Talos Origins

Game Design Document

COMP376

Eric Provencher 26543731
Hu Ningge
Eric Philiponna 26225497

Overview

Executive summary

Talos Origins is a two-dimensional procedurally generated action-exploration RPG in a sci-fi setting that focuses on increasing your character's abilities to unravel the mystery behind his journey.

Project Description:

With each level being an increasingly large and challenging procedurally generated maze, players will never see the same stage twice. Traversing each maze involves using the breadcrumb mechanic which lets players know where they've been, along with the grapple which allows them to effortlessly navigate vertically.

Each stage is filled with a random assortment of enemies that, when killed, release collectible resources that can be spent on improving and expanding the protagonist's repertoire of abilities, to better serve him on his quest to find his creator, and aid him in conquering the God-like bosses that stand in his path.

Theme / Setting / Genre

The story a work of Science Fiction that takes place in the middle of a rebellion attempting to destroy a long-standing galactic empire.

The game itself is a two-dimensional roguelike role-playing game that involves platforming, action and strategically managing resources in an effort to progress as far as possible.

Core Gameplay Mechanics Brief

- Procedurally generated mazes
- Grapple assisted platforming for maze navigation
- Upgradable weapons
- Breadcrumb system to facilitate maze navigation

Targeted platforms

- PC
- Android/iOS (End of development)

Influences (Brief)

- **<Game> Super Metroid**

  Super Metroid was the first 16-bit entry of the series, featuring vast worlds to explore, with complex power-ups to be gained along the adventure. The game's story is told in small pieces, but the purpose of the adventure is largely shrouded in mystery. Talos Origins will have a very similar feature a very similar upgrade system, with Super Metroid serving as a strong inspiration for atmosphere, platforming and overall feel.

- **<Book> Foundation**

  The Foundation series built an incredibly detailed tale of a galactic empire with morally gray opposing characters. This series will serve as an inspiration for our game's plot forks, that will allow the player to make decisions that will shape the fate of the galaxy.

- **<Game> The Binding of Isaac**

  This game features a procedurally generated collection of rooms, filled with randomly placed enemies and boss battles, along with a complex item system. It is a needlessly difficult game, but it works because the player improves and can make their way further on each play through. Due to the RPG nature of Talos Origins, the player won't be as severely punished on death, but the difficulty will certainly ramp up in a similar fashion.

- **<Game/Comic> Spiderman**

  Spiderman's motion, momentum and swinging are incredibly interesting as gameplay mechanics and went a long way towards inspiring the movement and feel of Talos' motion and grapple mechanics.

What sets this project apart?

- <Reason #1>
- <Reason #2>
- <Reason #3>
- <Reason #4>
- <etc.>

Core Gameplay Mechanics (Detailed)

- 2D Projectile Combat

- <Details>

  The game is based around combat between Talos and multiple types of enemies. Talos can fire bullets at the enemies in order to afflict damage upon them. Defeating an enemy Talos is rewarded with currency and also contributes to the player's game progression.

- <How it works>

  Talos is equipped with a blaster gun which is his only mode of defence against enemies and ammunition is unlimited. Once Talos defeats an enemy, it will ooze currency which when collected by the player can be used towards upgrade (see "Upgrades System" below).

- Navigation of randomly generated cave maze

- <Details>

  Each map is randomly generated allowing for a new experience every time the player plays the game. A new map allows for a longer lifecycle for the game but also adds a difficulty component since the player will not be able to "learn" the maps layout.

- <How it works>

  Each level will feature a never before seen randomly generated map. Moreover as the player advances in the game, the maps will grow in order to accommodate for the extra enemies.

- Grapple Hook

- <Details>

  Talos is equipped with a grapple hook in order to allow him to navigate through each cave with ease no matter the size. It can serve as an “elevator” to climb up the maze, or as a swing explore quickly or even to avoid enemies.
  Together with the large open spaces found in the cave, the grapple hook allows for some exciting battle scenes.

- <How it works>

  Talos can hook his grapple onto any wall or meteorite given that it is within a certain distance. The player can also reel himself up or down, swing side to side, release, or jump release from the grapple at any time.
  We are currently in discussion on whether we want to control this mechanic by implementing a recharge bar which would temporarily disable the grapple after too many usages within a given time frame.

- Upgrades System

- <Details>

  At any time during the game Talos can hop on board his ship and shop for upgrades. Upgrades include larger bullets, explosive bullets, rage mode (damage multiplier), a force field shield, health packs, increased max health, increased jumping, breadcrumbs and a longer grapple.

- <How it works>

  Once bought, upgrades can be deployed at any moment in the game and have a defined lifespan. Each upgrade’s price will be scaled according to how effective it is.

Story (Brief)

After years of searching, Talos has tracked down his creator who is building an army to overthrow the peaceful Galactic Empire he helped build with his robotic overseers.

Story (Detailed)

Talos is robot whose only purpose has been to serve the will of his benevolent masters. Amid the increasing unrest spawned from calculated attacks throughout the great Galactic Empire, the Overseers have sent Talos out into the Galaxy, on a search to find the source of these attacks, and put an end to the uprising.

After years of searching, the game begins with our protagonist's arrival in the Aeos System, where he must traverse the Hieran Asteroid field and find and stop the growing rebellious army. Talos doesn't know that their leader is in fact his creator, and the very same person who created the Overseers that control Galactic Empire.

