. Future implementations may add ingredients that Scoopy can catch for bonus points or power up.

## Friendly Characters

Does not apply.

## Support AI

### Player and Collision Detection

### Pathfinding

Does not apply.

# Section VII – Technical

## Target Hardware and operating system

Windows 10 or 11

## Supported game controllers and peripherals

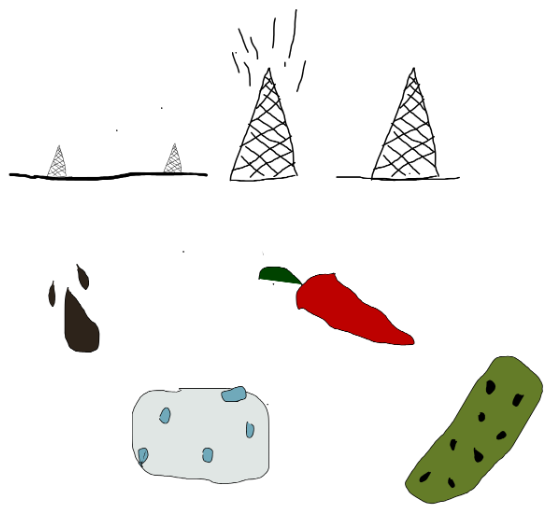
Does not apply.

# Section VIII – Game Art

Only show concept art here (rough sketches)

## Concept Art







## Style Guides





## Characters



## Environments



## Equipment

Does not apply.

## Cut scenes

Does not apply.

## Miscellaneous







