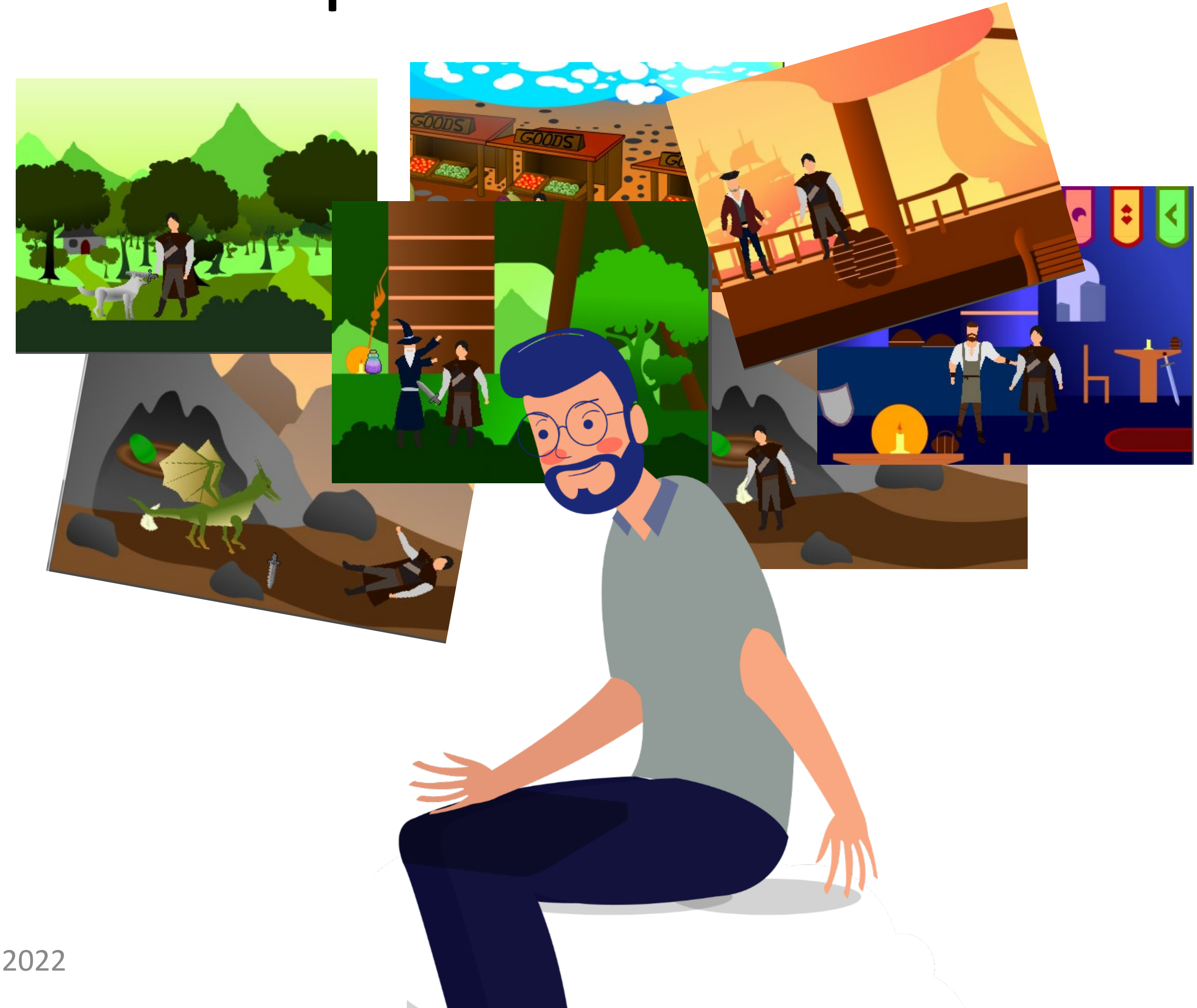
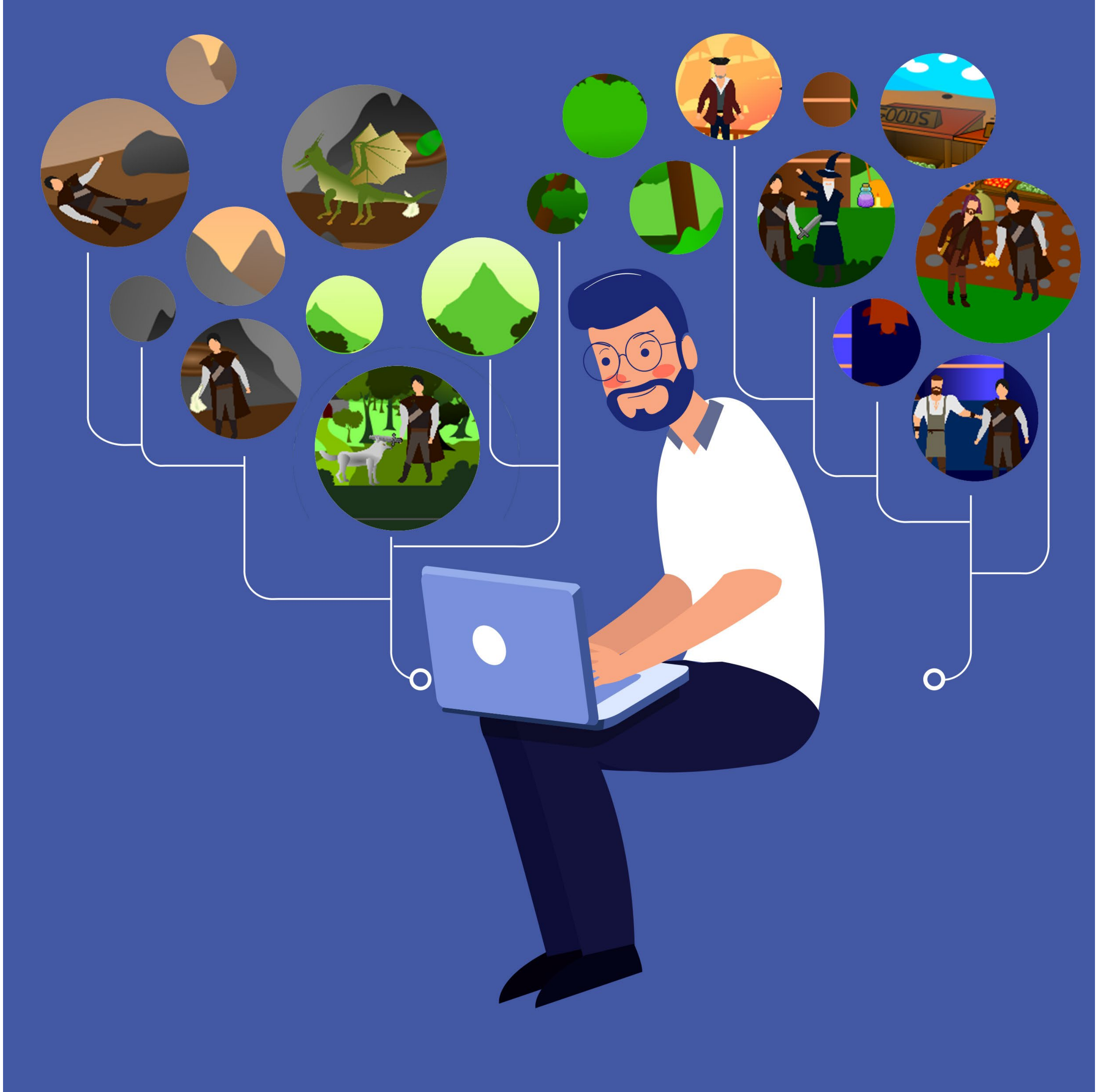
Plot generation of computer narrative game with graph transformations