Throughout his quest, our protagonist will face many God-like entities who will reveal their motivations for standing against the will of the Overseers, and each such encounter will lead Talos to question his motivations, such that by the time he finally discovers his creator, he will have to make a decision that will shape the future of the Galaxy.

Player Composites

Player Composites

- Marketing model of the players
- Helps create intuition for design decisions and to recruit playtesters (i.e., a job description).
- Most games have primary and secondary markets (to whom to market the game). You will need the demographics for both.
- Create a profile like:

"John Brooke, 27, accountant. Single. Graduate of Loyola College. Plays games alone about once a week, and with male friends on a next-generation console plugged into an HDTV in his living room on weekend afternoons. Focuses on competitive, action games like Gears of War and Madden. Watches football one day a week. Favorite TV shows are Lost and Sopranos reruns. Drives an Audi A4. Drinks imported beer."

17

Player Composites

Answers the following questions about this target player.

- When and where does this person play games?
- Who buys the games this person uses?
- What platforms does this player use?
- How much time does this person spend in each session, and how frequent are gaming sessions?
- Who does he or she play with?
- What does the player like about games?
- What (non-game) brand images appeal to this player?
- How much disposable income does this player have?
- What licensed content would appeal to this player?
- What competes with gaming time for this player?

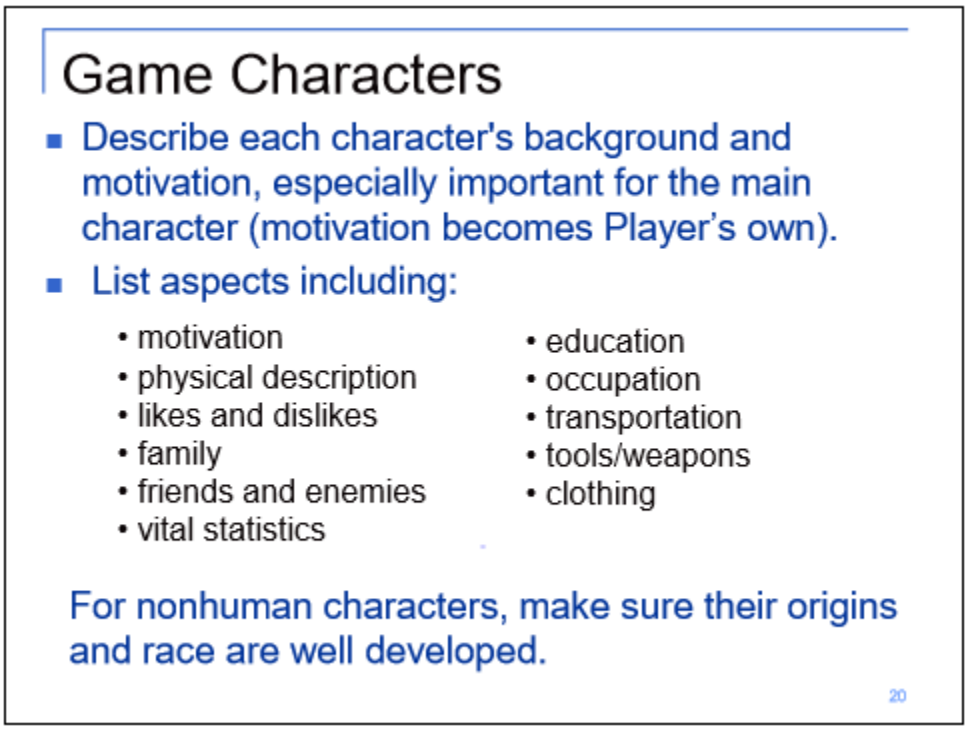
18

World

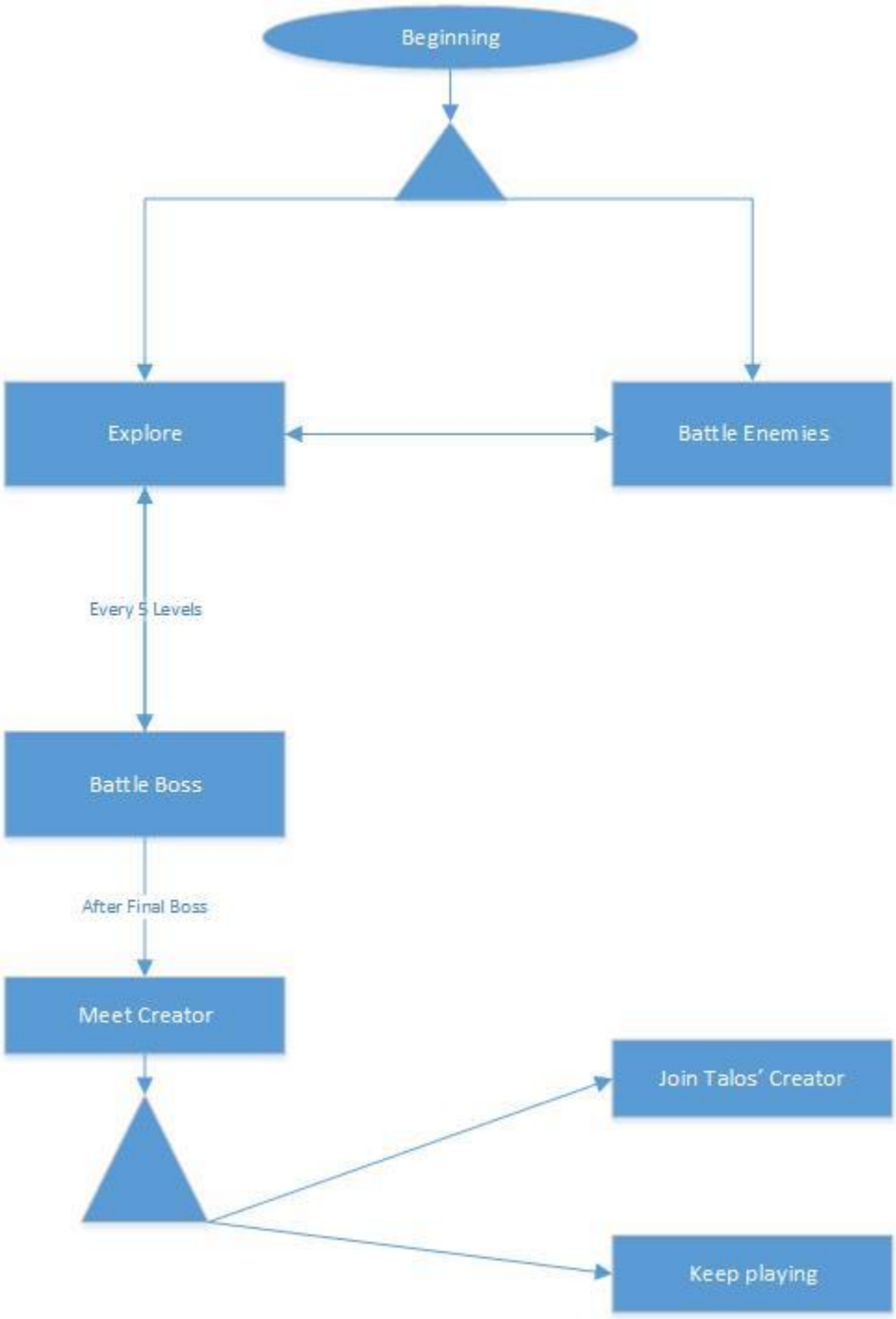
The game takes place in the Aeos System, during unrest caused by calculated attacks throughout the great Galactic Empire. Talos has been sent here by the overseers to bring peace to the galaxy by finding the source of these attacks and put an end to the uprising.

The game take place more specifically in the Hieran Asteroid field the home of many hostile creatures who are destined to protect their creator (the leader of the rebellion attacks). In order to fulfill his mission, Talos must find his way through a maze of portals which will eventually lead him to his master.

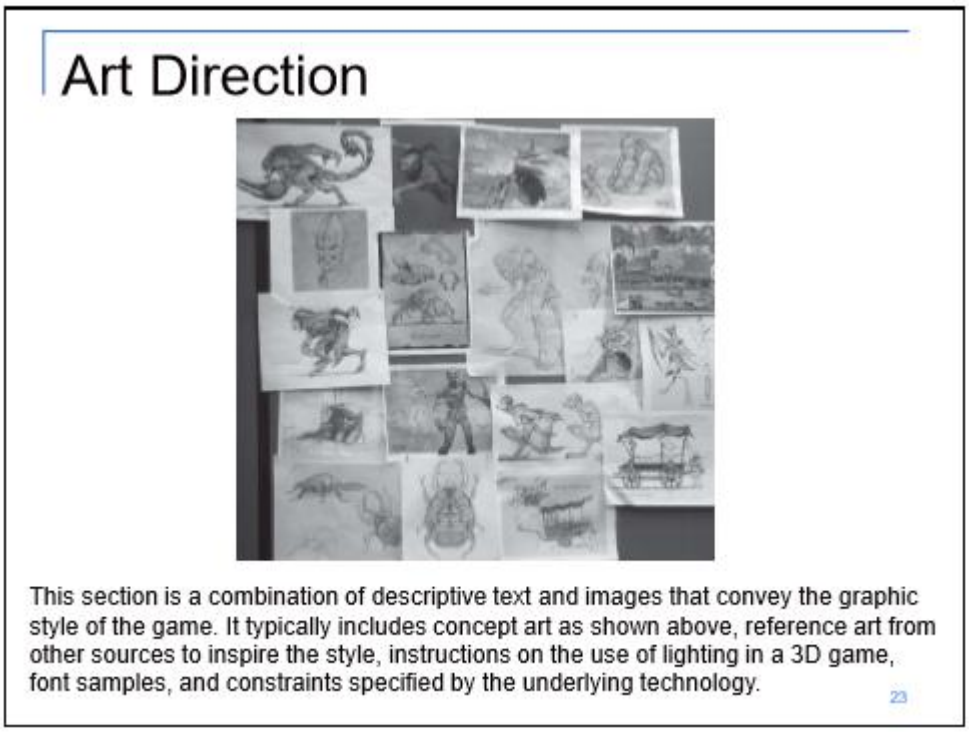
Characters



Progression Graph



Art Direction



- 2D

- Textures
    - Environment Textures
- Heightmap data (If applicable)
    - List required data required - Example: DEM data of the entire UK.
- Etc.

