Development of a Procedural Storytelling Game with Dynamic Dialogues

Frantzolas Christos

Objective

- Procedural Storytelling
- Dynamic Dialogs
- Text-based game (Interactive Fiction)
- Natural Language Input and Output
- IF Game Development Platform

2

The genre

- Graphical Adventures
- Parser Games
- Hyperlink-based Games
- Menu-based Games
- Choice-based Games
- Puzzles vs Narrative
- Domain-Specific Programming Languages (TADS, Inform)

The history

YOU ARE STANDING AT THE END OF A ROAD BEFORE A SMALL BRICK BUILDING.
AROUND YOU IS A FOREST. A SMALL STREAM FLOWS OUT OF THE BUILDING AND
DOWN A GULLY.
>ENTER BUILDING
YOU ARE INSIDE A BUILDING, A WELL HOUSE FOR A LARGE SPRING.
THERE ARE SOME KEYS ON THE GROUND HERE.
THERE IS A SHINY BRASS LAMP NEARBY.
THERE IS FOOD HERE.
THERE IS A BOTTLE OF WATER HERE.
>GET LAMP
OK
>INVENTORY
YOU ARE CURRENTLY HOLDING THE FOLLOWING:
BRASS LANTERN
>QUIT
DO YOU REALLY WANT TO QUIT NOW?
>Y
OK

YOU SCORED 27 OUT OF A POSSIBLE 350, USING   4 TURNS.
YOU ARE OBVIOUSLY A RANK AMATEUR. BETTER LUCK NEXT TIME.
TO ACHIEVE THE NEXT HIGHER RATING, YOU NEED 9 MORE POINTS.

.▄

Advent

Infocom

Puzzle games

TADS & Inform

Competitions

Literary Complexity

Façade, Galatea

AI Dungeon

The platforms
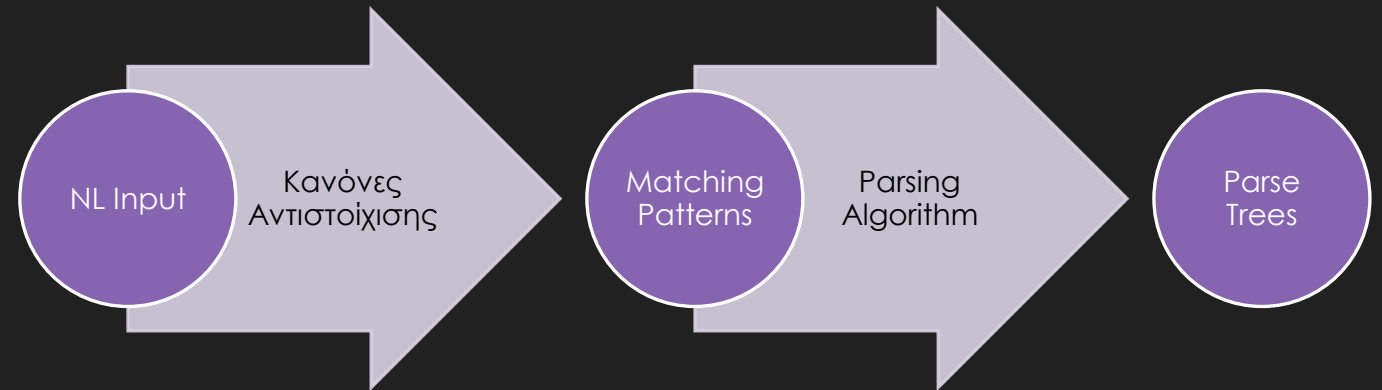
TADS

Inform

Twine

Ren'Py

Quest

Squiffy

Theoretical Background

Parsers

- Context Free Grammars
- Semantic Parsing
  - Rule Based Approaches
  - Statistical Methods
  - Neural Networks

NL Input → Κανόνες Αντιστοίχισης → Matching Patterns → Parsing Algorithm → Parse Trees

