

Ship Invasion

Game Design Document

Authors: Emily Rosenfeld, A01198339 / Marcos Dayan, A01782876/ Remy Patgher, A01784177

Team: REM Cycle

Copyright notice / author information / boring legal stuff nobody likes

Index

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

Game Design

Summary

Ship Invasion es un videojuego de cartas TCG, en el que se juega en modo 1vs1 PvE, y el objetivo principal es preservar la mayor parte de una flota naval dada, que será sometida a constantes ataques por turnos de diferentes proyectiles y misiles representados por cartas, que tanto el jugador como el contrincante, estarán usando constantemente para destruir la flota del competidor.

Hay dos principales tipos de cartas, las cartas de ataque, las cuales sirven para escoger cierta coordenada y lanzar un devastador ataque sobre ella, y las cartas de defensa, las cuales consisten en ya sea poner flotas, o proteger contra un ataque alguna flota. Dentro de éstas, existen diferentes calidades de cartas, Bronze, Silver y Gold, en donde una carta de ataque de mejor calidad, va a atacar un radio o una cantidad mayor de casillas, y una carta de defensa de mejor calidad, va a poner una embarcación de mayor tamaño, o proporcionar una protección mayor a una embarcación.

Gameplay

Una partida consta de 2 etapas. La etapa del despliegue, y la etapa de destrucción.

La partida inicia con la etapa de despliegue. Una pantalla del territorio de la flota, este está representado con un grid, en el que se acomodarán 5 cartas de flota que serán repartidas aleatoriamente al jugador. Habrá un lapso de 30 segundos para que se acomoden esas 5 cartas iniciales, y pasado ese lapso, se tirará un volado digital para decidir si empieza a jugar el jugador o el Environment.

En la etapa de la destrucción, se reparten 6 cartas al jugador, tres de ellas de ataque, y tres de ellas de defensa. Durante ésta etapa, se dará un lapso de 5 minutos en donde se repartirán en turnos de 10 segundos cada uno para que el jugador escoja una carta para jugar. Acá es donde entra la estrategia, se debe escoger entre realizar una ataque con las cartas disponibles, o defender una posición y poniendo un escudo a un barco, o desplegando una embarcación.

¿Quién gana? Gana el jugador que destruya la embarcación del otro antes de que se acabe el tiempo, o en caso de que se acabe el tiempo, el jugador que tenga más unidades de embarcación desplegadas. En caso de que tengan la misma cantidad de unidades de embarcación, gana el jugador quien haya hecho más daño al otro, y en el final de los casos, se tiene un empate.

Mindset

Este juego es una estrategia mental, el jugador tiene que estar constantemente concentrado para ganar. Por lo tanto el jugador se sentirá poderoso al ganar, y ansioso de jugar mejor cuando pierda.

Al ver como el juego no tiene un patron exacto, el jugador no se aburrirá pues no será repetitivo.

De igual forma, el jugador tendrá la posibilidad de iniciar el juego con una apuesta. Esto con la finalidad de tratar de juntar la mayor cantidad de monedas, pues éstas sirven para personalizar tus embarcaciones, así como el tablero de juego.

Por otro lado, las mismas monedas podrán ser utilizadas para comprar privilegios dentro del juego, es decir, el jugador puede pagar cierta cantidad de monedas para tirar doble turno o para duplicar el ataque de una de sus cartas, sin embargo, estas últimas tienen un costo algo, por ello, los jugadores querrán juntar más monedas.

Technical

Screens

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

(example)

Controls

How will the player interact with the game? Will they be able to choose the controls? What kind of in-game events are they going to be able to trigger, and how? (e.g. pressing buttons, opening doors, etc.)

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
 1. Mood
 1. Dark, calm, foreboding
 2. Objects
 1. *Ambient*
 1. Fireflies
 2. Beams of moonlight
 3. Tall grass
 2. *Interactive*
 1. Wolves
 2. Goblins
 3. Rocks
2. Castle
 1. Mood
 1. Dangerous, tense, active
 2. Objects

1. *Ambient*
 1. Rodents
 2. Torches
 3. Suits of armor
2. *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
 1. BasePlayer
 2. BaseEnemy
 3. BaseObject
2. BaseObstacle
3. BaseInteractable

(example)

Derived Classes / Component Compositions

1. BasePlayer
 1. PlayerMain
 2. PlayerUnlockable
2. BaseEnemy
 1. EnemyWolf
 2. EnemyGoblin
 3. EnemyGuard (may drop key)
 4. EnemyGiantRat
 5. EnemyPrisoner
3. BaseObject

1. ObjectRock (pick-up-able, throwable)
2. ObjectChest (pick-up-able, throwable, spits gold coins with key)
3. ObjectGoldCoin (cha-ching!)
4. ObjectKey (pick-up-able, throwable)
4. BaseObstacle
 1. ObstacleWindow (destroyed with rock)
 2. ObstacleWall
 3. ObstacleGate (watches to see if certain buttons are pressed)
5. BaseInteractable
 1. 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? 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
 1. Human-like
 1. Goblin (idle, walking, throwing)
 2. Guard (idle, walking, stabbing)
 3. Prisoner (walking, running)
 2. Other
 1. Wolf (idle, walking, running)
 2. Giant Rat (idle, scurrying)
2. Blocks
 1. Dirt
 2. Dirt/Grass
 3. Stone Block
 4. Stone Bricks
 5. Tiled Floor
 6. Weathered Stone Block
 7. Weathered Stone Bricks
3. Ambient

1. Tall Grass
 2. Rodent (idle, scurrying)
 3. Torch
 4. Armored Suit
 5. Chains (matching Weathered Stone Bricks)
 6. Blood stains (matching Weathered Stone Bricks)
4. Other
 1. Chest
 2. Door (matching Stone Bricks)
 3. Gate
 4. Button (matching Weathered Stone Bricks)

(example)

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
 1. Soft Footsteps (dirt floor)
 2. Sharper Footsteps (stone floor)
 3. Soft Landing (low vertical velocity)
 4. Hard Landing (high vertical velocity)
 5. Glass Breaking
 6. Chest Opening
 7. Door Opening
2. Feedback
 1. Relieved "Ahhhh!" (health)
 2. Shocked "Ooomph!" (attacked)
 3. Happy chime (extra life)
 4. 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)

Schedule

(define the main activities and the expected dates when they should be finished. This is only a reference, and can change as the project is developed)

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

(example)