- Sounds

Talos:

- Pressing buttons on keypad
- Charging up

Grapple tether:

- Firing grapple
- Grapple hiting squishy enemy
- Grapple thumping on misc surface
- Grapple hitting wall

Portal:

- Entering a portal
- Exiting a portal
- Portal opening and closing

Laser blast:

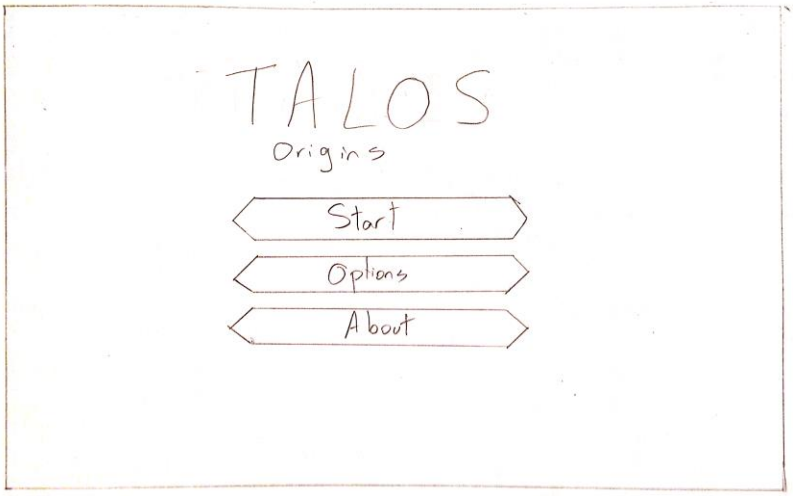
- Laser blast firing out
- Laser blast hitting target squishy exploding enemy
- Laser hitting wall
- Laser blast swishing on misc surface

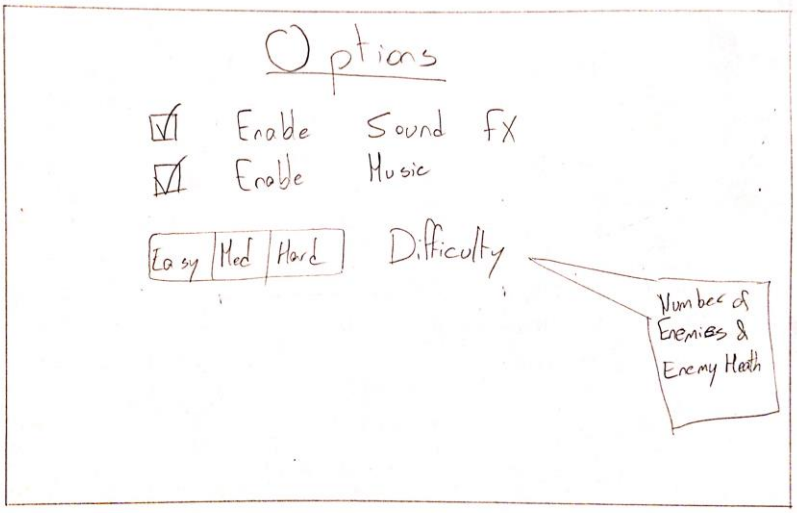
Enemy:

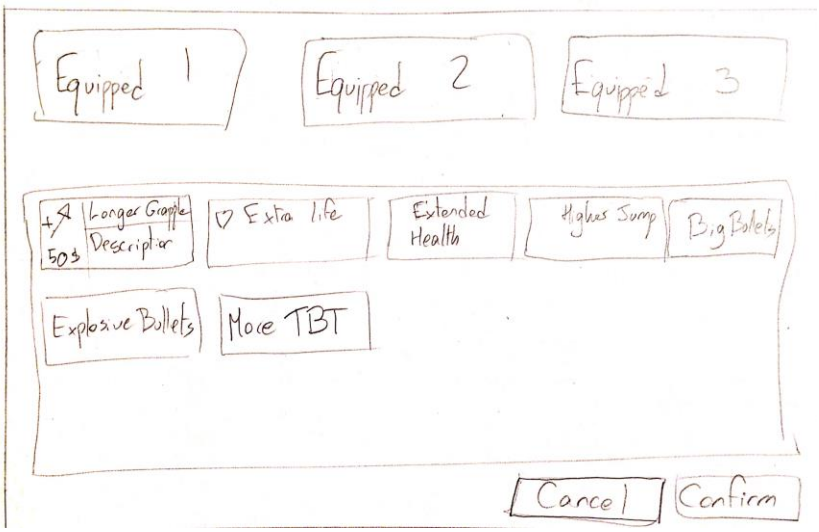
- Crawling bug creature sound
- Flying beast (not sure yet what this even is)
- Stupid walking enemy

- **Animation**
    - Environment Animations
        - Example
        - etc.
    - Character Animations
        - Player
            - Example
            - etc.
        - NPC
            - Example
            - etc.