Procedural Generation

- Content Generation:
  - Pseudo random number generators
  - Parametric Methods
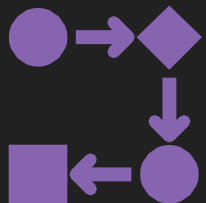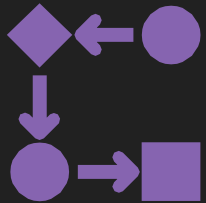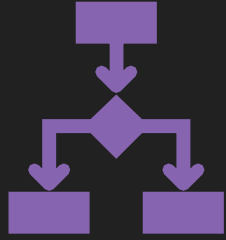 - Generative Grammars
  - Spatial Methods
  - Simulations
  - Genetic Algorithms

Narrative Generation

- Mark-Up & Templates
- Tags
- Generative Grammars
- Augmented CF Grammars
- Graph Grammars
- Agent Interactions
- Plot Mining
- GPT-2 & 3

Dynamic Dialogs

- Διεπαφή
    - NL Input and "Cheap" AI
    - Yes/No Conversations
    - Talk To
    - Menu-based
    - Ask/Tell

Scene

Topic

Quip

Facts

NPC
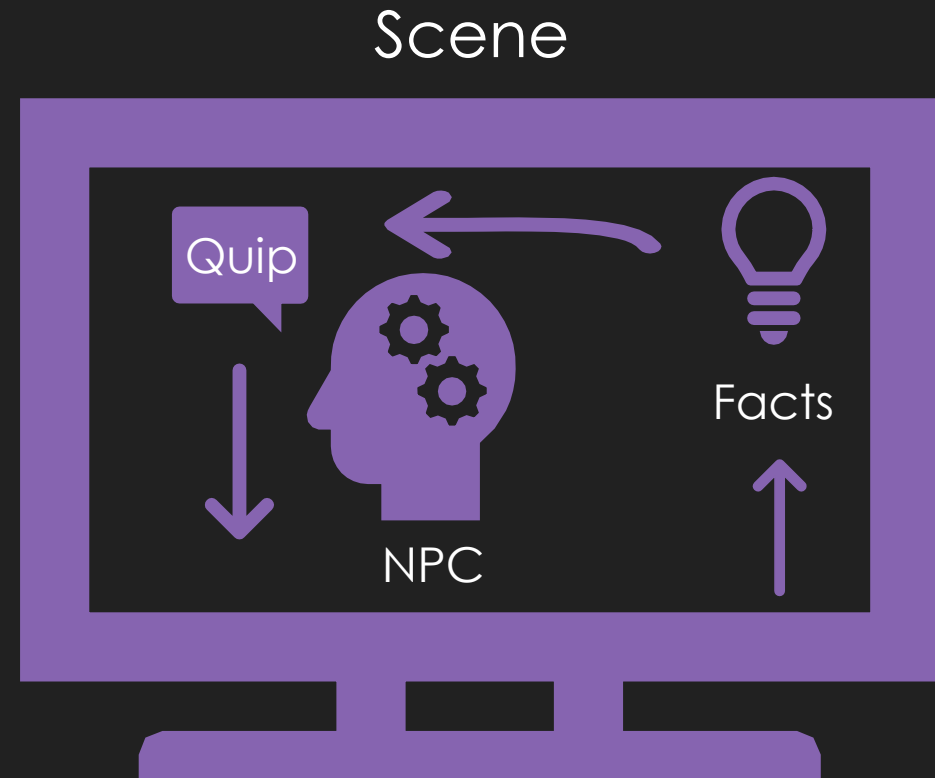
Player

IF Game Design
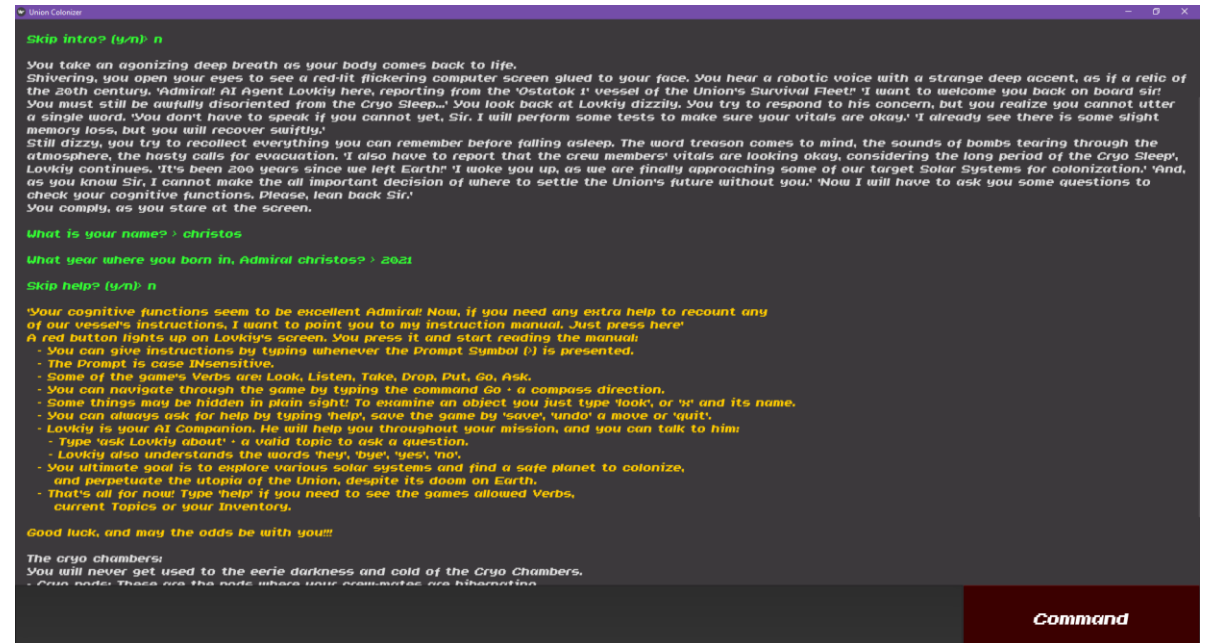
○ Man-made and procedural content

○ "Opening up" the plot

○ Incorporating Knowledge

○ Player Experience:

    ○ Plot Cohesion

    ○ Player Stories

Designing and Developing the Game

Union Colonizer

Game Plot:
Union Colonizer

- Plot and Player Objectives
- Dialogs
- GUI

Tools

- Python vs Γλώσσες Ειδικού Σκοπού
- Object Oriented Programming
- Threading
- JSON File Format
- Numpy random
- NLTK
- GUI – kivy Module