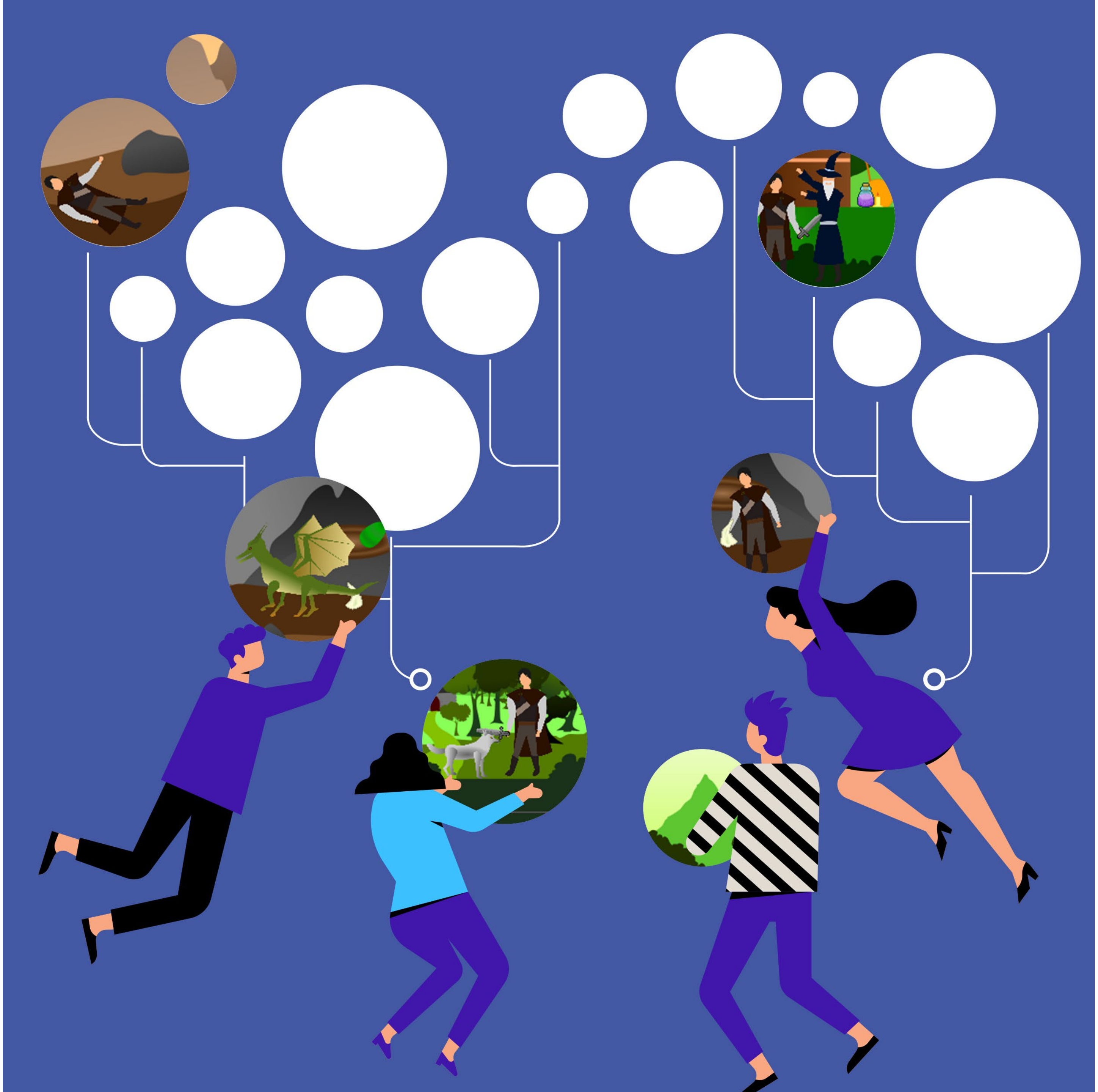
Iwona Grabska-Gradzińska
WFAIS UJ

AIRA, 13 X 2022

Game plots are nonlinear




GOODS



Plot formalisation

Владимир Яковлевич Пропп
Władimir Propp (1895–1970)

Propp model

The goal of Wladimir Propp was to describe the morphology of Russian folk tales, based on 7 types of characters and 31 types of actions.

$$\text{б}^{1}\ e^{1}\ b^{1}\ A^{1}\ C \uparrow \frac{[\text{Д}^{1}\ \text{Г}^{1}{}_{neg}\ Z_{neg}]}{\text{Д}^{7}\ \text{Г}^{7}\ Z^{9}}\ R^{4}\ \text{Л}^{1} \downarrow [\text{Пр}^{1}\ \text{Д}^{1}\ T^{1}\ Z^{9} = \text{Сп}^{4}]^{3}\text{:}$$

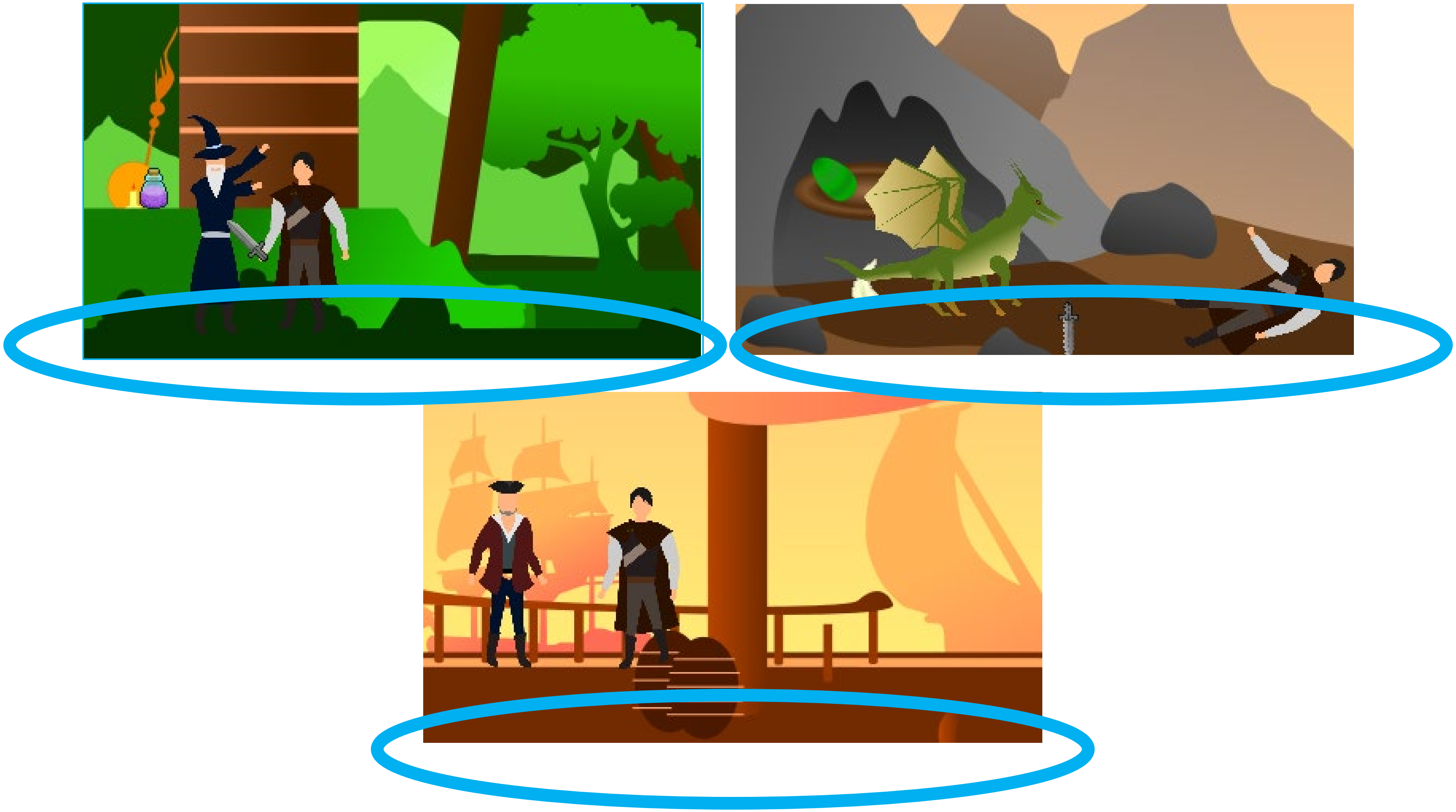
$$e^{2}\ \text{б}^{2}\ w^{1}\ \text{г}^{3}\ b^{2}\ g^{3}\ A^{1} \frac{C \uparrow (\text{Д}^{1}\ \text{Г}^{1}{}_{neg})^{3}\ (\text{Д}^{1}\ \text{Г}^{1}{}_{neg})^{3}\ Z_{contr}}{B^{4}\ C \uparrow (\text{Д}^{1}\ \text{Г}_{poz})^{3}\ (\text{Д}^{1}\ \text{Г}^{1}{}_{poz})^{3}}\ \text{Б}^{1} - \text{П}^{1}\ \text{Л}^{4} \downarrow$$