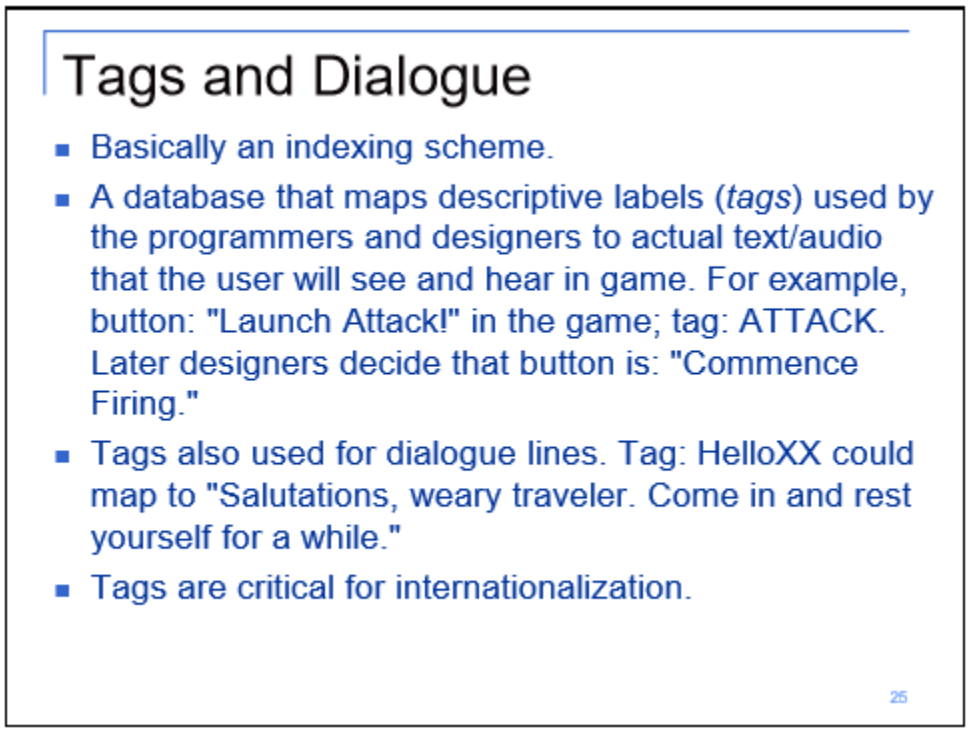
UI Storyboards







Tags and dialogue



Technology Plan

Technology Plan

- Enumerates the technology needed for producing the game, whether it will be bought or built in-house.
  - that built in-house: include rationale for why and specifications.
- A video game requires several major pieces of software.
- Some are part of the game itself: core engine (rendering, physics, and networking) and the gameplay code (user interface, artificial intelligence, and game logic).
  - Most difficult are core engine decisions: open source or mod of an engine.

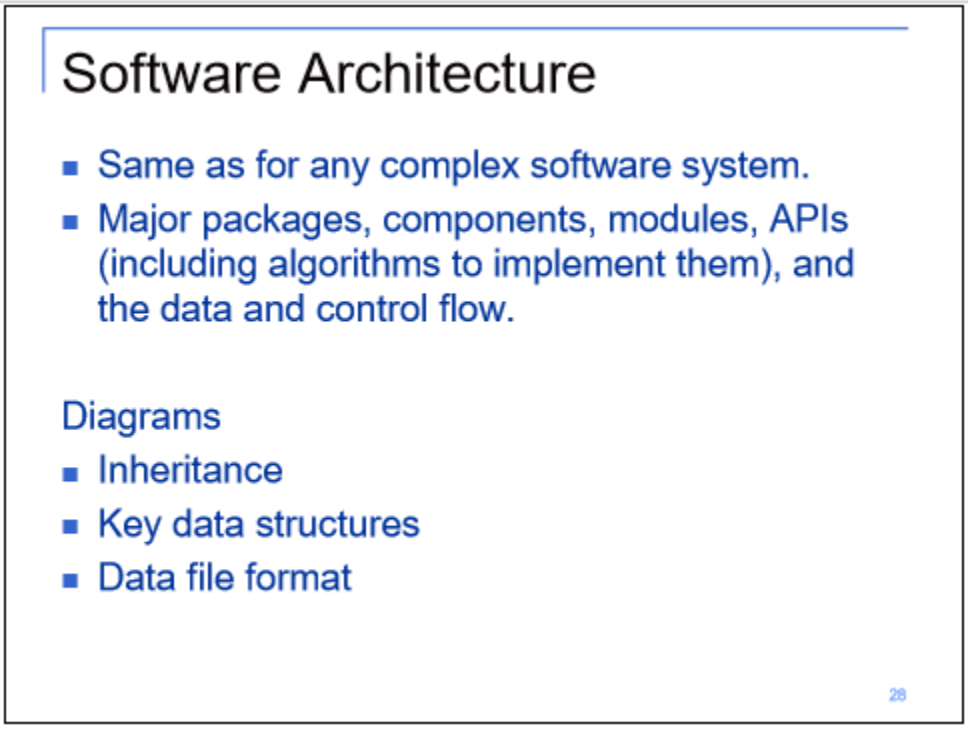
26

Technology Plan, cont.

- Other tools: maintaining the design document (e.g., Microsoft Word, wiki), the artist tools (e.g., 3DS Max, Photoshop), programmer tools (e.g., XNA, Visual Studio, VTune), management tools (e.g., Project, Excel, Trac), asset management (e.g., AlienBrain, Subversion), level editors, art exporters, and so on.
- Hardware: development consoles, workstations, servers, and often special-purpose technology like motion capture and 3D scanners
- Reusing existing technology allows you to focus more on design and innovation and less on reinventing the wheel!