Basic Structure

Game Class

File Handling

Loader & Saver Classes

Custom JSON module

JSON Game Files

Chapters
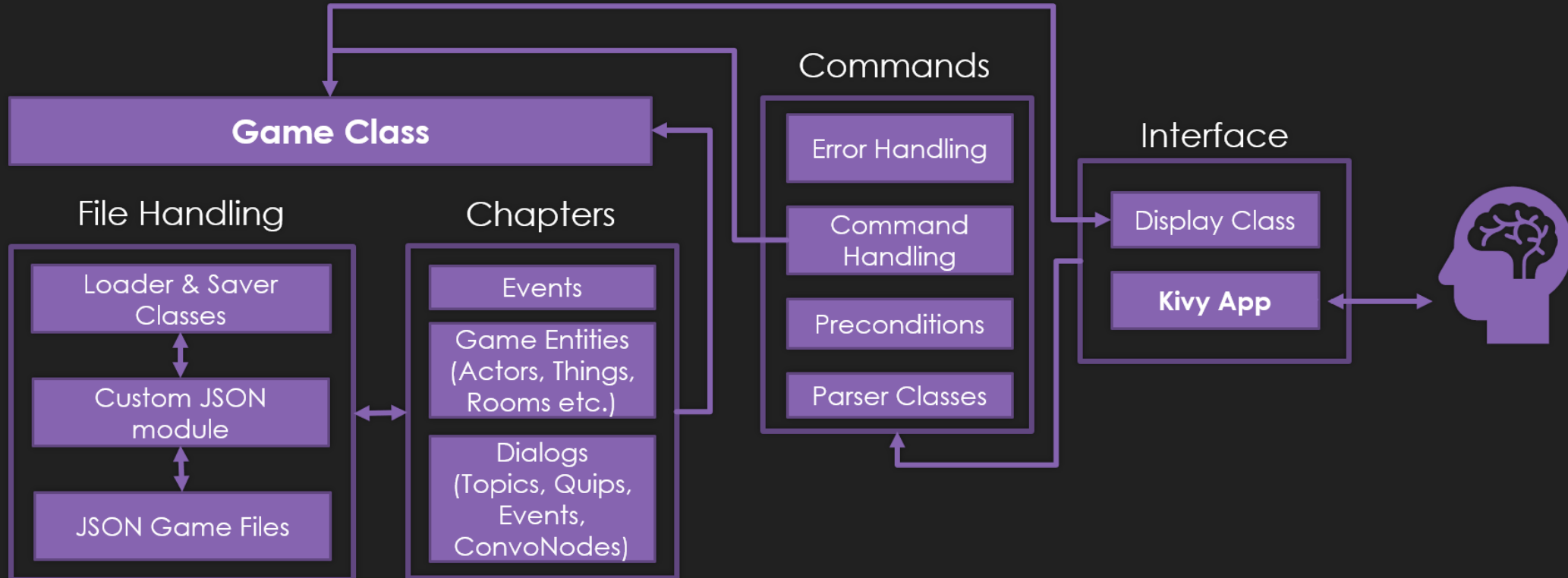
Events

Game Entities (Actors, Things, Rooms etc.)

Dialogs (Topics, Quips, Events, ConvoNodes)

Commands

Error Handling

Command Handling

Preconditions

Parser Classes

Interface

Display Class

Kivy App
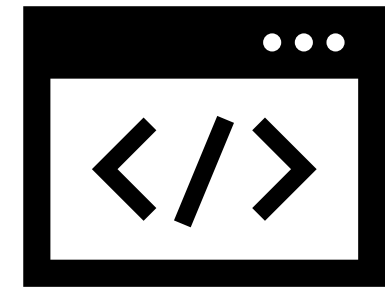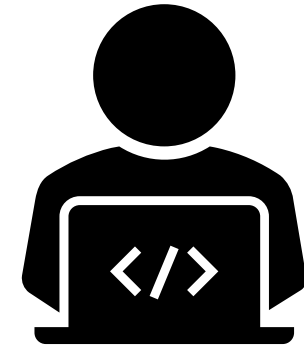
The Command Handling Pipeline

Preparsing → Command Type Classification → Tokenization → POS Tagging

Multiple Objects Processing ← Noun Phrase Parsing ← Syntax Parts identification ← Syntax Trees

