



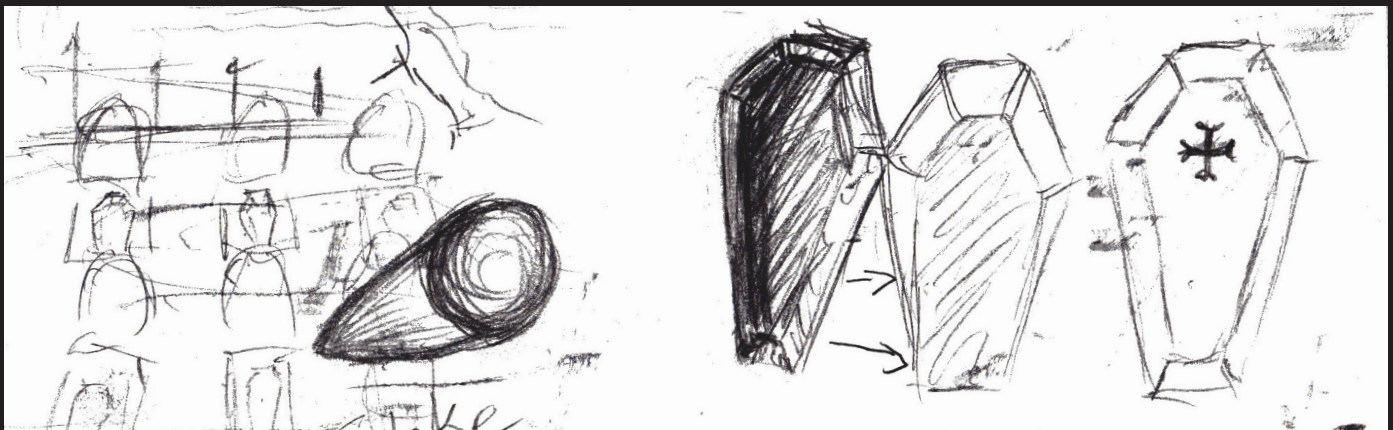
PROCESS BOOK TO

# STAKE-A-VAMPIRE

A SIMPLE GAME

ITGM 719: Scripting for Interactivity  
Professor: Josephine Leong  
Term: Spring 2013

**ALICIA SCARBOROUGH**



# TABLE OF CONTENTS

PROJECT DESCRIPTION.....	P. 3-4
OUTCOMES.....	P. 3
OVERVIEW.....	P. 3
GRADING CRITERIA.....	P. 4
PERSONAL GOALS.....	P. 5
PROJECT SYNOPSIS.....	P. 5
TECHNICAL.....	P. 5
DESIGN.....	P. 5
INTERACTION.....	P. 5
NEEDS OUTLINE.....	P. 6-11
ESSENTIAL ARTWORK NEEDS.....	P. 6
SCREENS.....	P. 6
NAVIGATION BUTTONS.....	P. 6
ANIMATIONS AND GRAPHICS.....	P. 6
ADDITIONAL ARTWORK/MEDIA WISH LIST.....	P. 7
SOUNDS.....	P. 7
ARTWORK.....	P. 7
BONUS ITEMS.....	P. 7
ESSENTIAL PROGRAMMING NEEDS.....	P. 8
INITIAL GAME SETUP.....	P. 8
HOME SCREEN.....	P. 8
LEVELS SCREEN.....	P. 8
HOW TO PLAY SCREEN.....	P. 8
END GAME SCREEN.....	P. 8
PLAY SCREEN.....	P. 9
USER INTERACTION AND AI.....	P. 10
USER INTERACTION.....	P. 10
ARTIFICIAL INTELLIGENCE (AI).....	P. 10
CONDITIONAL EVENTS.....	P. 10
ADDITIONAL PROGRAMMING WISH LIST.....	P. 11
ADDITIONAL CHARACTERS.....	P. 11
FLOW CHARTS.....	P. 12-15
MAIN.....	P. 12
MENUS.....	P. 13
GAME.....	P. 14
GAME PLAY.....	P. 15
CONCEPT SKETCHES.....	P. 16-17
DESIGNED ASSETS.....	P. 18-19
DESIGN USER INTERFACE.....	P. 20-23
START SCREEN.....	P. 20
HOW TO PLAY SCREEN.....	P. 21
LEVELS SCREEN.....	P. 22
PLAY SCREEN.....	P. 23
END GAME SCREEN.....	P. 23
FINAL THOUGHTS.....	P. 24

# PROJECT DESCRIPTION

## Outcomes

After completing this project, students will be able to outline, prototype, finalize and critique a small-scale game or experimental interactive design that demonstrates master of the following ActionScript programming concepts:

- custom functions
- event handling
- transformation of object properties
- arrays
- custom classes
- random-number generation
- built-in ActionScript functions
- collision detection

## Overview

For your final project, you can choose to develop a small Flash-based game or an experimental “plaything.” A plaything is a term used in this course to describe an experimental, interactive toy. A plaything is not a game that can be won or lost. It simply offers innovative, visually interesting environment for exploration. Both project types must demonstrate master of the same minimum number of programming concepts as stated in the Grading Criteria section of this document.

# PROJECT DESCRIPTION

## Grading Criteria

Your project must demonstrate effective application to the following ActionScript elements:

- *Custom Functions* - These only include functions that you name and assign actions to yourself. It does not include functions that are “built-in” to ActionScript.
- *Events*—These include any necessary event handlers for your program to work.
- *Transformation of Physical Properties*—These include any of the following: x position, y position, scale, width, height, alpha, and visibility.
- *Arrays*
- *Custom Classes*—you need to create at least one custom class to define the behavior of a movie clip in an external AS file.
- *Random Number Generation*
- *Collision Detection*
- *Built-In ActionScript functions*

Your final project grade will be scored on a 100-point scale. Twenty of these possible points are reserved for evaluating your project’s sense of cohesiveness and completeness. To receive full credit in this area, your project must show commitment to a theme.