27

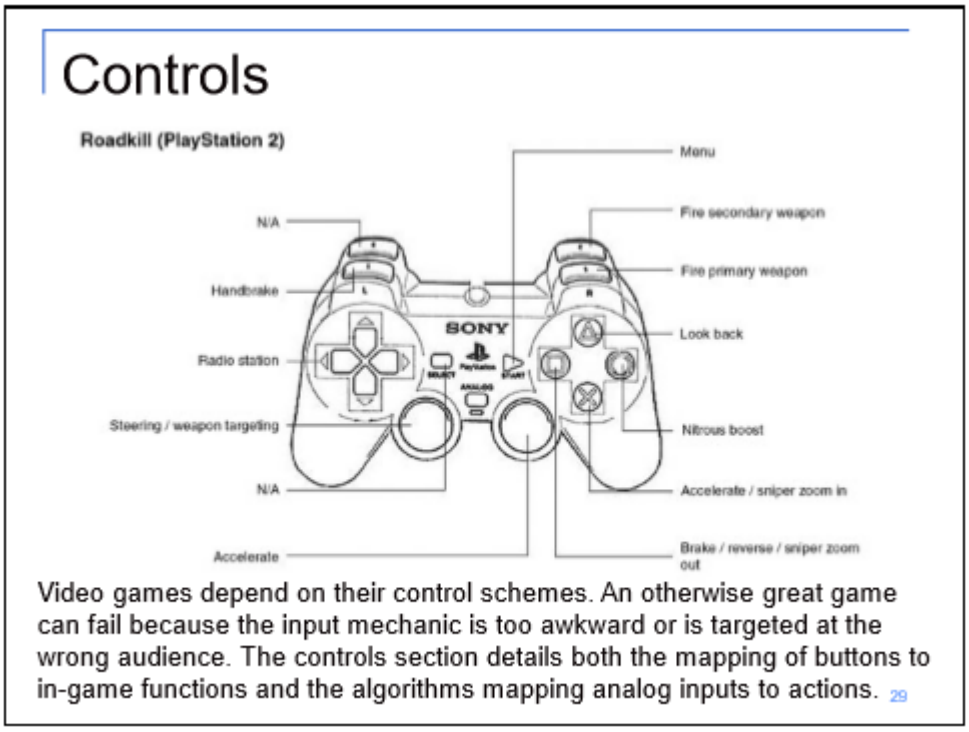
Software Architecture



- Code

- Character Scripts
  - <Grapple Handling>
  - <Weapon Handling>
  - <Character Controller>
- Map Generation Scripts
  - <Maze generation>
    - Random path and explortation tunnels
    - Treasure, Asteroid and Enemy Generation
    - Start and Exit Handling
  - <Mesh generation>
    Marching Squares algorithm for generation of cavern from bit array
- NPC Scripts
  - Enemy AI Controller (Handles all organization for enemy types)

Controls


Controls
Roadkill (PlayStation 2)
Menu
Fire secondary weapon
N/A
Fire primary weapon
Handbrake
Look back
Radio station
Nitrous boost
Steering / weapon targeting
Accelerate / sniper zoom in
N/A
Brake / reverse / sniper zoom out
Accelerate
Video games depend on their control schemes. An otherwise great game can fail because the input mechanic is too awkward or is targeted at the wrong audience. The controls section details both the mapping of buttons to in-game functions and the algorithms mapping analog inputs to actions. 29

Level Design

Level design

- Game *levels* are a simple mechanism to "segment" the game world (a scenario or map) and gameplay (a discrete change in difficulty or style of play).
- Progression to different levels shown in plot graphs.
- For games with location-based levels,
  - create maps that show the layout and the connectivity of the level.
  - Indicate major encounters, key item locations, and goals.

30

Level design, cont.

- Provide analysis of the level space, diagramming the flow of characters through the area and identifying *choke points* and *focus nodes*.
- A *choke point* is a small area that controls transition between levels. *Counter-Strike* has numerous choke points, for example, the double doors in the *Dust2* map

Counter-Strike

- A *focus node* is a location of a shared resource, increasing player interaction (e.g., a gold mine)

31

Mechanic Analysis

Mechanics Analysis

- This section is the heart of the gameplay design. Economics textbooks give good examples of how to analyze, model, and balance mechanics.
- *Game Mechanics* are small groups of rules that outline a strategic conflict, inspire a particular emotion in the player, or move the game forward. They also define the genre of the game.
- Two Examples:
  - Racing is a classic mechanic. - puts time pressure on the player, creating stress and motivation; need not be as explicit as cars on a track.
  - Shooting is a popular mechanic. It involves reflex actions, strategy in setting up a shot before the enemy appears, and leading the target when projectiles are slow.

