Qwivel

Design Document

#### Copyright info and credits and whatnot.

This template is CC0 licensed.

You can make a local copy of this document under:  
  
File >> Make a copy…  
  
Which can be modified without requesting further permissions.

## Index

1. [Index](#e1pmgau96gcf)
2. [Game Overview](#xthk3v7lq97k)
   1. [Summary](#raan6xudj5eb)
   2. [Gameplay](#fxytzxib1owi)
   3. [Mindset](#8zxra7igzw0j)
   4. [Highlights](#tfim2ti7q8ps)
3. [System Design](#p4vl5v43z96z)
   1. [Input](#zbokdsdefjtt)
   2. [Output](#w3nnp8hyo8d0)
   3. [Behaviors](#c1o8uh4doenn)
   4. [Requirements](#70sj49658smi)
      1. Unique Systems
      2. Other Systems
4. [World Design](#w0lj0vhvnapk)
   1. [Setting](#ezvi41t0pltj)
   2. [Subsettings](#bj0bu1603pmi)
5. [Level Designs](#d6feb8qxmgwk)
6. [Visual Design](#3zwibvw9917o)
   1. [Color Palette](#wwis5rfh62o1)
   2. [Stylistic Attributes / Elements](#nfbv3574fe99)
   3. [Effects](#sf7zt0yut0qz)
7. [Music/Sound Design](#m44lom9y3ath)
   1. [Sound Palette / Instrumentation](#fm089gw44vyz)
   2. [Stylistic Attributes / Elements](#sjnj25bv5m3n)
   3. [Effects](#3hz10q59qfbe)
8. [Required Assets](#6nuyp0zcsxjo)
9. [Schedule](#mkx3te5ni251)
10. [Other Notes](#r57nsvy2tsyz)

# Game Overview

## Summary

Qwivel is a two-player, turn-based quiz game where players must get the best out of 6 questions to win. Question originates from CS topics discussed in the Computing I-II courses. Topics are; data structures, object-oriented programming, abstraction and algorithm analysis.

## Gameplay

## The player answers are matched to the question’s correct answer and if correct the count of the correct answers is stored for that player. This system should then display a new question for the other player to respond to. The number of correct responses is recorded. The process is repeated changing the players’ turn each round and the winner is decided after 6 rounds, ie. each player answered 3 questions.

## For the first round, the system should randomly select a player to respond to the question posed. For the second round and other rounds, a player should be selected based on which player was picked to answer the previous question. If player 2 is selected at the beginning, then player 1 is selected next and the players keep switching turns until 6 questions have been asked. At the end of the 6th round, the player with the most correct choices wins the game.

## If the players have an equal number of correct answers the game ends in a draw and replay is required. If there is a winner replay is optional. At the end of the 6 rounds, display a summary of the wrong answers and correct solution

## Mindset

How will the player feel throughout the game? What is their initial \*expectation\*? Should they feel powerful enough to overcome all the \*obstacles\* right out of the gate? Or should they feel overwhelmed at the challenge, in search of the proper \*tactics\*?

What is the pace of the game? Is it a race against time, or does it require careful and deliberate planning/exploration?

## Highlights

* What makes this game stand out?
* Are there any new mechanics?
* New uses of old mechanics?
* Interesting aesthetics?
* Twists?
* Interesting algorithms?

# System Design

## Input

How will the player interact with the game? Keyboard, mouse, joystick?

What kind of actions will the player be able to perform?

Walking, running, shooting, jumping, attacks, combo moves, etc?

If there are too many, how can you make them easy to use?

## Output

What kind of reactions will the system generate?

What happens when a player collides with another entity?

Will platforms fall when walked on? Will enemies get knocked back on attack?

Will enemies need to respond to the actions of other enemies?

## Behaviors

* Player
  + Can directly listen for input
  + Can spawn variety of projectiles
* Enemies
  + Can spawn variety of projectiles
  + Can follow paths
  + Can be knocked back (by projectile)
  + Can be stunned (by projectile)
  + Uses state machines
* Switches/buttons
  + Can be toggled
  + Can interact with platforms
  + Can spawn projectiles and enemies
* Platforms
  + Can follow paths
  + Can be switched on and off
* Props
  + Can be interacted with (rigid body)

## Requirements

### Unique Systems

Are there any unique systems/algorithms you’re going to have to implement? Describe them, and the process you’re planning to use to implement them.