In addition, all functionality that is essential for your project to be tested must work properly. For example, a memory game does not necessarily need a scoring mechanism or timer for someone to use it. However, it must allow the user to flip over cards and have matching cards be recognized. Without this essential functionality, the game’s shortcomings prevent the user from interacting with the game in a most basic way.

For each of the eight required ActionScript elements you use correctly, you will receive 10 points. You can demonstrate a ninth element for an extra 10 points. You are free to use as many scripting elements as you like to create your project, but only nine can be identified as the elements you wish to have graded.

# PERSONAL GOALS

## **Project Synopsis**

For this project I chose to create a whack a mole game that used vampires instead of moles. The vampires would pop out of the coffins and the player would have stake them with a stake to “whack” the vampire. Players would have a quota (number of vampires to stake) to meet before time runs out so that they can advance to the next level or unlock a level to play at a later time.

## **Technical**

This project allowed me to stretch my legs with ActionScript and really get into the concepts of Object Oriented Programming thus allowing me to make a more robust game in a small time frame.

## **Design**

The initial idea for this project was to have simplistic type vampires that moved and looked like puppets popping out of the coffins. This part was the most fun to create out of the entire project.

## **Interaction**

To add challenge to the game I decided to design and program it as if it were to be implemented as a Facebook application.

# NEEDS OUTLINE

## Essential Artwork Needs

### Screens

1. Home Screen
2. Levels Screen
3. How to Play Screen
4. Play Screen
5. End Screen

### Navigation Buttons for:

- Home Screen
  - Play
  - Levels
  - How To Play
- Levels
  - Level 1 - click to play level one
  - Level 2 - click to play level two
  - Level 3 - click to play level three
  - Level 4 - click to play level four
  - Back to Home
- How to Play
  - Play
  - Back to Home
- Play
  - Start Over
  - End Game
    - Back to Home
    - How to Play
- End Screen:
  - Back to Home
  - Play Again

### Animations and Graphics

1. a stake
2. a vampire to stake
3. coffins for the vampire to pop out of
4. a dust poof animation when the vampire is staked
5. stage design: an ambient dungeon for the background
6. a clock timer
7. a score card

# NEEDS OUTLINE

## Additional Artwork/Media Wish List

### Sounds:

- Ambient vampire music to run in the background (added)
- A turn to dust poof sound (added)

### Artwork

#### *Additional vampires to stake:*

- Sparkle vampire - Twilight (added)
- Nosferatu vampire (added)
- Depressed vampire - Interview with a Vampire (Anne Rice's Vampires) (not added)
- Muppet vampire - similar to the sesame street's count (added)
- Dark and foreboding vampire - like the ones in Underworld (not added)

### Bonus items

*These items were not added due to time constraints but will be added in the future.*

- Garlic +10pts and coffin closer - it keeps a particular coffin closed thus reducing the possible coffins that vampires will spring out of Works for ten seconds.
- Holy Water +15pts - kills all vampires in the open
- Cross +20pts and coffin closer - Keeps two coffins (beside each other) closed. Works for twenty seconds.

# NEEDS OUTLINE

## Essential Programming Needs

### Initial Game Setup

#### Home Screen

##### Add play button

apply the listener that will take the player to start a new game on the play screen

##### Add how to play button

apply the listener that will take the player to the how to play instruction screen

#### Levels Screen

##### Add Levels 1-3 buttons

Apply the listener that will take the player to the appropriate level to play IF the level is not locked

***This feature has been changed to be four levels and the locked levels have been removed. All levels should be accessible to play.***

##### Add back to Home Buttons

apply the listeners that will take the player back to the home screen

### How to Play Screen

#### Add Back to Home Button

apply the listener that will take the player back to the home screen

#### Add Play Game button

Apply the listener that will take the player to start a new game on the play screen

### End Game Screen

#### Add Play Again Button

Apply the listener that will take the player to start a new game on the play screen

#### Add Back to Home button

apply the listener that will take the player back to the home screen



# NEEDS OUTLINE

## Play Screen

### Add Exit Game button

Apply the listener that will ask the player if they would like to stop playing

*Remove the “Are you sure” screen and just sent them to the end game screen.*

### Add start Over button

apply the listener that will restart the game on this screen

### Set counter to 2 minutes (this is the default for all the levels)

Changed the counter to be 25 seconds.

### Set score to 0 (the score can carry over from previous levels if the player hits the requirements for the next level for continuation of the game)

### Ensure that all coffin movie clips are at the beginning point

Determine the number of coffins to show based on level

*Originally had a grid but decided to make varying patterns to randomize and display based on level.*

### Select initial coffins for vampires to spring out of

Randomly select a coffin

Set has a Vampire variable to true

### Change mouse cursor to stake

Hide the mouse cursor

Add the stake graphic to stage and have it follow the mouse coordinates with onMouseMove event

# NEEDS OUTLINE

## Essential Programming Needs

### User Interaction and AI

#### User Interaction

**Add listener to mouse clicks to remove successfully staked vampires or get booster to help in the game - complete and it uses mouse and collision detection for the stake and the character movie clips.**

#### Artificial Intelligence (AI)

**Randomly choose a coffin to show a vampire //Can show an empty coffin too - complete**

**Animate the coffins and vampires in and out of the scene - complete**

#### Conditional Events

**Detect if there is still time left on the counter:**

**If there is time then start/continue opening coffins**

**Detect if Coffin Opens**

**Detect if Coffin has a vampire**

**Show Vampire - or - show nothing**

**Detect collision of stake and vampire:**

**Has vampire been staked when its coffin was opened?**

**Remove vampire and play poof animation**

**Add Score**

**Does the Vampire have a time bonus?**

**Add time**

**If timer reaches zero and the user has not met the required amount of kills - End Game**

**Show “Times Up” - removed this and just took the player to the end game screen.**

**Go to End Screen**

**Show score and how many vampires total were staked**

**Show average and total score.**

# NEEDS OUTLINE

**If timer reaches zero and they met the minimum amount of vampire kills to reach the next level**

## **Proceed to the next level**

Carry the score over to the next level  
- complete

Keep tally of total vampires staked -  
complete

Reset level tally of vampire staked to  
0 - complete

Reset the time to 2 minutes - revised  
to be 25 seconds instead.