Character Check → Qualifier Check → Preconditions → Command Execution
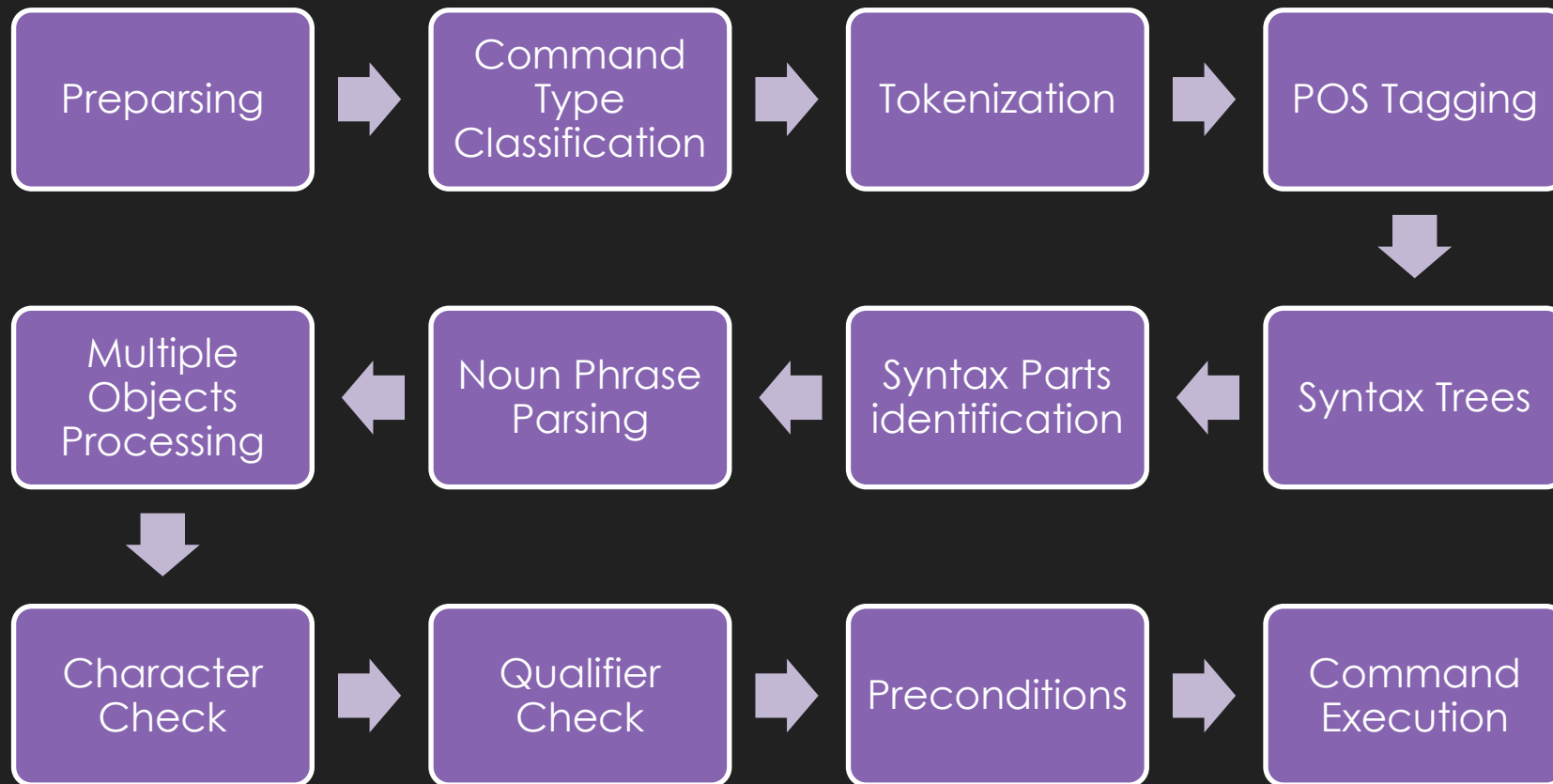
Entities