Game typology

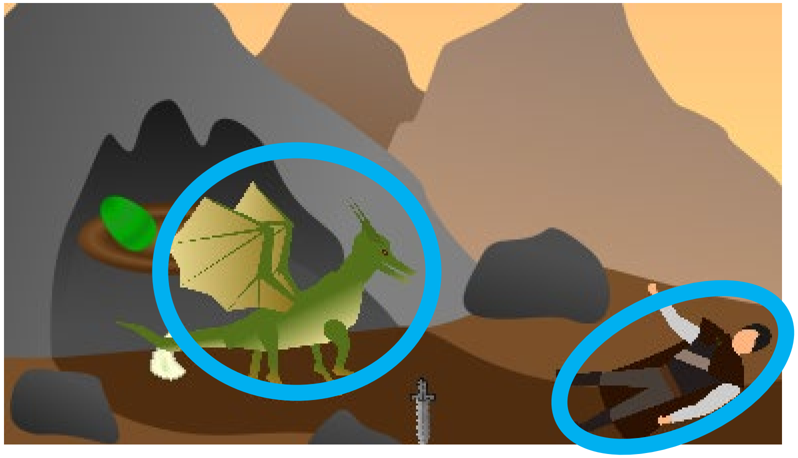
Genre	Role of the plot
Action games	Sometimes the plot allows you to explain the initial world, justifies the effects of the player’s actions.
Adventure games	Most often, the plot is the basis for the mechanics. Cause-effect relationships between events can be discovered in fictional interactions. It is more linear than in role-playing games, the player’s decision making is limited by story nodes.
Puzzle games	If the plot is present at all, it has an ornamental significance.
Role-playing game	Often RPG games have a world more open than typical adventure games, the plot allows you to predict the effects of the long-term actions and plan long-term strategies, and use the player’s general knowledge of the universe.
Simulations	If the plot is present at all, it has an ornamental significance.
Sport games	If the plot is present at all, it has an ornamental significance.
Strategy games	The plot presents the world in which the game is set.

Game world

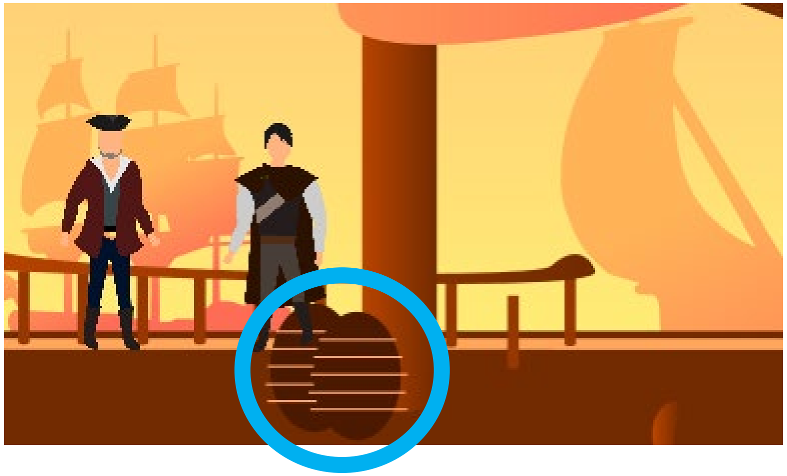
Game world structure



Game world structure



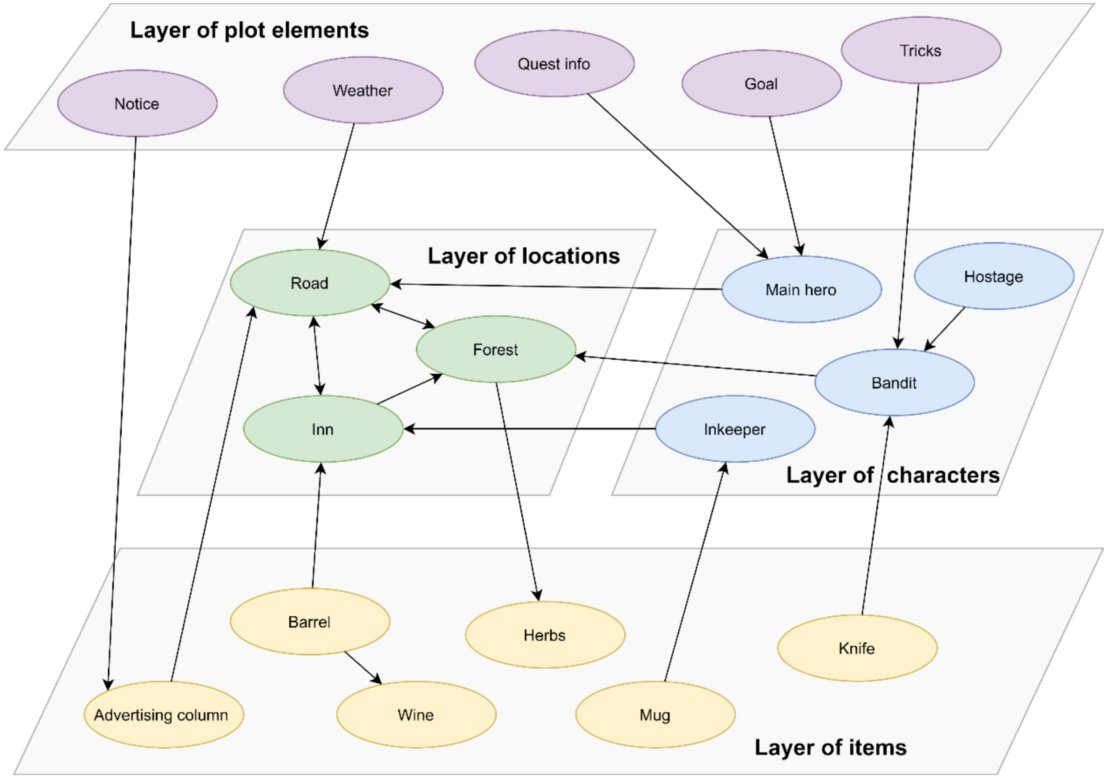
Game world structure



Graph model of a narrative game

Based on the concept of *n*-layered graph

Includes the game world description and it’s modifications expressed as graph productions, resulting from the player's actions.



n-layered graph

An n-layer graph for natural number $n \geq 2$ is:

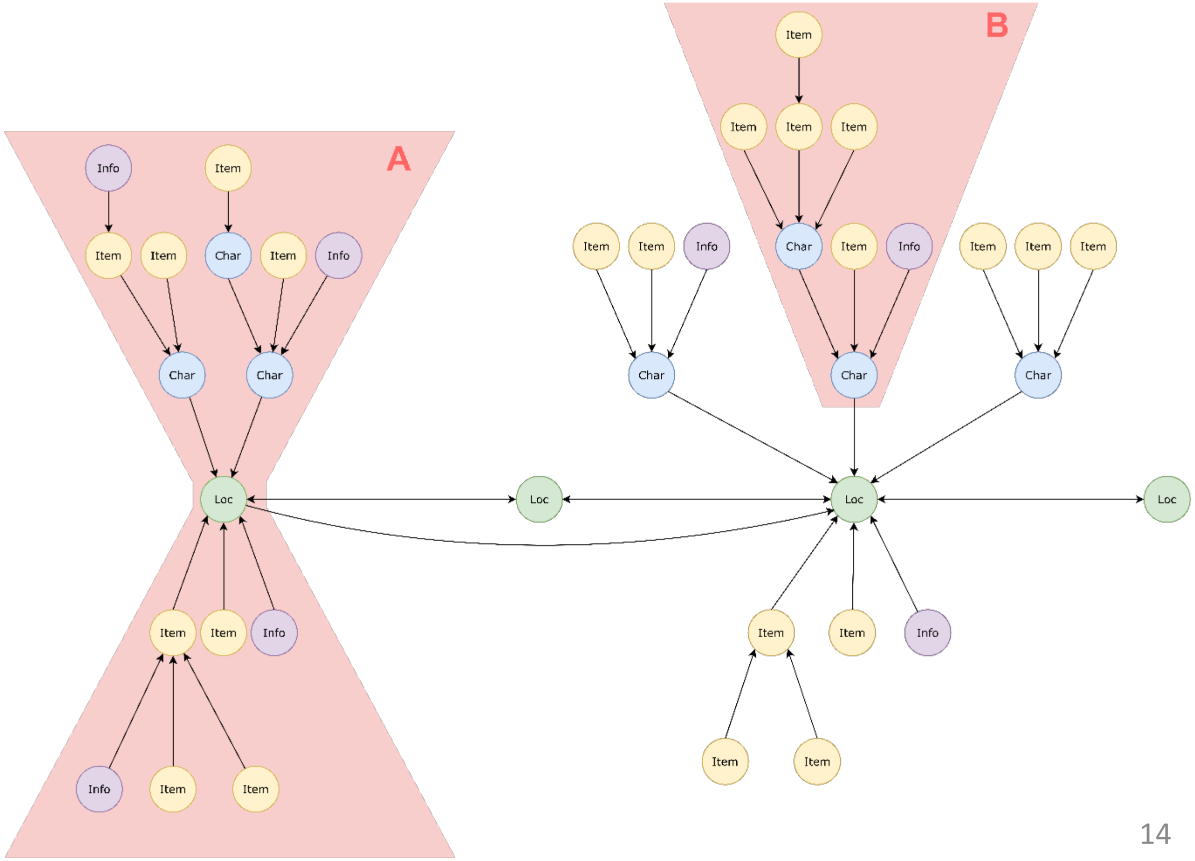
$$H = (X, ch, E, s, t, lab),$$

where:

- X is an n-element set of nodes, called layers, such that $\forall G \in \Gamma(L, A) : \; V_G \cap X = \varnothing$,
- $ch : \; X \ni x \rightarrow G_{ch(x)} \in \Gamma(L, A)$ is a function that places a nested graph inside a node of X, such that for all graphs $G_{ch(x)}, x \in X$ the sets of nodes, i.e. $V_{ch(x)}$, are disjoint, the sets of edges, i.e. $E_{ch(x)}$, are disjoint, and the sets of used labels, i.e. $lab_{ch(x)}(V_{ch(x)})$, are disjointed,
- E is a set of edges, called external, which is disjointed with all edges $E_{ch(x)}$ for $x \in X$,
- $s, t : \; E \rightarrow \bigcup_{x \in X} V_{ch(x)}$ are source and target functions defined on the set E respectively,
- external edges are interlayer, i.e. $\forall x \in X, e \in E : \; s(e) \in V_{ch(x)} \Longrightarrow t(e) \notin V_{ch(x)}$,
- $lab : \; X \rightarrow L_H$ is an injective layer labelling function.

Multisheaf graph

Graph is a multisheaf graph if there is one and only one layer with no restrictions to node connections and nodes from other layers create trees with nodes from above mentioned distinguished layer as roots.



Multisheaf graph

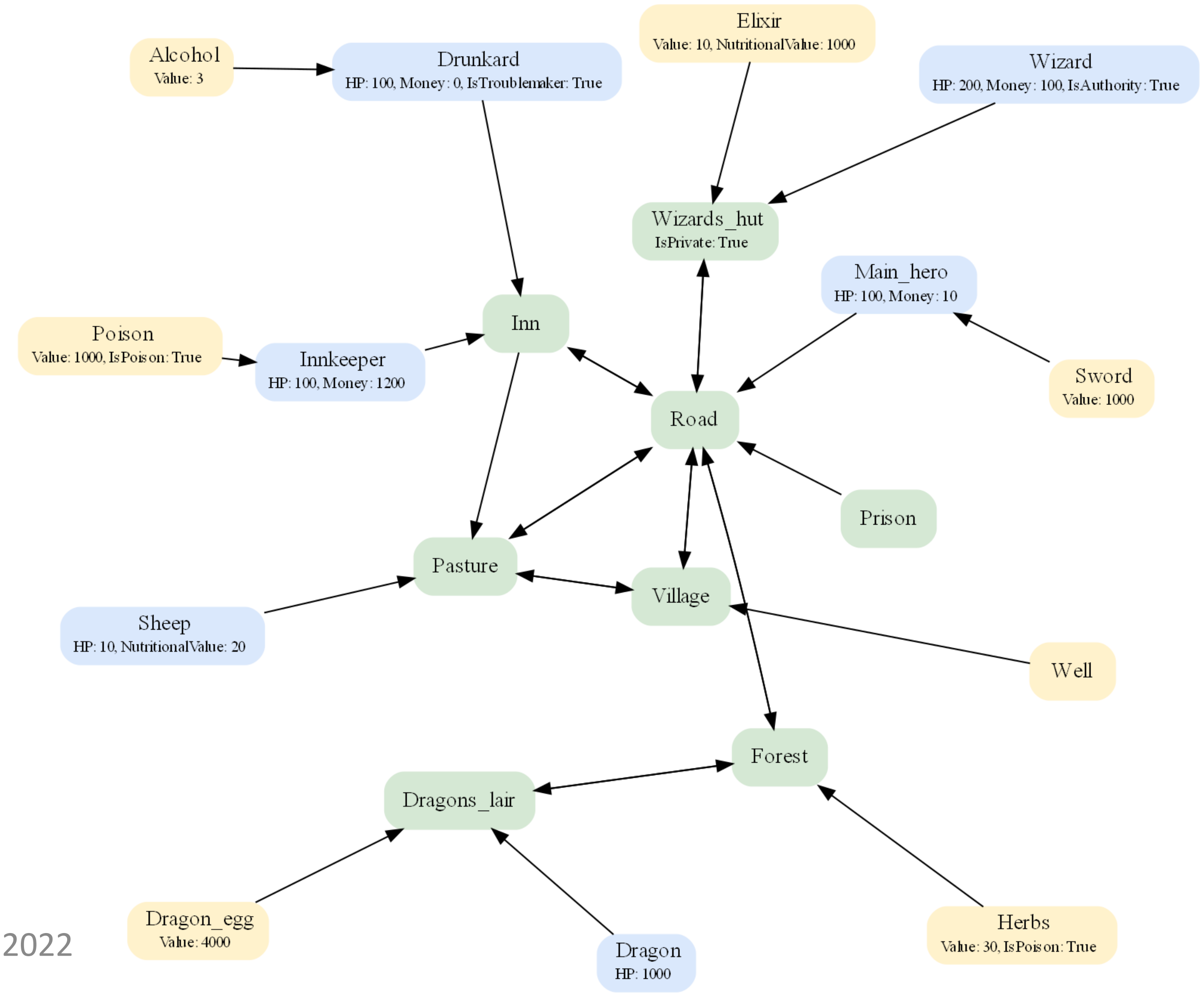
An n-layer graph

$$H = (X, ch, E, s, t, lab)$$

is called a multisheaf graph if there is $x_R \in X$ such that:

- there is no external edge $e \in E$ such that source node $s(e)$ is in the graph $G_{ch(xR)}$ nested in x_R and the target node $t(e)$ is not in the set of the graph $G_{ch(xR)}$,
- there are no paths in graph H in which any node occurs twice if this node is not in the $G_{ch(xR)}$,
- every node, which is not in $G_{ch(xR)}$, is the source of exactly one edge (internal or external).

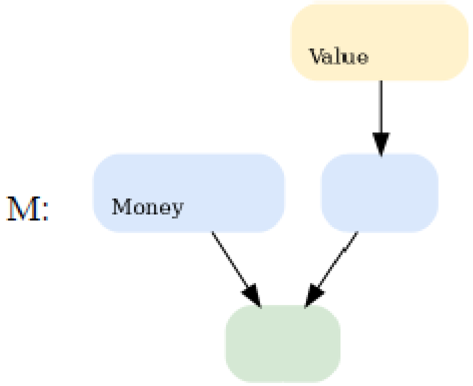
Game world as an instance of *n*-layered graph



Actions in the game

Generic production

Generic production *p* is triple (*M*, *C*, *P*), where *M* is an *n*-layer graph with partial labelling, *C* is a predicate checking procedure, and *P* is a graph modifying procedure.