**If not then end the game when the time is up**

**Show “Times Up”** - *removed please see previous notes on this.*

## **Go to End Screen**

Show score and how many  
vampires total were staked

*Added the average and the total  
score to this screen*

**Is level unlocked** - *removed; decided that this was a nice to have quality*

**Trying to exit game** - *removed decided that this was a nice to have quality.*

## **Additional Programming Wish List**

### **Additional Characters**

*These additional characters can either give boosters or apply a time penalty if staked by accident*

**Alternate vampires - staking certain vampires can give more points, a time bonus or can freeze time**

1. Sparkle vampire - Twilight +50pts
2. Nosferatu vampire +30pts
3. Depressed vampire - removed
4. Muppet vampire - +100pts
5. Dark and foreboding vampire - removed

***Werewolves - removed***

***Blade - deducts points if staked***

***Priests - deducts points if staked***

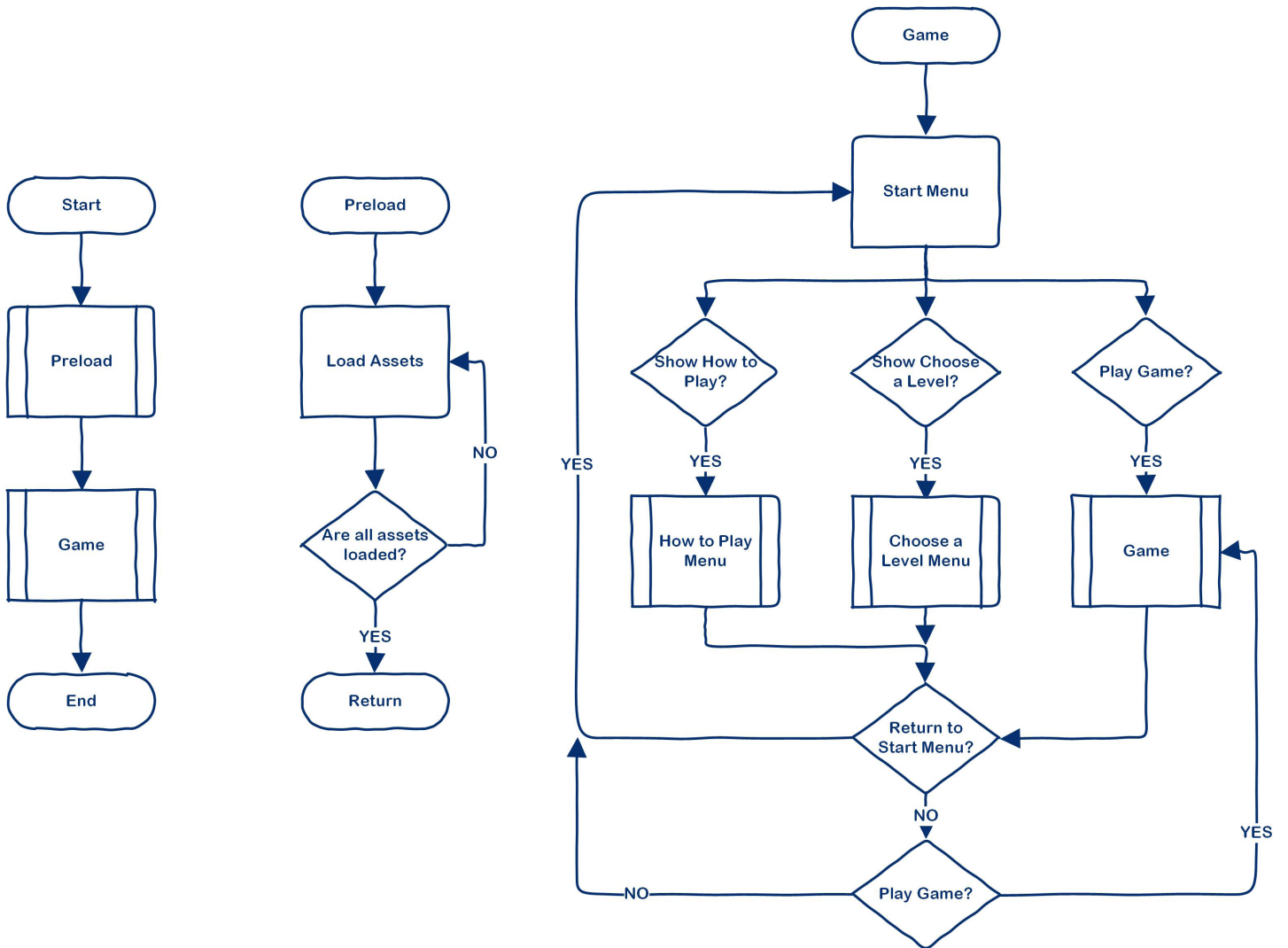
**Bonus items - removed but will be added at a later revision of the game**