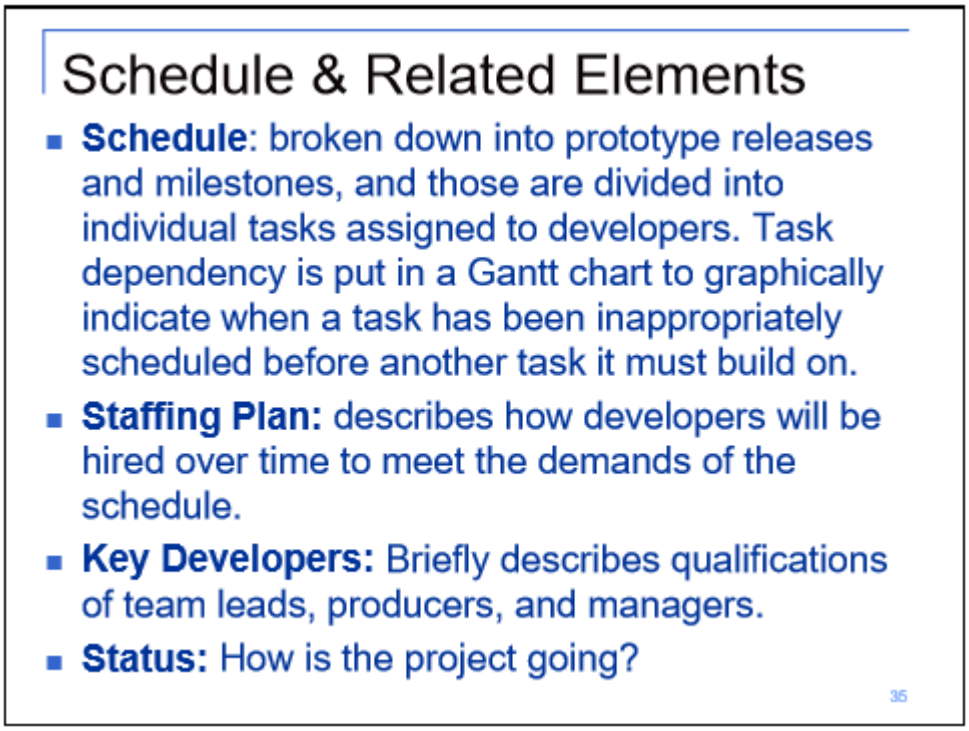
32

Mechanics Analysis

- Indicate what other games use similar mechanics and describe the differences. Motivate the gameplay reason for each mechanic - prevent an undesirable strategy, simulate some aspect of the game's setting,… Explain alternative mechanics that could have been employed.
- Give guidelines for making it balanced and useful for gameplay. Support with text, graphs and diagrams, and probability and other mathematical tools (e.g., outcome matrices).
- For each measurable property of the game (e.g., points, gold, health), give a target graph of how it should ideally vary over time.

33

Schedule



- <Object #1>
- Time Scale
  - Milestone 1
  - Milestone 2
  - Etc.

- <Object #2>
- Time Scale
  - Milestone 1
  - Milestone 2
  - Etc.

- <Object #3>
- Time Scale
  - Milestone 1
  - Milestone 2
  - Etc.

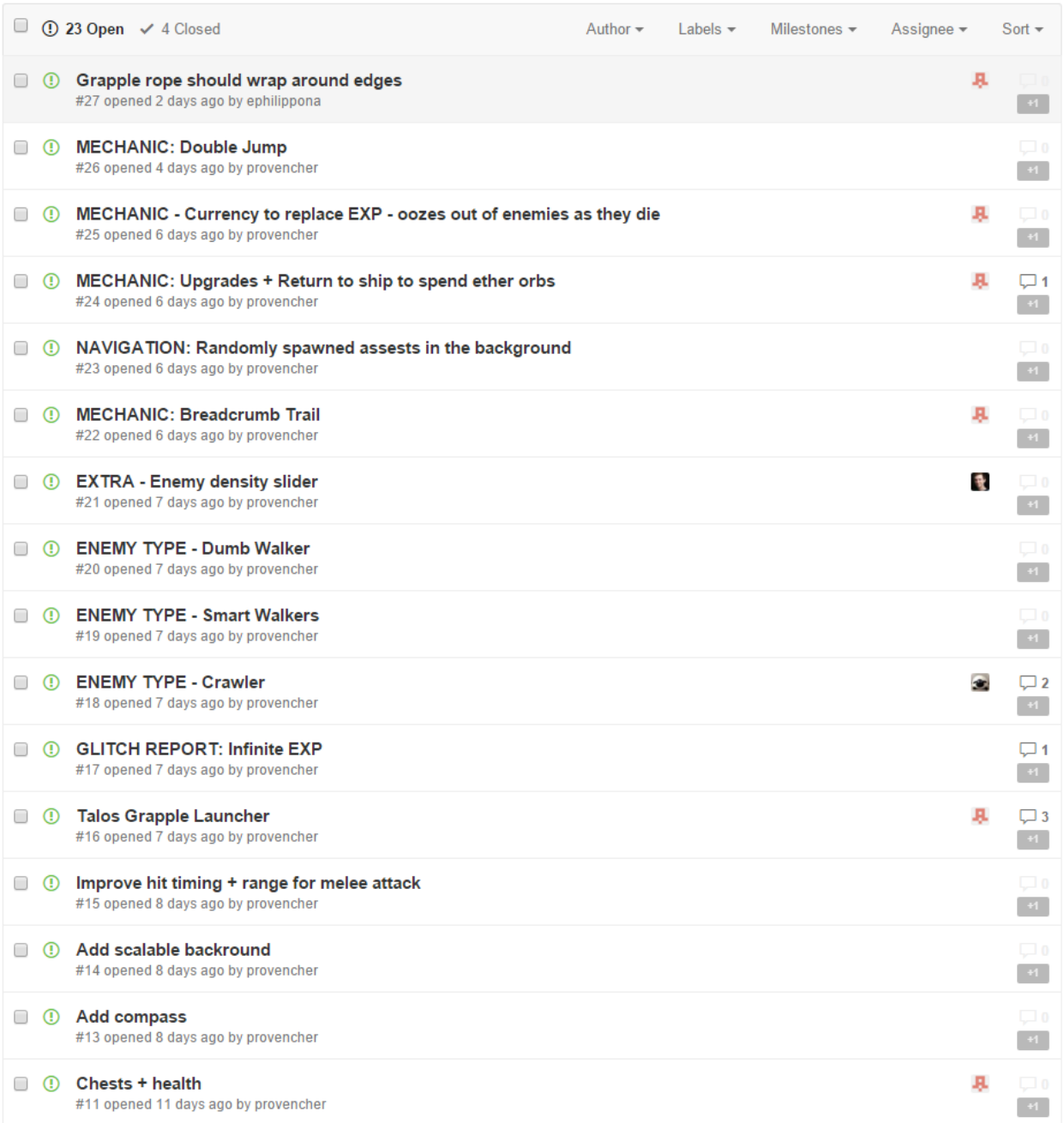
- <Object #4>
- Time Scale
  - Milestone 1