- Entity Class
  - Things
  - Rooms
  - Actors
  - Special Objects

Events

- Event Class
- Conditions
- Game Queries
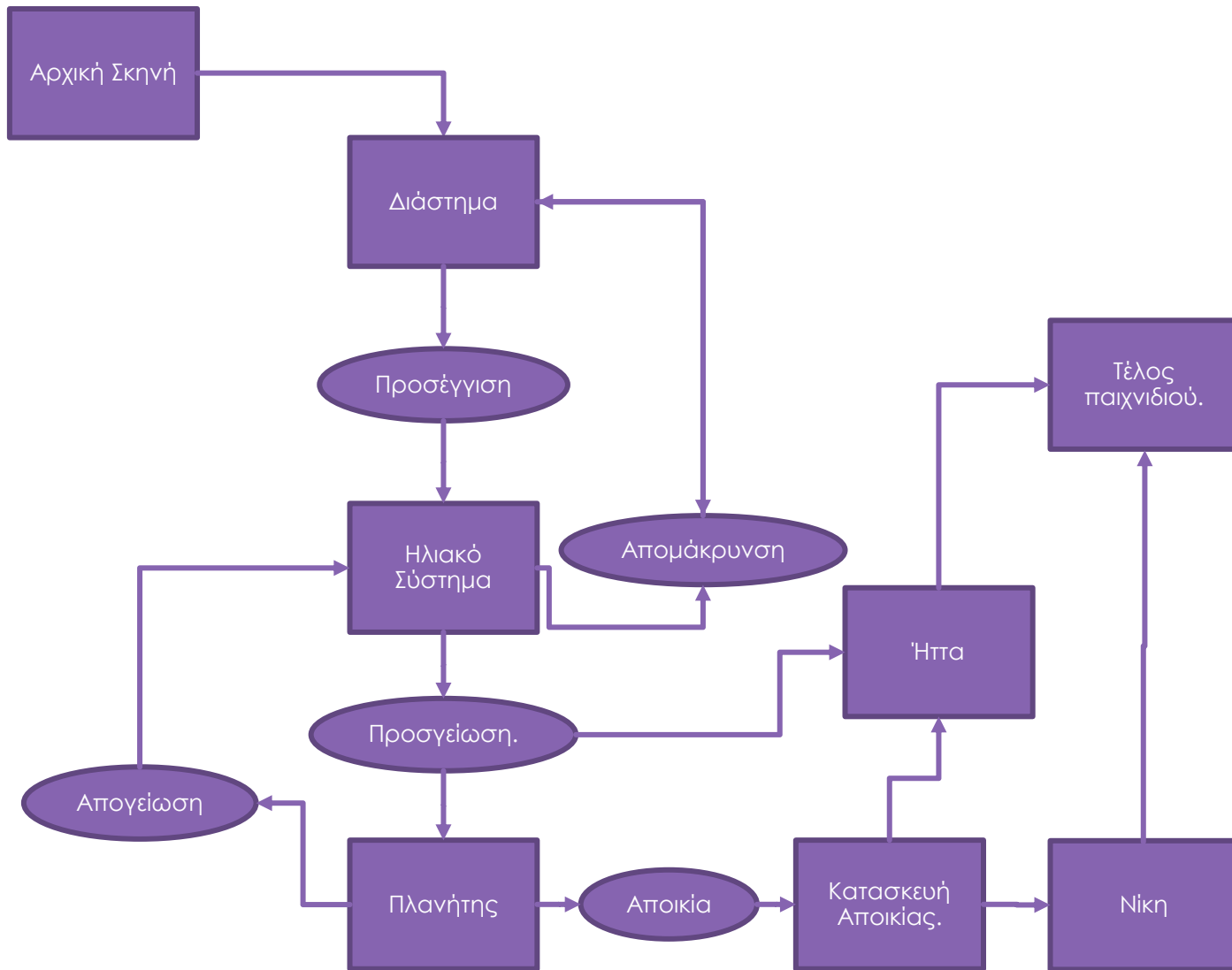- Game State Changes