Garlic and Cross - Coffin closers that work for twenty seconds

Holy Water +150pts and kills all vampires out in the open

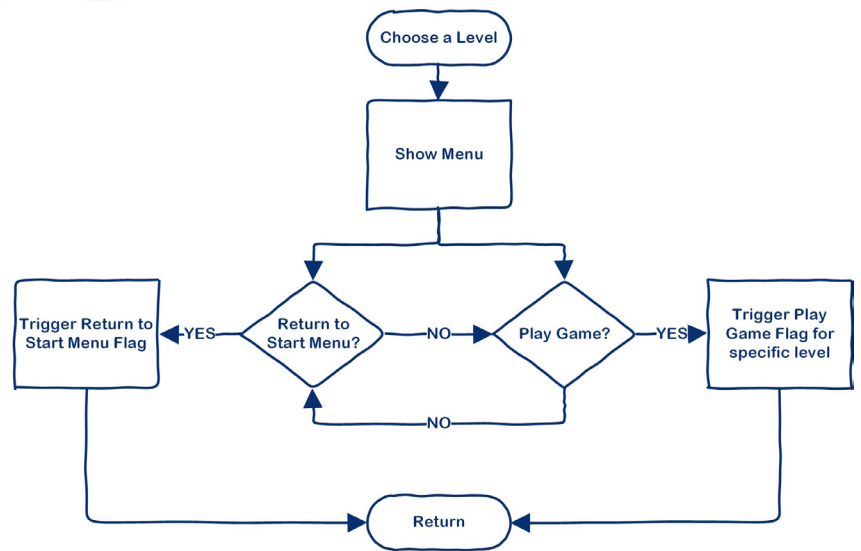
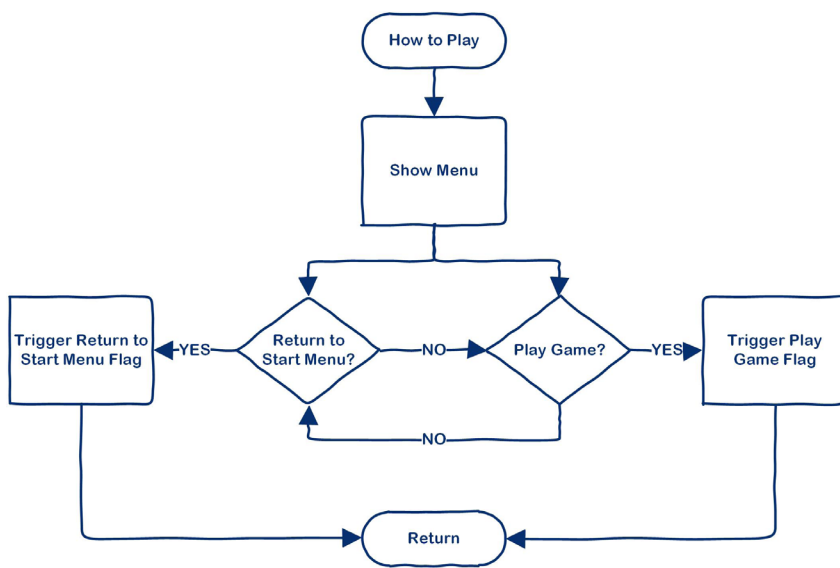
# FLOW CHARTS