C:

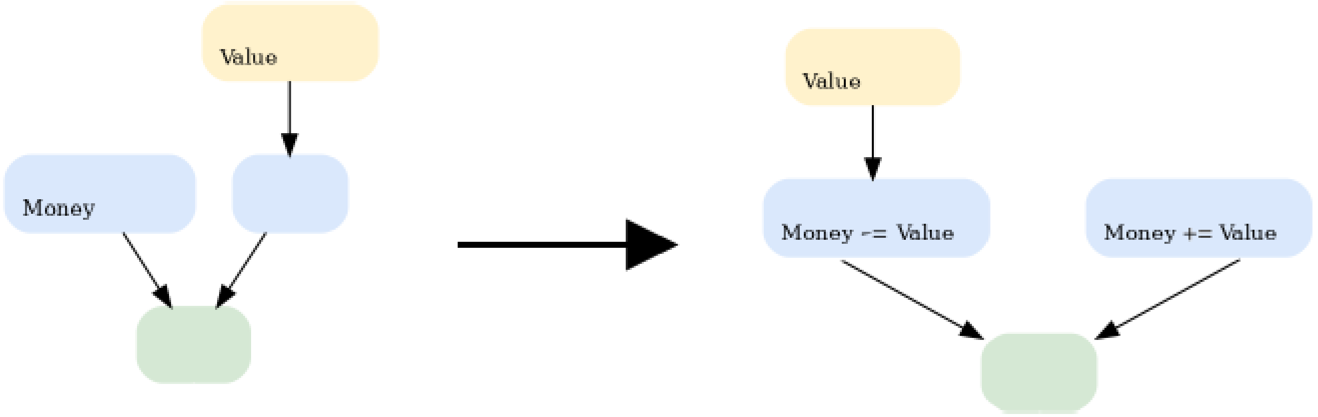
- Accepts match if character's attribute Money is the float number greater then item's attribute Value.

P:

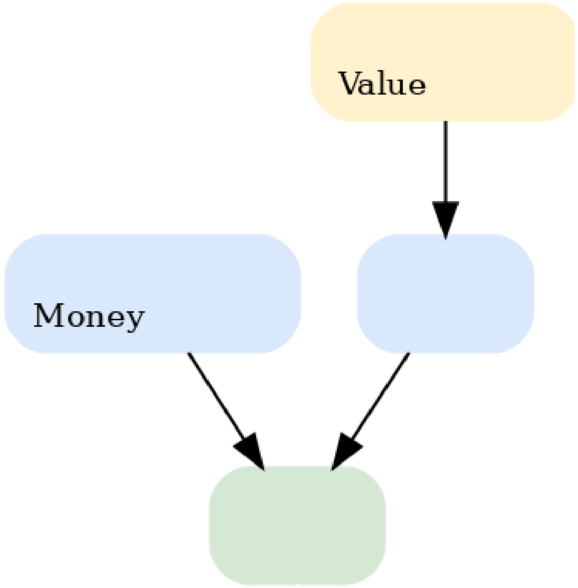
- Remove the edge between the item (yellow node) and its parent character (blue node).
- Add an edge between this item and the other character (blue node).
- Decrease the Money attribute of first character by the amount of Value attribute.
- Increase the Money attribute of second character by the amount of Value attribute.

Left- and right-hand-side representation

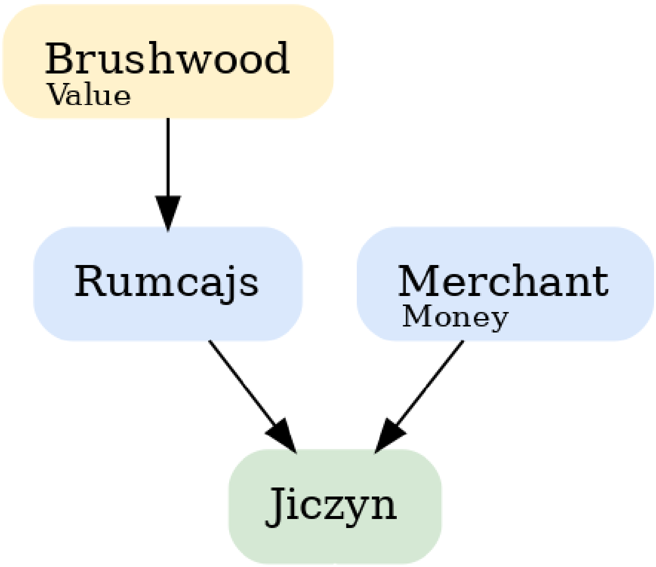
In some cases production can be transform to RHS-LHS schema



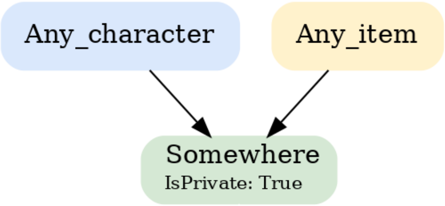
Left-hand-side of the productions as a partially labelled *n*-layer graph