Dialogs

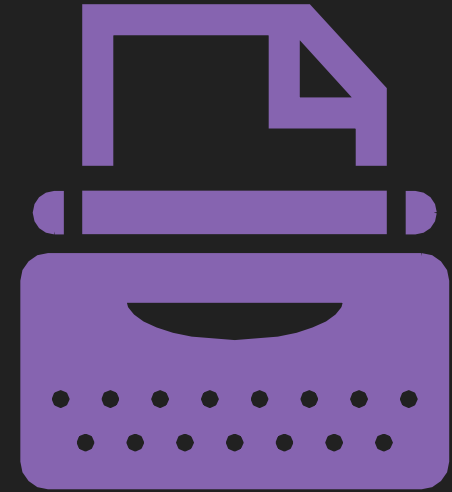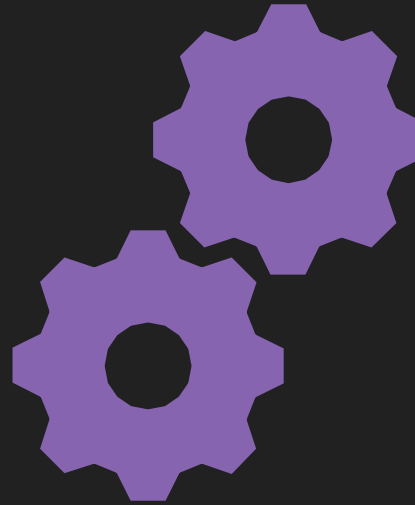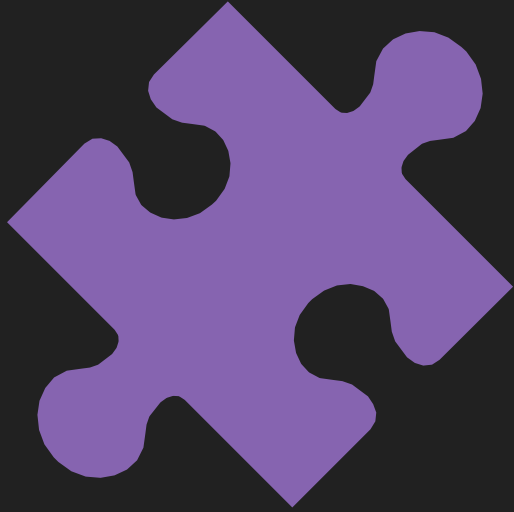Dialogs as Events

Topics

Dialog Events

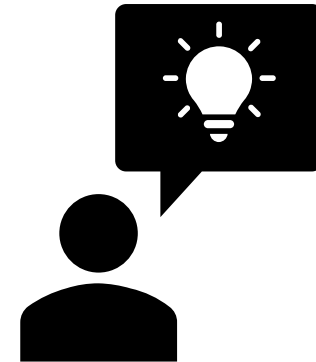Quips

Conversation Trees

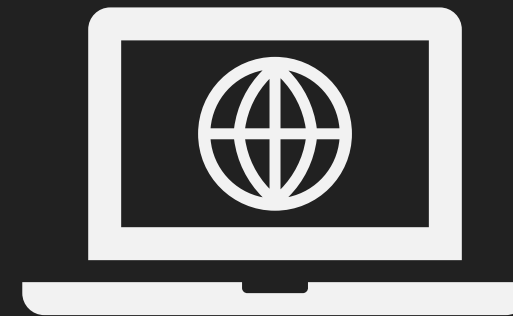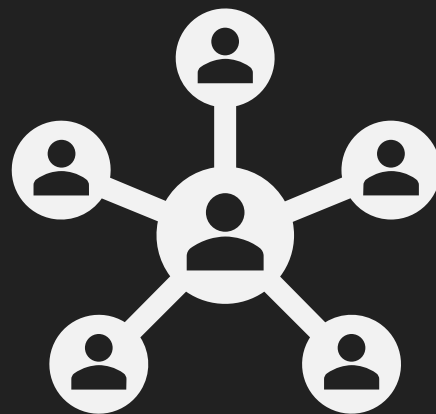**Chapters**

Procedural Generation

- Solar Systems
- Planets
- Descriptions
- Names
- Rooms
- Threats

Future Improvements

Beta Testing & Future Improvements

- More content
  - Objects
  - Procedural Generation
  - Descriptions
  - Plot
- Error Descriptions
- Dialogs
- Knowledge graph

- Mark-Up Strings
- Performance
  - Memory
  - Speed
- Authoring Tools
- Debugging Tools

Github

LinkedIn

Thank you! Any Questions?