

Define this out. Highlight areas that need chemistry input

<https://ironbellystudios.app.box.com/ironbellygdd>

Joe is in blue

Questions in red

General commentary in gray

Game Design Considerations(may or may not apply):

## 1. Overall

- Basic Concept
  - Spaceship flying around in 3d space, collecting simple molecules while avoiding macromolecules.
- Platform
  - Focus on web implementation (over phone implementation).
- **Estimated Playtime**
  - Estimated somewhere between 10 minutes and hour for average user

## 2. Story

- Background Story
  - Minimal story
  - Collect molecules as resources for base ship which is built-up over time.
- Characters
  - Space Base
  - Molecules
  - The Spaceship
- Tone / Mood
  - Space Opera
- Graphical Style (cartoony, realistic)
  - Molecules are rendered as ball-and-stick molecules.
  - Everything else will be cartoony
- Audio Style
  - TBD

## 3. Gameplay

- **Objective**
  - You have a base ship that you are collecting resources for. Resources can be things like fuel, food, base add-ons. Add on cans be things like research facilities, engines, etc.
  - Katamari like game?
  - mini missions in levels
  - story mayhap?
  - Collect something?
    - Learn game
    - collect different things for some sort of upgrade
  - Destroy something?
    - Collect tnt and shoot it at things to destroy them
  - Catch something?
  - Build something?
  - Protect something?

- Survive something?
- Difficulty/Playability/Fun
  - Difficulty Levels?
    - this is a major element of making a game fun, the game cannot be too easy or too hard
  - Artificial Intelligence?
    - maybe if the motion of the molecules
  - Score?
  - Time?
  - Game over condition?
    - how does this affect user's time on a level?
- **Game Structure(chapters, levels, sandbox)**
  - joe is thinking levels
  - 2 minutes a level would be 15 levels
  - how many minutes should be spent on a level?
- Gameplay modes(singleplayer, story, battle,multiplayer)
  - singleplayer
- **Controls**
  - arrow keys / acceleration for steering
  - how to control speed of spaceship?
- Physics / Asset Interactions / Manipulations
  - pickup/destroy?
- **Planned Perspective(1st or 3rd)**
  - joe: first perspective, but approach this with flexibility

#### 4. Chemistry Specifics

- Assets
  - Molecules as obstacles and collectables
- Objectives

#### 5. User Interface

- Main Menu Design & Content
- Gameplay Options
- Game settings(resolution)
- In-game HUD Design & Content
- In-game Menu Design & Content
- Game over screen

#### 6. Learning Curve

- Focus on making the game extremely easy to pick up and set down
- Game Introduction
- Game Tutorials
  - Early levels will act as tutorials
- Level progression?

#### 7. Player Rewards - probably will want some sort of this

- Store?
- Score / Currency?
- Player Visual Customization
- Player Level Up

## **8. Assets**

- Characters
- Weapons
- Equipment and Upgrades
- World / Environment
- Audio

## **9. Multiplayer**

- [this is not an expected feature](#)

# Game Name Here

---

Game Design Document

Copyright notice / author information / boring legal stuff nobody likes

---

# *Index*

1. [Index](#)
2. [Game Design](#)
  - a. [Summary](#)
  - b. [Gameplay](#)
  - c. [Mindset](#)
3. [Technical](#)
  - a. [Screens](#)
  - b. [Controls](#)
  - c. [Mechanics](#)
4. [Level Design](#)
  - a. [Themes](#)
    - i. Ambience
    - ii. Objects
      1. Ambient
      2. Interactive
    - iii. Challenges
  - b. [Game Flow](#)
5. [Development](#)
  - a. [Abstract Classes](#)
  - b. [Derived Classes](#)
6. [Graphics](#)
  - a. [Style Attributes](#)
  - b. [Graphics Needed](#)
7. [Sounds/Music](#)
  - a. [Style Attributes](#)
  - b. [Sounds Needed](#)
  - c. [Music Needed](#)
8. [Schedule](#)

### **Summary**

X is in prison, he wants to get away from police but everytime he tries to do that, they catch him. He discovers that only way to get away from police is to leave the country.

### **Gameplay**

The goal of the game is freedom. Obstacles are cops, humans, vehicles, buildings, pits and animals.

### **Mindset**

Player is aggressive , powerful, nervous and hurried. We provoke all these emotions with the help of obstacles and cops.

## Screens

1. Title Screen
  - a. Catch me if you can, Start Game, How to Play,
2. Map Select
3. Game
  - a. Inventory
  - b. Assessment / Next Level
4. End Credits

*(example)*

## Controls

Controls will be direction buttons(forward, backward), action buttons(Jump and Fire).

## 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**
    - i. **Ambient**
      1. **Fireflies**
      2. **Beams of moonlight**
      3. **Tall grass**
    - ii. **Interactive**
      1. **Wolves**
      2. **Goblins**
      3. **Rocks**
2. **Castle**
  - a. **Mood**
    - i. **Dangerous, tense, active**
  - b. **Objects**
    - i. **Ambient**
      1. **Rodents**
      2. **Torches**
      3. **Suits of armor**
    - ii. **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)*

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

*(example)*

### 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? Cartoony? 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.

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
  - a. Dirt
  - b. Dirt/Grass
  - c. Stone Block
  - d. Stone Bricks
  - e. Tiled Floor
  - f. Weathered Stone Block
  - g. Weathered Stone Bricks
3. Ambient
  - a. Tall Grass
  - b. Rodent (idle, scurrying)
  - c. Torch
  - d. Armored Suit
  - e. Chains (matching Weathered Stone Bricks)
  - f. Blood stains (matching Weathered Stone Bricks)
4. Other
  - a. Chest
  - b. Door (matching Stone Bricks)
  - c. Gate
  - d. 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.)*

## 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**
  - a. Relieved "Ahhhh!" (health)
  - b. Shocked "Ooomph!" (attacked)
  - c. Happy chime (extra life)
  - d. 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
    - i. game world
    - ii. menu world
2. develop player and basic block classes
  - a. physics / collisions
3. find some smooth controls/physics
4. develop other derived classes
  - a. blocks
    - i. moving
    - ii. falling
    - iii. breaking
    - iv. cloud
  - b. enemies
    - i. soldier
    - ii. rat
    - iii. etc.
5. design levels
  - a. introduce motion/jumping
  - b. introduce throwing
  - c. mind the pacing, let the player play between lessons
6. design sounds
7. design music

*(example)*