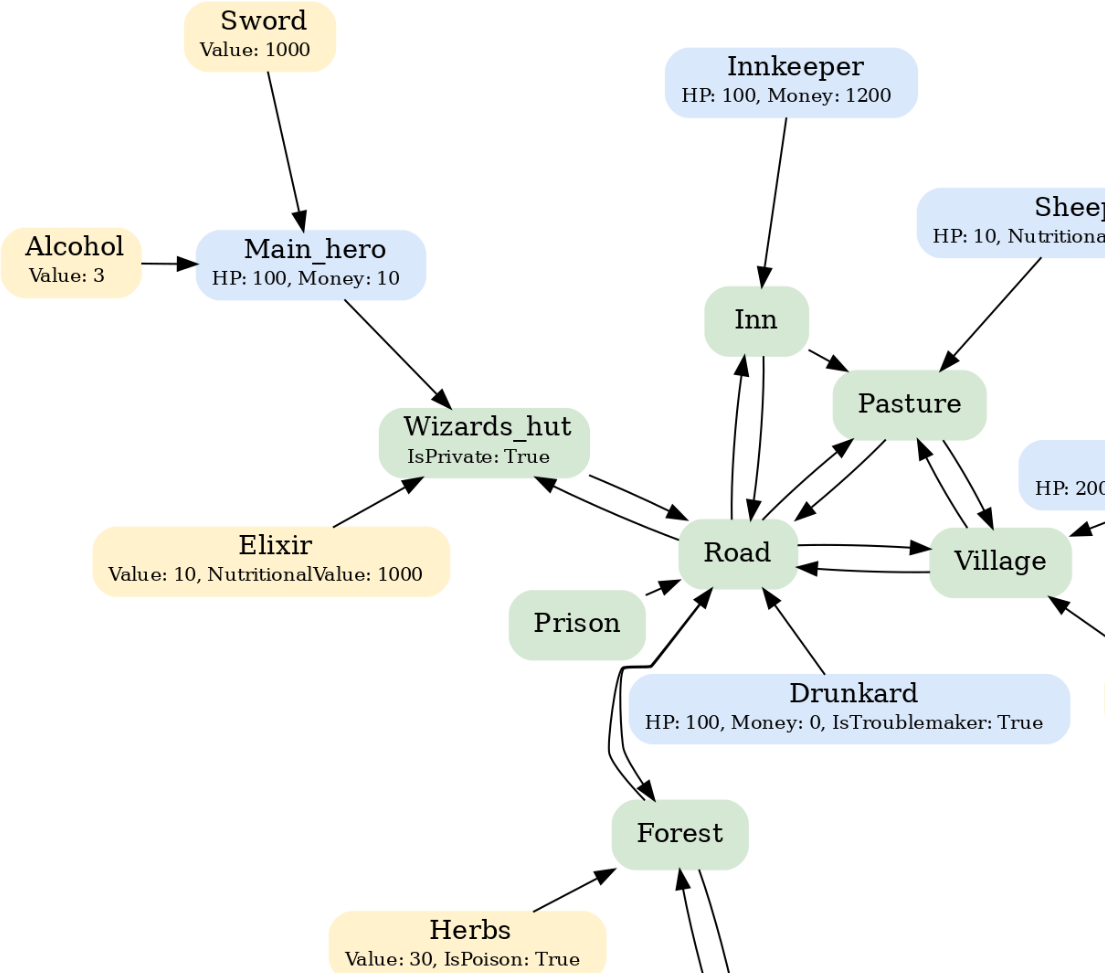


Left-hand-side of the productions as a totally labelled *n*-layer graph

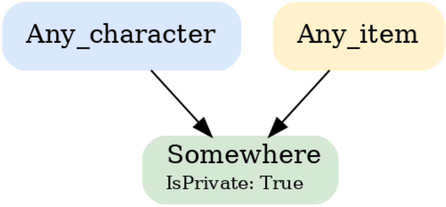


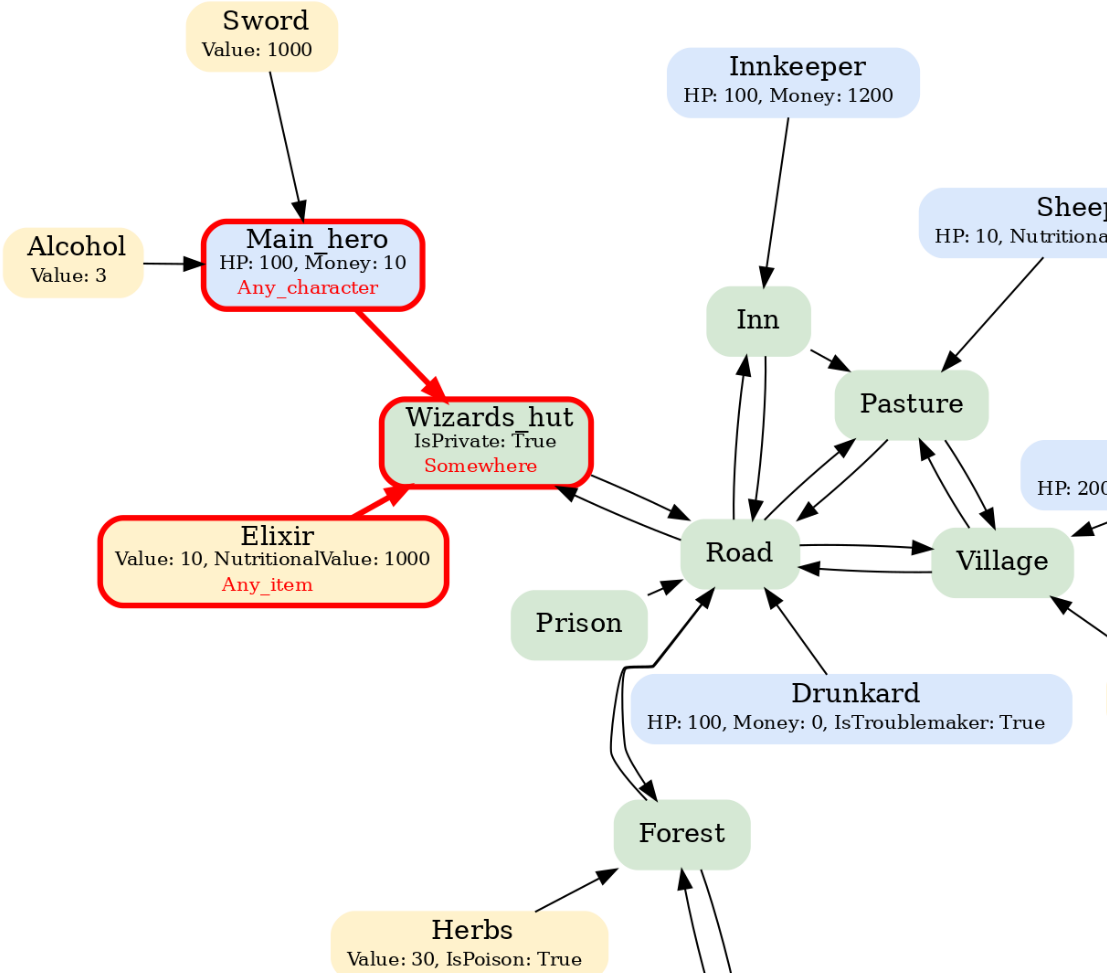
Left-hand-side match



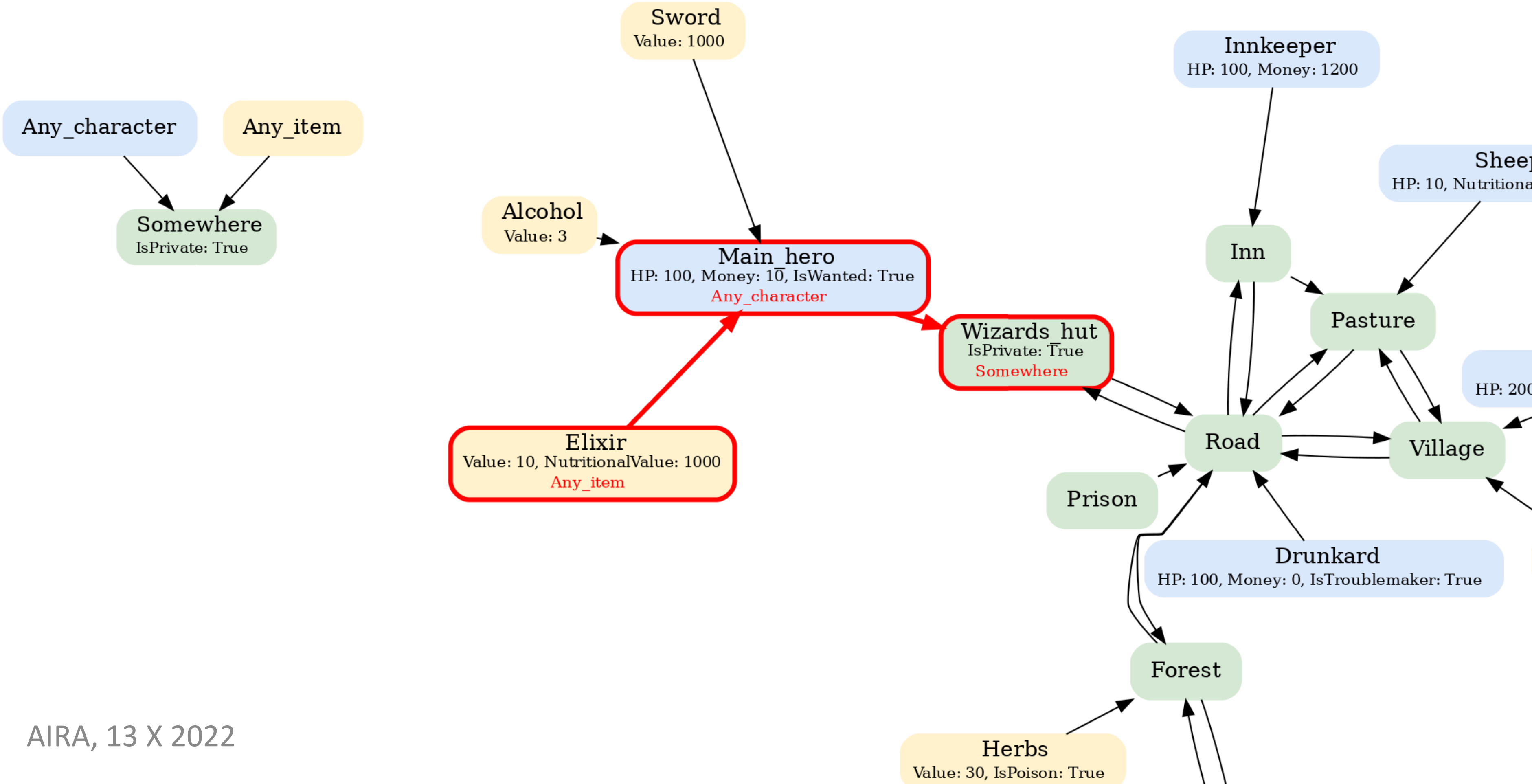


Left-hand-side match





Left-hand-side match



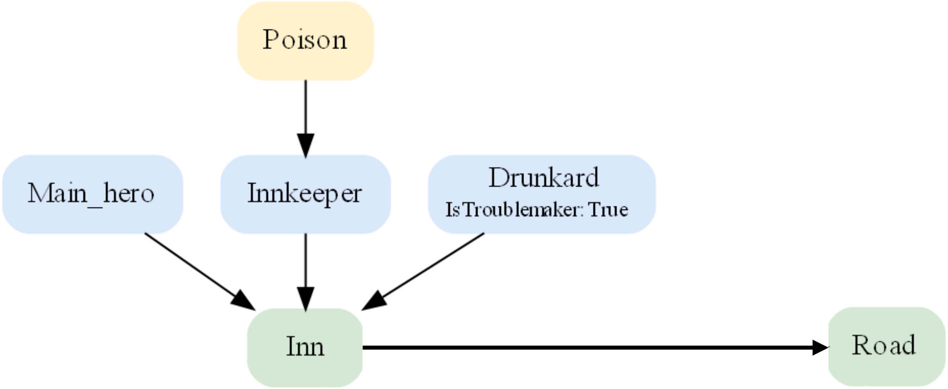
Generic match of partially labelled attributed graphs

Let K be a partially or totally labelled attributed graph. Let Q be a partially or totally labelled attributed graph or an instance of such graph. A generic match $k : K \rightarrow Q$ is a pair of injective functions $k_E : E_K \rightarrow E_Q$ and $k_V : V_K \rightarrow V_Q$ which preserve properties of nodes and edges:

