

# Game Name Here

## *Game Design Document*

---

Copyright notice / author information / boring legal stuff nobody likes

PLEASE NOTE:

You can copy this into your Drive by going to **“File” >> “Make a copy...”**

You do not need to request permission to edit this document if you make a private copy.

## *Index*

---

### Index

1. [Index](#)
2. [Game Design](#)
  - a. [Summary](#)
  - b. [Gameplay](#)
  - c. [Mindset](#)
3. [Technical](#)
  - . [Screens](#)
  - a. [Controls](#)
  - b. [Mechanics](#)
4. [Level Design](#)
  - . [Themes](#)
- i. Ambience
- ii. Objects
  1. Ambient
  2. Interactive
- iii. Challenges
  - a. [Game Flow](#)
  5. [Development](#)
  - . [Abstract Classes](#)

- a. [Derived Classes](#)
- 6. [Graphics](#)
  - . [Style Attributes](#)
- a. [Graphics Needed](#)
- 7. [Sounds/Music](#)
  - . [Style Attributes](#)
- a. [Sounds Needed](#)
- b. [Music Needed](#)
- 8. [Schedule](#)

# *Game Design*

---

## **Summary**

You just robbed a bank and are on the run with police and dogs chasing after you! The more obstacles you hit, the more money you lose. Don't get caught by the police over, or its game over.

## **Gameplay**

Endless runner in a side scroller style, the player will be jumping over and sliding under obstacles and over gaps. The goal of the game is to stay alive as long as possible and to keep as much money as you can.

## **Mindset**

The player should feel a sense of urgency as the continue to try and navigate through the obstacles to stay alive as long as possible.

# *Technical*

---

## **Screens**

- 1. Title Screen
  - a. Options
- 2. Level Select
- 3. Game
  - . Inventory
- a. Assessment / Next Level
- 4. End Credits

(example)

## Controls

It will be an auto run, so all the player will have to control at this point is sliding and jumping. This is subject to change as different challenges or ideas arise.

## Mechanics

Are there any interesting mechanics? If so, how are you going to accomplish them? Physics, algorithms, etc.

# Level Design

---

*(Note : These sections can safely be skipped if they're not relevant, or you'd rather go about it another way. For most games, at least one of them should be useful. But I'll understand if you don't want to use them. It'll only hurt my feelings a little bit.)*

## Themes

1. Forest
  - a. Mood
    - i. Dark, calm, foreboding
      - b. Objects
        - . Ambient
          1. Fireflies
          2. Beams of moonlight
          3. Tall grass
    - i. Interactive
      1. Wolves
      2. Goblins
      3. Rocks
2. Castle
  - . Mood
    - . Dangerous, tense, active
      - a. Objects
        - . Ambient
          1. Rodents
          2. Torches
          3. Suits of armor
    - i. Interactive
      1. Guards
      2. Giant rats
      3. Chests

(example)

## Game Flow

1. Player starts in forest
2. Pond to the left, must move right
3. To the right is a hill, player jumps to traverse it ("jump" taught)
4. Player encounters castle - door's shut and locked
5. There's a window within jump height, and a rock on the ground
6. Player picks up rock and throws at glass ("throw" taught)
7. ... etc.

*(example)*

## Development

---

### Abstract Classes / Components

1. BasePhysics
  - a. BasePlayer
  - b. BaseEnemy
  - c. BaseObject
2. BaseObstacle
3. BaseInteractable

*(example)*

### Derived Classes / Component Compositions

1. BasePlayer
  - a. PlayerMain
  - b. PlayerUnlockable
2. BaseEnemy
  - . EnemyWolf
  - a. EnemyGoblin
  - b. EnemyGuard (may drop key)
  - c. EnemyGiantRat
  - d. EnemyPrisoner
3. BaseObject
  - . ObjectRock (pick-up-able, throwable)
  - a. ObjectChest (pick-up-able, throwable, spits gold coins with key)
  - b. ObjectGoldCoin (cha-ching!)
  - c. ObjectKey (pick-up-able, throwable)
4. BaseObstacle
  - . ObstacleWindow (destroyed with rock)
  - a. ObstacleWall
  - b. ObstacleGate (watches to see if certain buttons are pressed)
5. BaseInteractable
  - . InteractableButton

*(example)*

# Graphics

---

## Style Attributes

What kinds of colors will you be using? Do you have a limited palette to work with? A post-processed HSV map/image? Consistency is key for immersion.