Some examples might be Portal’s portal system, Spelunky’s level generation, or Spore’s character designer

### Other Systems

* Scene graph
* Application state management
* Nav meshes
* Bezier paths
* A\* path finding
* FSMs, decision trees
* Observer pattern (event listening)

# 

# World Design

## Setting

Time and place, if applicable. Medieval dungeon, perhaps?

## Subsettings

* Cell block
* Torture chamber
* Crypt
* Hell

# Level Designs

## Cell block

### Goals

Escape

### Obstacles

Cell door, guards

### Tactics

Murder guard, steal key

### Expectation

Might not succeed, but death comes either way

### Learned Mechanics

* Explicitly taught
  + Call for attention / distract.
  + Attack
* Implicitly learned
  + Enemies follow sounds
  + Attacks can be blocked
  + Dead enemies drop loot

… Add as many copies of the previous section as you need… Ctrl+Enter for a page break

# Visual Design

## Color Palette

What kind of color scheme are you planning to use?

Dull, desaturated colors?

Lively, vibrant colors?

How will they be applied to the game objects? Will they be silhouettes? Will they be outlined? Will they be shaded? How so?

Is there an existing work or two you want to draw from? Reference material is key.

## Stylistic Attributes / Elements

Any particular visual motifs (signature shapes/effects) you're planning to incorporate into the overall style?

Cute, round shapes? Aggressive, harsh triangles? Obscured silhouettes?

What about design languages? Should enemies or hazards be of a certain color or shape?

What about to call attention to specific parts of the game?

Nostalgic film noir, tense wide angle cameras, glitchy static?

Any visual influences (cultural, setting)?

Mayan, celtic, post-apocalyptic dystopia, alien, etc.?

## Effects

What kind of visual effects are you going to need for this game?

Motion blur? Bloom? Film noir?

# Music / Sound Design

## Sound Palette / Instrumentation

What are you going for with the instrumentation?

Dark, bassy tones? Happy, high-pitched tones?

Even electronically synthesized instruments can use a well-grounded basis. What real instruments do you want to base your palette on?

Where is the emphasis, which instruments (if any) will carry the focus?

Is there an existing song or two you want to draw from? Reference material is key.

## Stylistic Attributes / Elements

Any particular musical/auditory motifs (signature phrases/effects) you're planning to incorporate?

Does the game have a specific 'theme' you want to reuse throughout the soundtrack?

Will you use musical cues to draw attention to certain parts of the game?

## Effects

What post-processing techniques are you going to need?

Cross-fading? Reverberation? Filters?

# Required Assets

These should probably be roughly in order of use!

## Visual

1. Models
   1. Characters
      1. Player
      2. Enemy
   2. Props
      1. Vase
2. Animations
   1. Player
      1. Idle
      2. Walk
      3. Attack
      4. Hit
      5. Death
   2. Enemy
      1. Idle
      2. Walk
      3. Attack
      4. Hit
      5. Death
3. Maps
   1. Cell block
   2. Crypt

### 

### 

## Audio

1. Music
   1. Cell block theme
      1. Battle variation
2. Sounds
   1. Foot Steps
      1. Soft
      2. Hard
   2. Jumping
      1. Lift-off
      2. Landing
   3. Combat
      1. Hit
      2. Block

...

# Schedule

A rough list of steps towards release.

Aim for a portion of the milestone every X days/weeks.

* Pre-alpha
  + Player
  + Obstacles
  + Basic Enemies
  + Basic Level Designs
  + Temporary graphics
  + Game should be playable and the concept should be roughly defined
* Alpha
  + Advanced obstacles
  + Advanced enemies
  + Rough draft graphics/music/sounds
  + Rough level designs and level-specific assets
  + Game should be near complete and somewhat optimized
  + Assets should be roughed out but well-defined
* Beta
  + Complete set of obstacles/enemies
  + Complete levels
  + Finished assets
  + Debug, debug, debug
  + Optimize, optimize, optimize
* Gamma/delta
  + Release candidates
  + Finalize debugging
* Omega/final!
  + Done!

# Other Notes

blah blah blah additional notes, logs, etc.