- $\forall\, e \in E_K :\ k_V(s_K(e)) = s_Q(k_E(e))$
- $\forall\, e \in E_K :\ k_V(t_K(e)) = t_Q(k_E(e))$
- $\forall\, v \in V_K :\ v \in dom(lab_K) \Longrightarrow lab_K(v) = lab_Q(k_V(v))$
- $\forall\, v \in V_K :\ attr_K(v) \subseteq attr_Q(k_V(v))$
- $\forall\, e \in E_K :\ attr_K(e) \subseteq attr_Q(k_E(e))$

Match algorithm facilitations

- The subject of production is the player who is in a specific location, so we have two nodes (location and character) matched for free.
- The left sides of the production are usually small graphs because they reflect the actions of the hero, which usually involve interactions with one or two other characters and one or several items and short paths.
- When taking action, we rely on what we see around, so usually only the sheaf of the main location is expanded



Match algorithm chalenges

- The match graph is partially labeled
- Node labels are not unique
- We cannot rule out an ambiguity in the matching

Model’s features

Crucial for implementation process

- The graph is not only n-layer but also multisheaf
- There are no right-hand sides of production, only a sequence of instructions
- We allow partial labelling of nodes on the left side of production
- We check the acceptable range of attribute values using predicates

Tworzenie narracji komputerowej gry fabularnej z użyciem transformacji grafowych,
I. Grabska-Gradzińska, PhD thesis

Automatic story generation based on graph model using Godot Engine,
I. Grabska-Gradzińska, L. Nowak, E. Grabska

Towards Automatic Generation of Storyline Aided by Collaborative Creative Design
I. Grabska-Gradzińska, E. Grabska, L. Nowak, W. Palacz

Graph Rules Hierarchy as a Tool of Collaborative Game Narration Creation
L. Nowak, I. Grabska-Gradzińska, E. Grabska, W. Palacz, M. Wrona, A. Konopka et al.

Yuhua Luo (Ed.)

LNCS 12341

Cooperative Design, Visualization, and Engineering

17th International Conference, CDVE 2020
Bangkok, Thailand, October 25–28, 2020
Proceedings

Springer

Leszek Rutkowski · Marcin Korytkowski
Rafał Scherer · Ryszard Tadeusiewicz
Lotfi A. Zadeh · Jacek M. Zurada (Eds.)

LNAI 10245

Artificial Intelligence and Soft Computing

16th International Conference, ICAISC 2017
Zakopane, Poland, June 11–15, 2017
Proceedings, Part I

Part I

Springer

Carmelo Ardito · Rosa Lanzilotti ·
Alessio Malizia · Helen Petrie ·
Antonio Piccinno · Giuseppe Desolda ·
Kori Inkpen (Eds.)

LNCS 12936

Human-Computer Interaction – INTERACT 2021

18th IFIP TC 13 International Conference
Bari, Italy, August 30 – September 3, 2021
Proceedings, Part V

Part V

ifip

INTERACT 2021

Springer

MOREMEDIA

Implementation

The StoryGraph system was implemented using Python version 3.9. The prepared application is a console application, but it enables access via WWW thanks to REST API. REST API endpoints ensuring data validation and visualization for users who do not have access to the project source code.

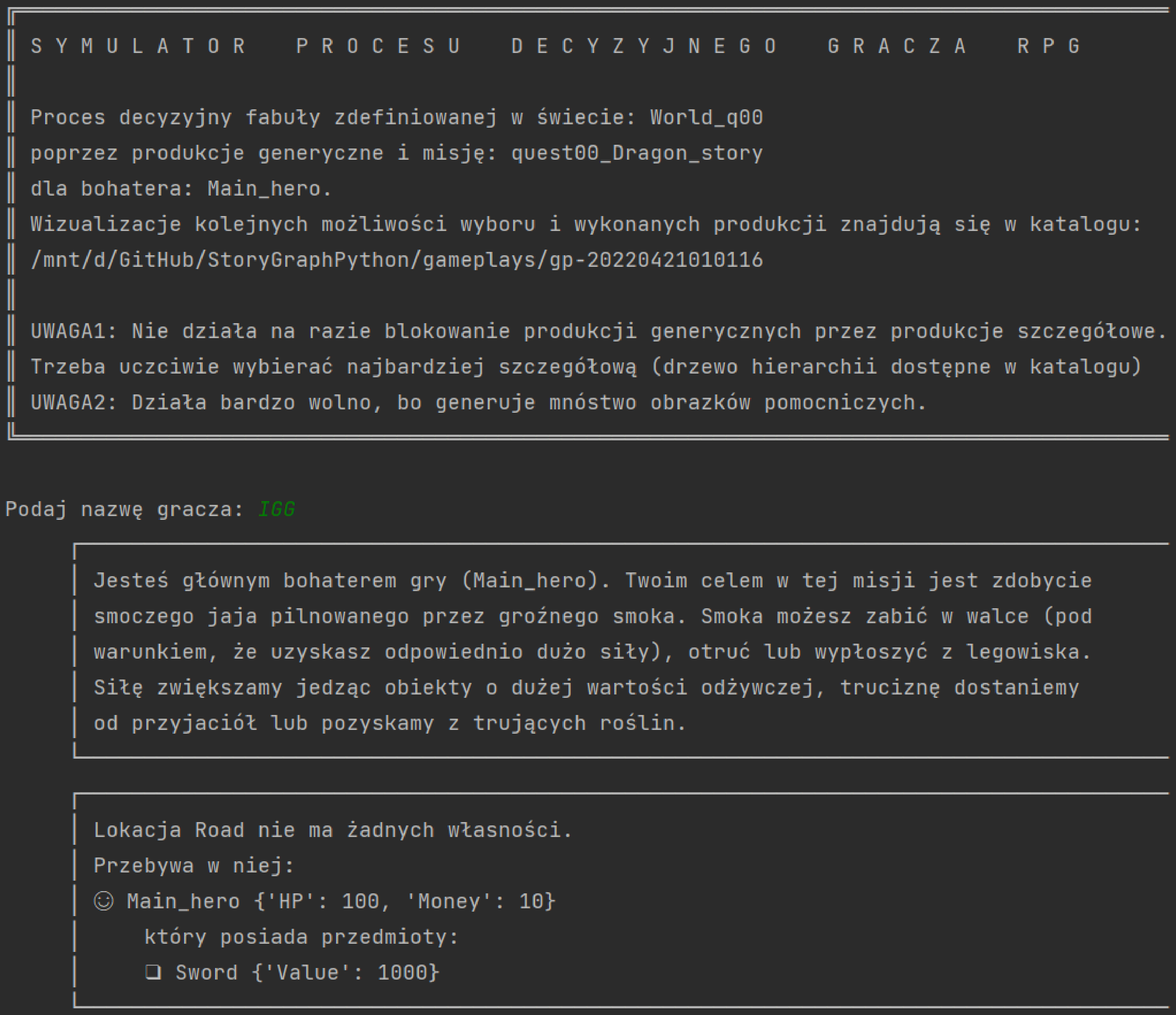
Initial validation is done via JSONSchema https://json-schema.org/.

Graph visualization using the GraphViz 3 library based on the DOT engine.

Visualization of each stage of the game on demand.

Data recording to JSON format at every stage of the application’s operation (state of the world, progress of the game).