## Main



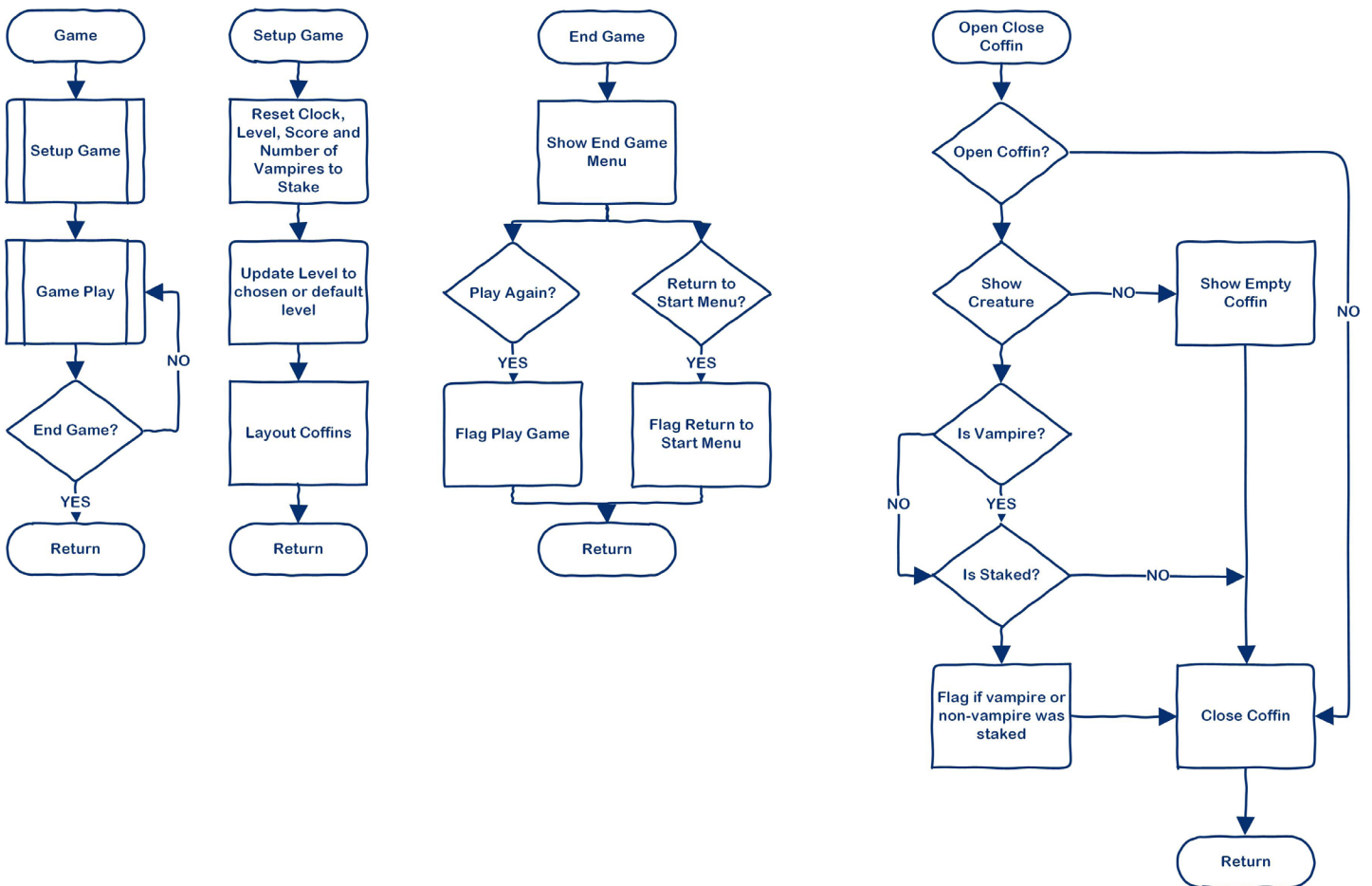
# FLOW CHARTS

## Menus



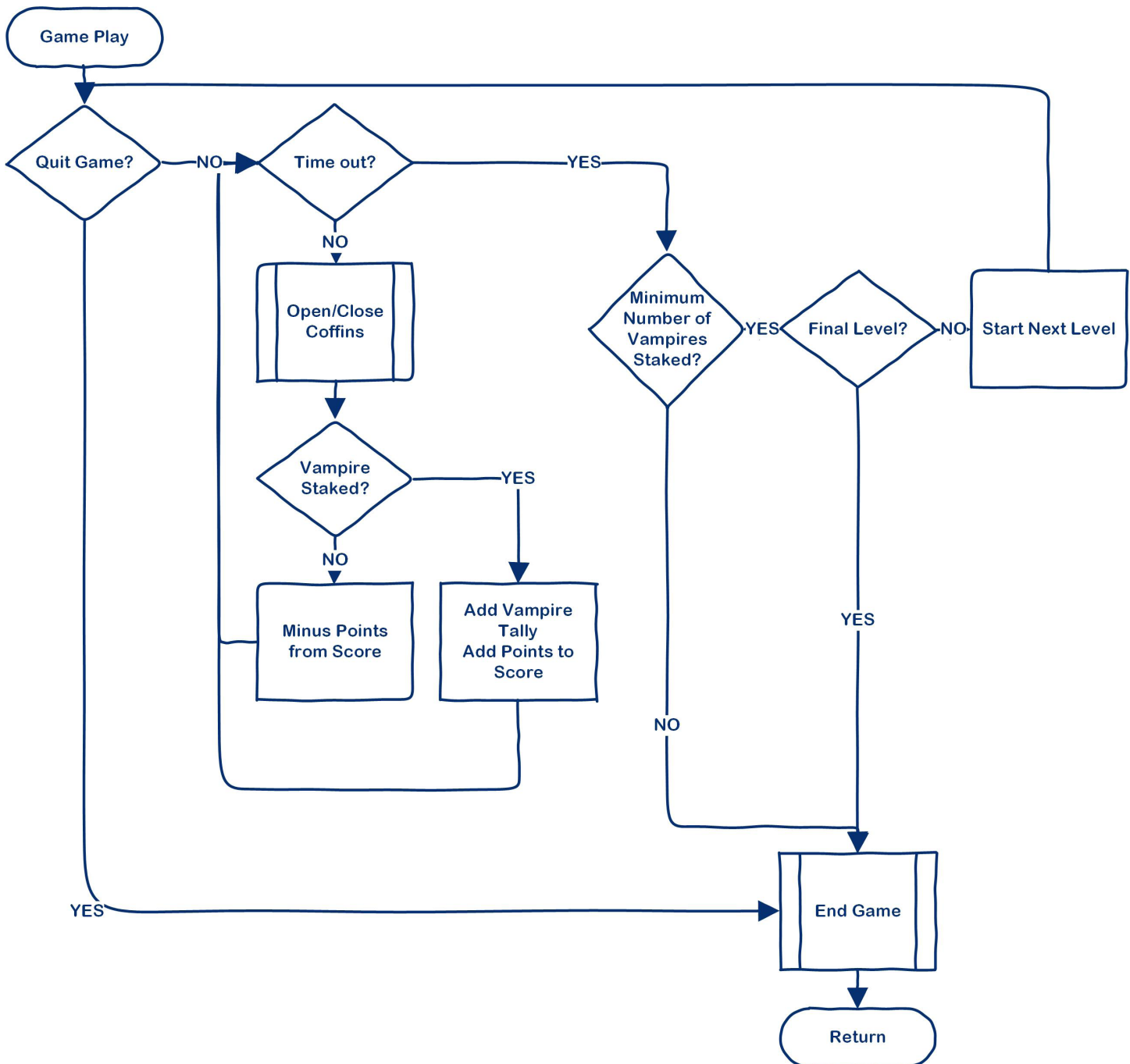
# FLOW CHARTS

## Game



# FLOW CHARTS

## Game Play



# CONCEPT SKETCHES



Dracula



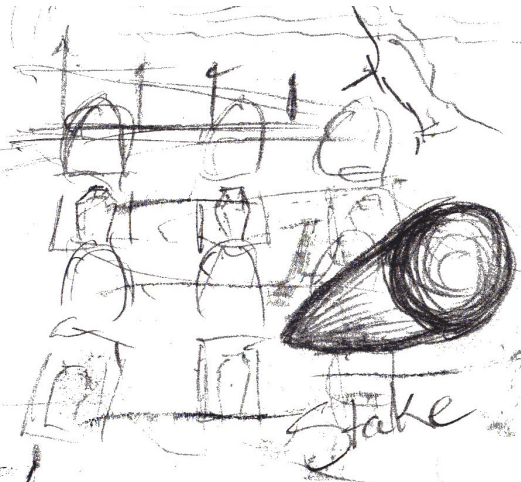
Nosferatu



Sparkly



Sad Pretty Vampire

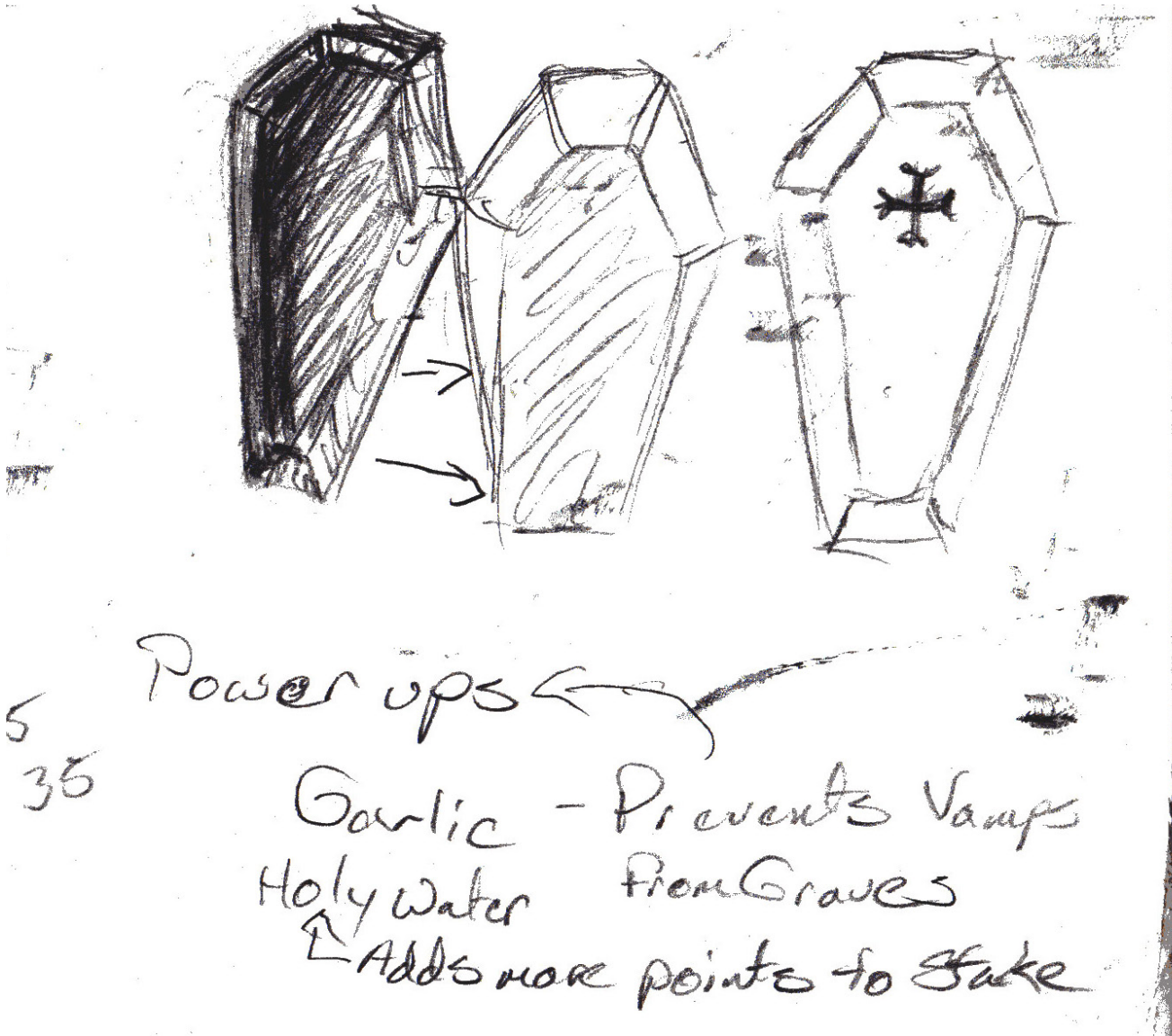


Snake

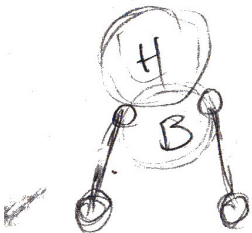
w 97.95  
h 116.65  
x 0.15  
y 52.35



# CONCEPT SKETCHES

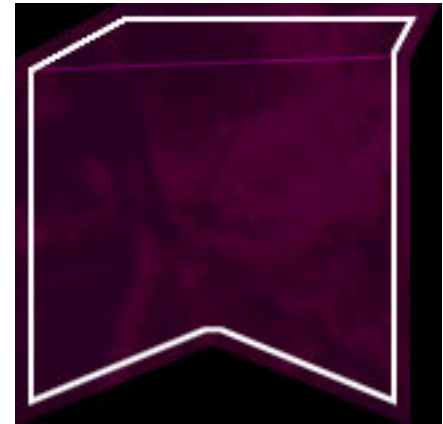