What kind of graphic style are you going for? Cartoonish? Pixel-y? Cute? How, specifically? Solid, thick outlines with flat hues? Non-black outlines with limited tints/shades? Emphasize smooth curvatures over sharp angles? Describe a set of general rules depicting your style here.

**I am going for a 2D pixel game, low detail pixel is simple and interesting to me if done correctly.**

Well-designed feedback, both good (e.g. leveling up) and bad (e.g. being hit), are great for teaching the player how to play through trial and error, instead of scripting a lengthy tutorial. What kind of visual feedback are you going to use to let the player know they're interacting with something? That they *\*can\** interact with something?

## Graphics Needed

1. Characters
  - a. Human-like
    - i. Goblin (idle, walking, throwing)
    - ii. Guard (idle, walking, stabbing)
    - iii. Prisoner (walking, running)
  - b. Other
    - i. Wolf (idle, walking, running)
    - ii. Giant Rat (idle, scurrying)
2. Blocks
  - . Dirt
  - a. Dirt/Grass
  - b. Stone Block
  - c. Stone Bricks
  - d. Tiled Floor
  - e. Weathered Stone Block
  - f. Weathered Stone Bricks
3. Ambient
  - . Tall Grass
  - a. Rodent (idle, scurrying)
  - b. Torch
  - c. Armored Suit
  - d. Chains (matching Weathered Stone Bricks)
  - e. Blood stains (matching Weathered Stone Bricks)
4. Other

- . Chest
- a. Door (matching Stone Bricks)
- b. Gate
- c. Button (matching Weathered Stone Bricks)

*(example)*

*(Note : If you're soloing you might not need to define this part, as you can just use the Derived Classes + Themes section as a reference. It's up to you.)*

## Sounds/Music

---

### Style Attributes

Again, consistency is key. Define that consistency here. What kind of instruments do you want to use in your music? Any particular tempo, key? Influences, genre? Mood?

Stylistically, what kind of sound effects are you looking for? Do you want to exaggerate actions with lengthy, cartoony sounds (e.g. mario's jump), or use just enough to let the player know something happened (e.g. mega man's landing)? Going for realism? You can use the music style as a bit of a reference too.

Remember, auditory feedback should stand out from the music and other sound effects so the player hears it well. Volume, panning, and frequency/pitch are all important aspects to consider in both music *and* sounds - so plan accordingly!

### Sounds Needed

1. Effects
  - a. Soft Footsteps (dirt floor)
  - b. Sharper Footsteps (stone floor)
  - c. Soft Landing (low vertical velocity)
  - d. Hard Landing (high vertical velocity)
  - e. Glass Breaking
  - f. Chest Opening
  - g. Door Opening
2. Feedback
  - . Relieved "Ahhhh!" (health)
  - a. Shocked "Ooomph!" (attacked)
  - b. Happy chime (extra life)
  - c. Sad chime (died)

*(example)*

### Music Needed

1. Slow-paced, nerve-racking "forest" track

2. Exciting “castle” track
3. Creepy, slow “dungeon” track
4. Happy ending credits track
5. Rick Astley’s hit #1 single “Never Gonna Give You Up”

*(example)*

*(Note : Again, if you’re soloing you might be able to / want to skip this section. It’s up to you.)*

## Schedule

---

*(what is a schedule, i don’t even. list is good enough, right? if not add some dates i guess)*

1. develop base classes
  - a. base entity
    - i. base player
    - ii. base enemy
    - iii. base block
  - b. base app state
    - .game world
    - i. menu world
2. develop player and basic block classes
  - . physics / collisions
3. find some smooth controls/physics
4. develop other derived classes
  - . blocks
    - .moving
    - i. falling
    - ii. breaking
    - iii. cloud
  - a. enemies
    - .soldier
    - i. rat
    - ii. etc.
5. design levels
  - . introduce motion/jumping
  - a. introduce throwing
  - b. mind the pacing, let the player play between lessons
6. design sounds
7. design music

*(example)*