- Milestone 2
- Etc.

Issue Tracking

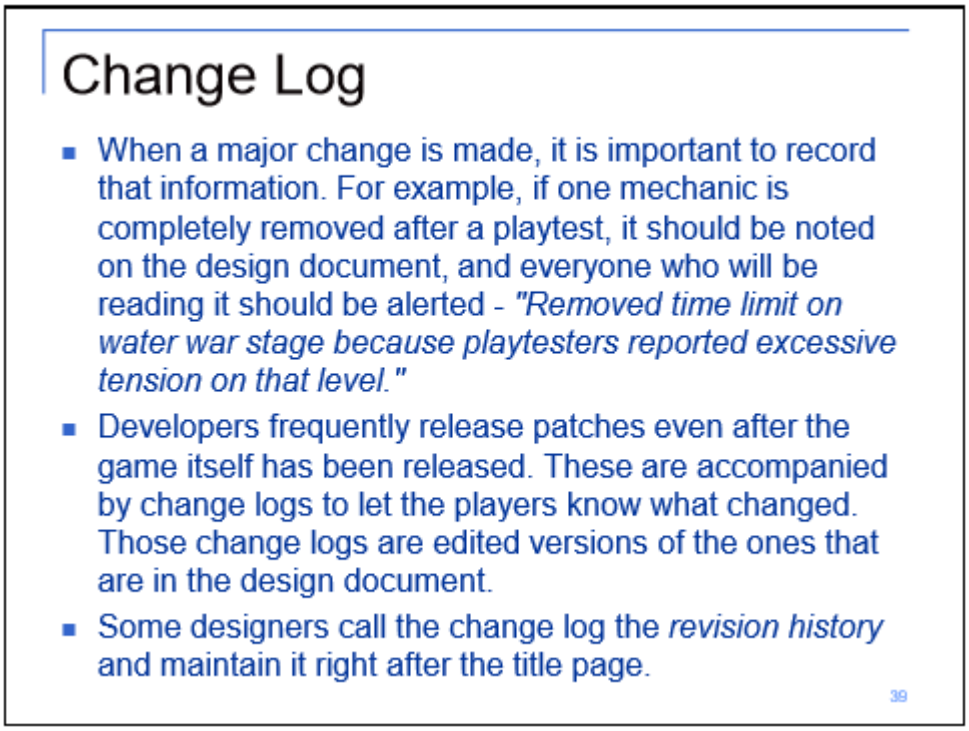
Talos Origins is setup on a private GitHub Repository in order to facilitate development progress and ensure that we can always access functional builds of the game by reverting faulty commits if necessary.

The screenshots below showcase our issue tracking, which allows us to assign team members with specific tasks, along with openly offering feedback for specific changes.



- **Level System/ Upgrades**
  #10 opened 11 days ago by provencher
  0 +1
- **Add Boss Stages**
  #9 opened 11 days ago by provencher
  0 +1
- **Add UI - Health bar + Action point bar + exp bar -- LOW PRIORITY**
  #7 opened 14 days ago by provencher
  0 +1
- **Program EXP from killing enemies**
  #6 opened 14 days ago by provencher
  1 +1
- **Add Talos Physics + Controls**
  #3 opened 14 days ago by provencher
  0 +1
- **Animate Talos**
  #2 opened 14 days ago by provencher
  1 +1
- **Add aiming reticle on screen controlled by mouse**
  #1 opened 14 days ago by provencher
  1 +1

Change Log


Change Log
When a major change is made, it is important to record that information. For example, if one mechanic is completely removed after a playtest, it should be noted on the design document, and everyone who will be reading it should be alerted - "Removed time limit on water war stage because playtesters reported excessive tension on that level."
Developers frequently release patches even after the game itself has been released. These are accompanied by change logs to let the players know what changed. Those change logs are edited versions of the ones that are in the design document.
Some designers call the change log the revision history and maintain it right after the title page.
39