## PUPPET-LIKE CONSTRUCTION

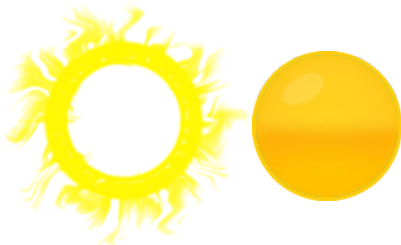


# DESIGNED ASSETS

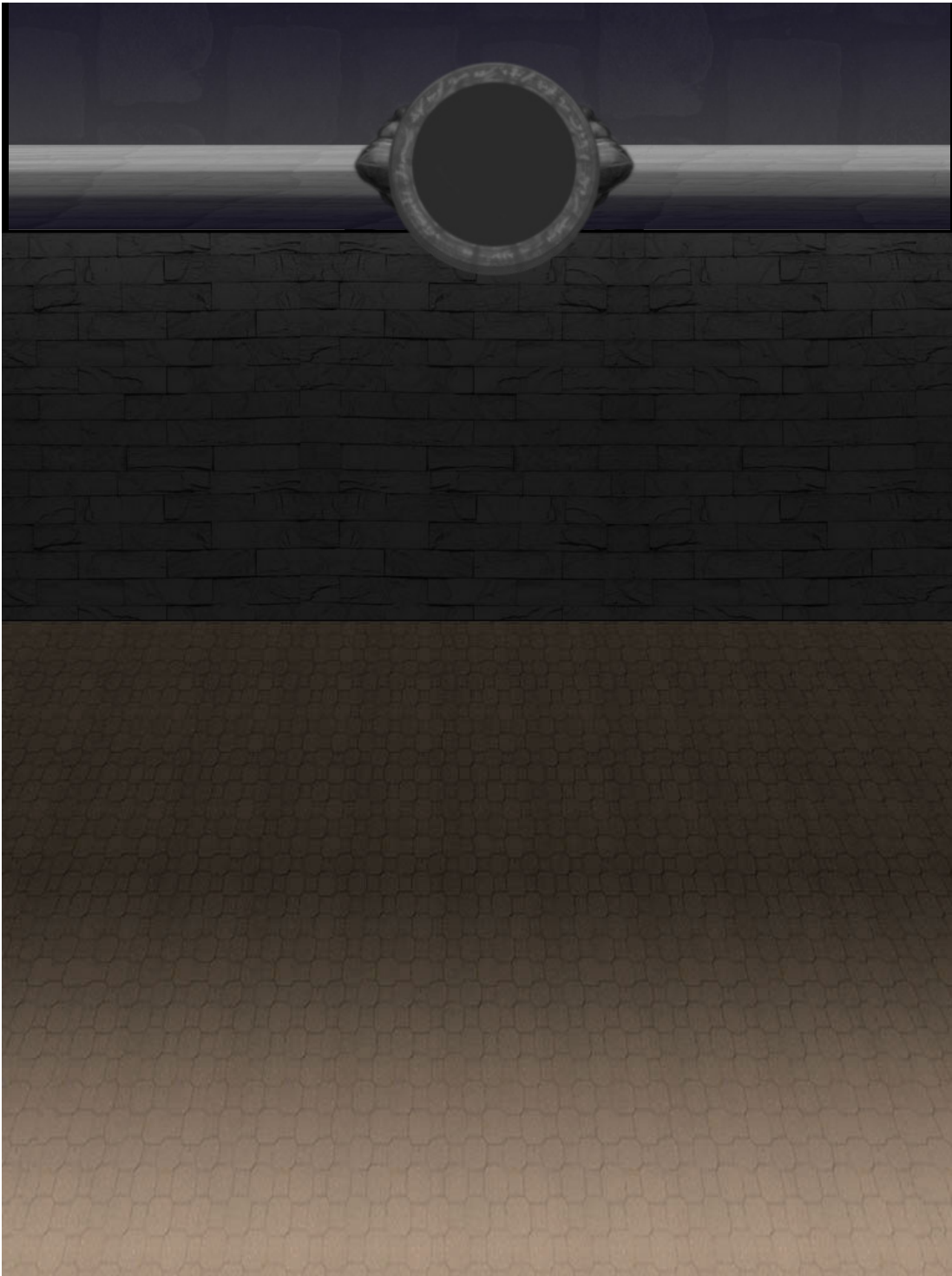
## STAKE A VAMPIRE



**←QUIT**



# DESIGNED ASSETS

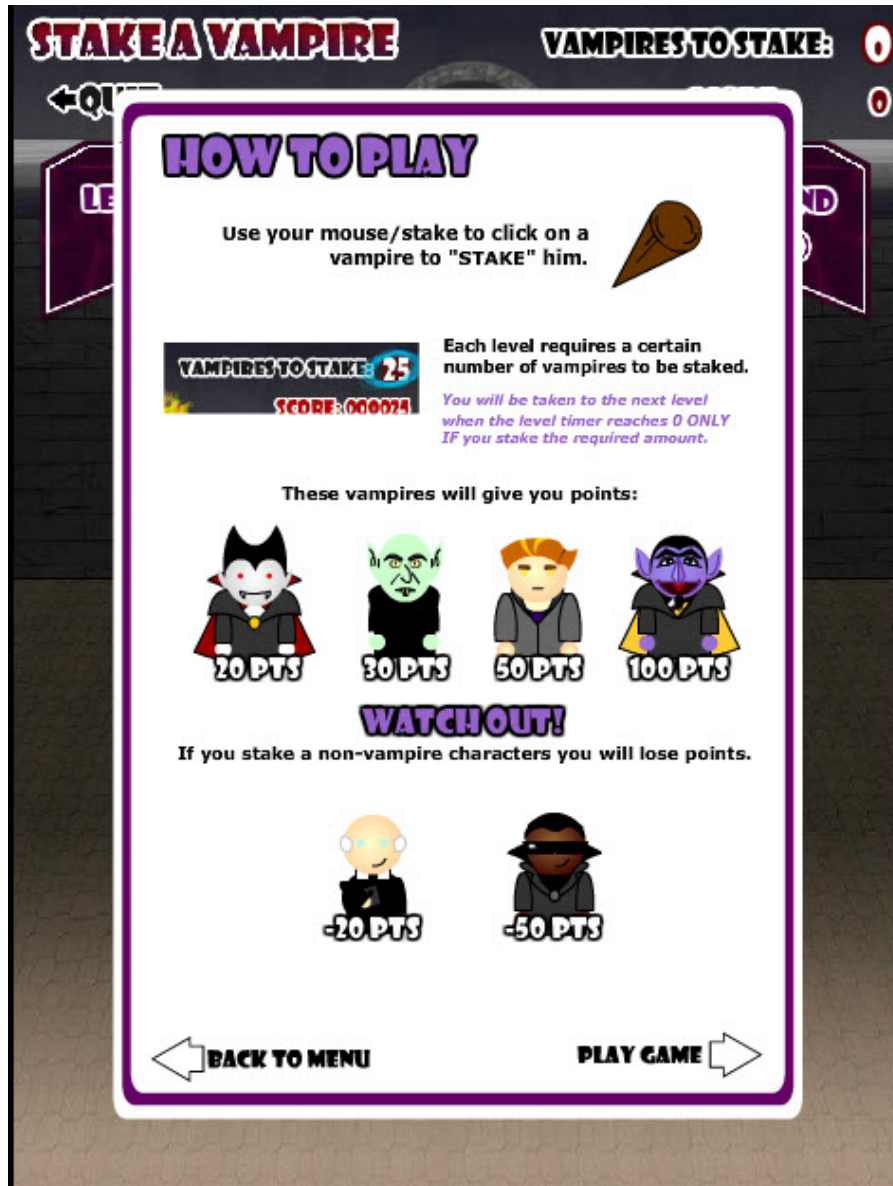


# DESIGN USER INTERFACE



**Start Screen**

# DESIGN USER INTERFACE



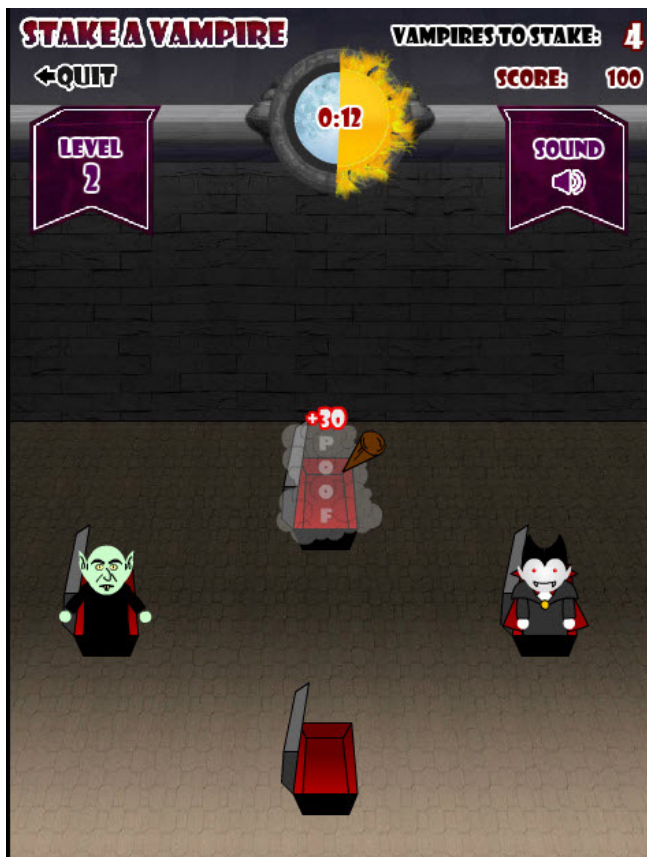
## How to Play Screen

# DESIGN USER INTERFACE

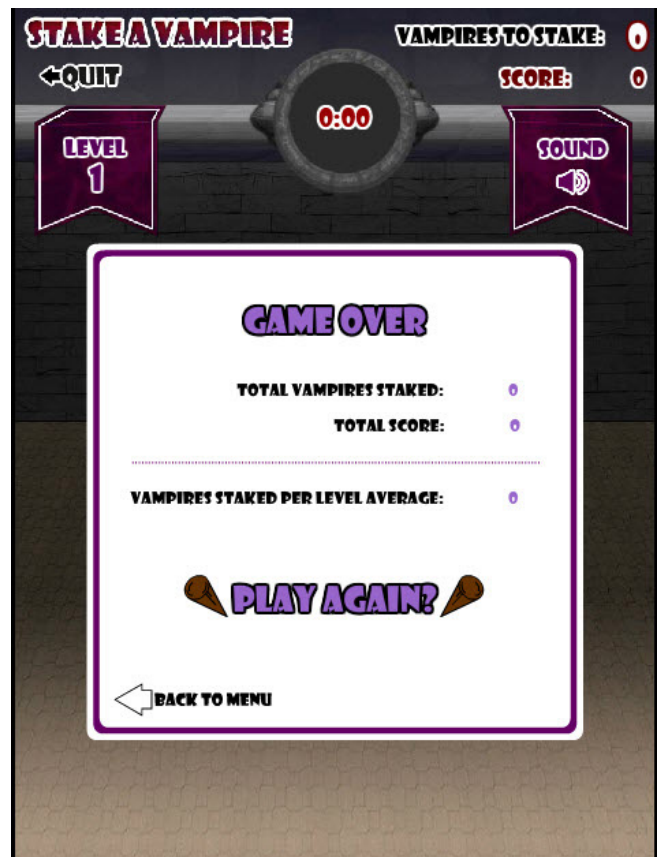


Levels Screen

# DESIGN USER INTERFACE



Play Screen



End Game Screen

# FINAL THOUGHTS

This project was indeed challenging due to the several moving parts. As one may notice is that several items were removed and others added to give the game more variety.

One thing that caused a setback was the initial layout of the game. If that was planned from the beginning then more items would have made it to the final version. The final layout works better than the initial prototype for it allows for controls to be placed in a logical setting.

The challenges that I had for this project was getting the events to dispatch to the children class. After research on the web it was found that I needed to pass a stage reference to them for it to register to the children class.

There were some snippets of code that I found useful on the web such as the pieMask (for the timer), preloader and TweenLite library for the in/out sliding of the section screens. All snippets have been cited within the classes that use them.