https://github.com/iwonagg/StoryGraphPhD/

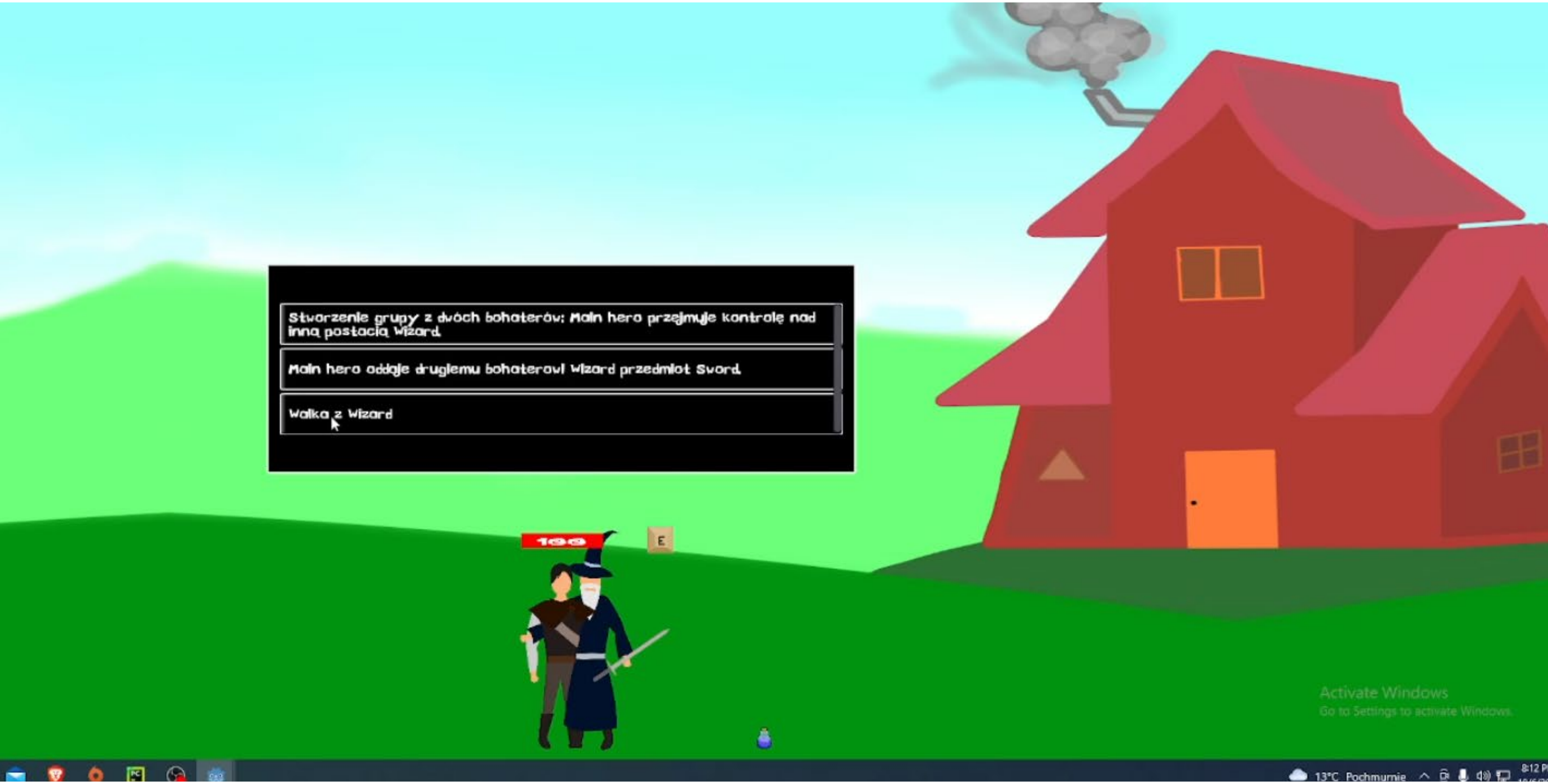


```
╔══════════════════════════════════════════════════════════════════════════════════════════
║ S Y M U L A T O R   P R O C E S U   D E C Y Z Y J N E G O   G R A C Z A   R P G
║
║ Proces decyzyjny fabuły zdefiniowanej w świecie: World_q00
║ poprzez produkcje generyczne i misję: quest00_Dragon_story
║ dla bohatera: Main_hero.
║ Wizualizacje kolejnych możliwości wyboru i wykonanych produkcji znajdują się w katalogu:
║ /mnt/d/GitHub/StoryGraphPython/gameplays/gp-20220421010116
║
║ UWAGA1: Nie działa na razie blokowanie produkcji generycznych przez produkcje szczegółowe.
║ Trzeba uczciwie wybierać najbardziej szczegółową (drzewo hierarchii dostępne w katalogu)
║ UWAGA2: Działa bardzo wolno, bo generuje mnóstwo obrazków pomocniczych.
╚══════════════════════════════════════════════════════════════════════════════════════════

Podaj nazwę gracza: IGG
     ┌─────────────────────────────────────────────────────────────────────────────────────
     │ Jesteś głównym bohaterem gry (Main_hero). Twoim celem w tej misji jest zdobycie
     │ smoczego jaja pilnowanego przez groźnego smoka. Smoka możesz zabić w walce (pod
     │ warunkiem, że uzyskasz odpowiednio dużo siły), otruć lub wypłoszyć z legowiska.
     │ Siłę zwiększamy jedząc obiekty o dużej wartości odżywczej, truciznę dostaniemy
     │ od przyjaciół lub pozyskamy z trujących roślin.
     └─────────────────────────────────────────────────────────────────────────────────────

     ┌─────────────────────────────────────────────────────────────────────────────────────
     │ Lokacja Road nie ma żadnych własności.
     │ Przebywa w niej:
     │ ☺ Main_hero {'HP': 100, 'Money': 10}
     │     który posiada przedmioty:
     │     ❏ Sword {'Value': 1000}
     └─────────────────────────────────────────────────────────────────────────────────────
```

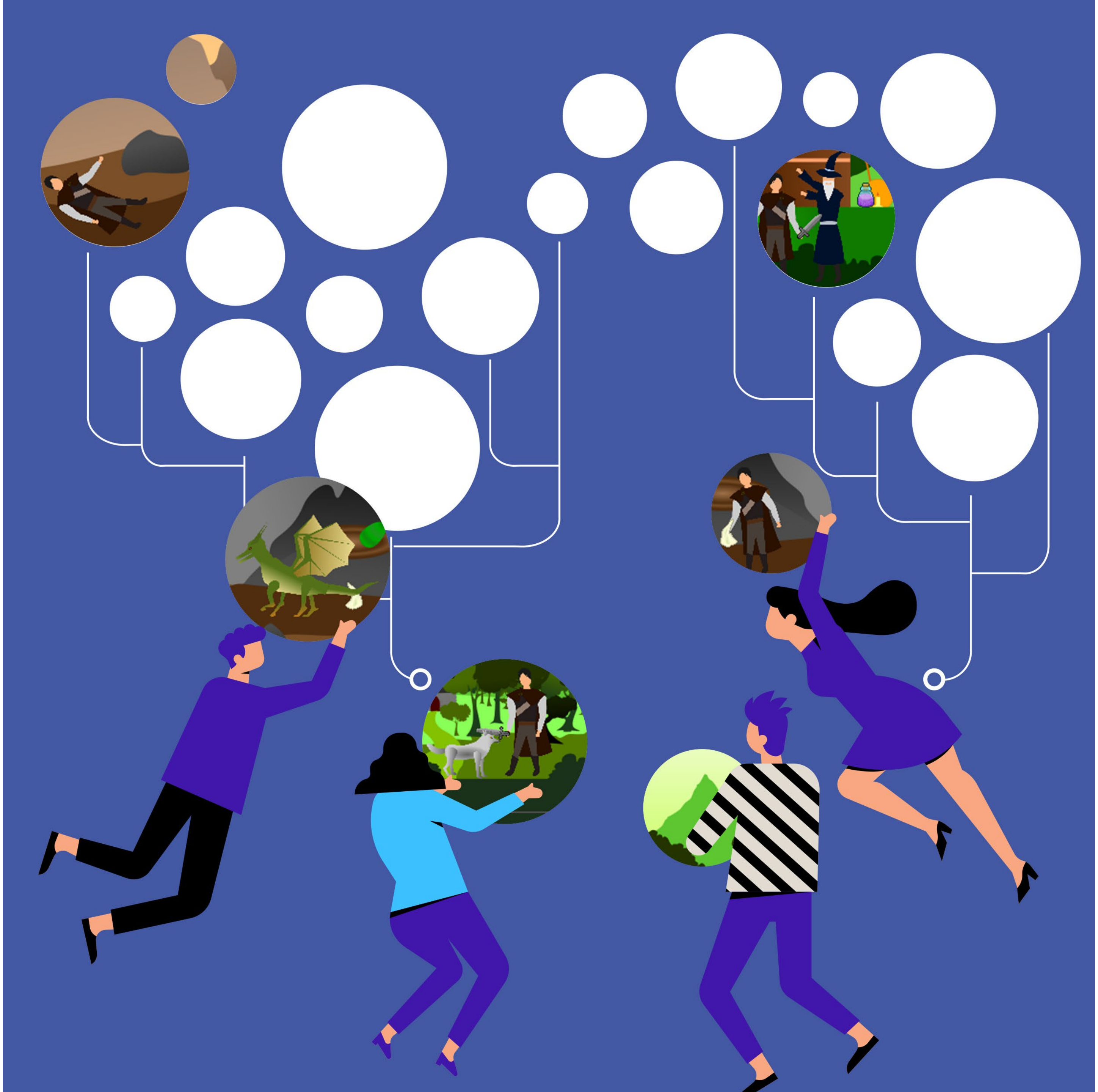
Game based on the system



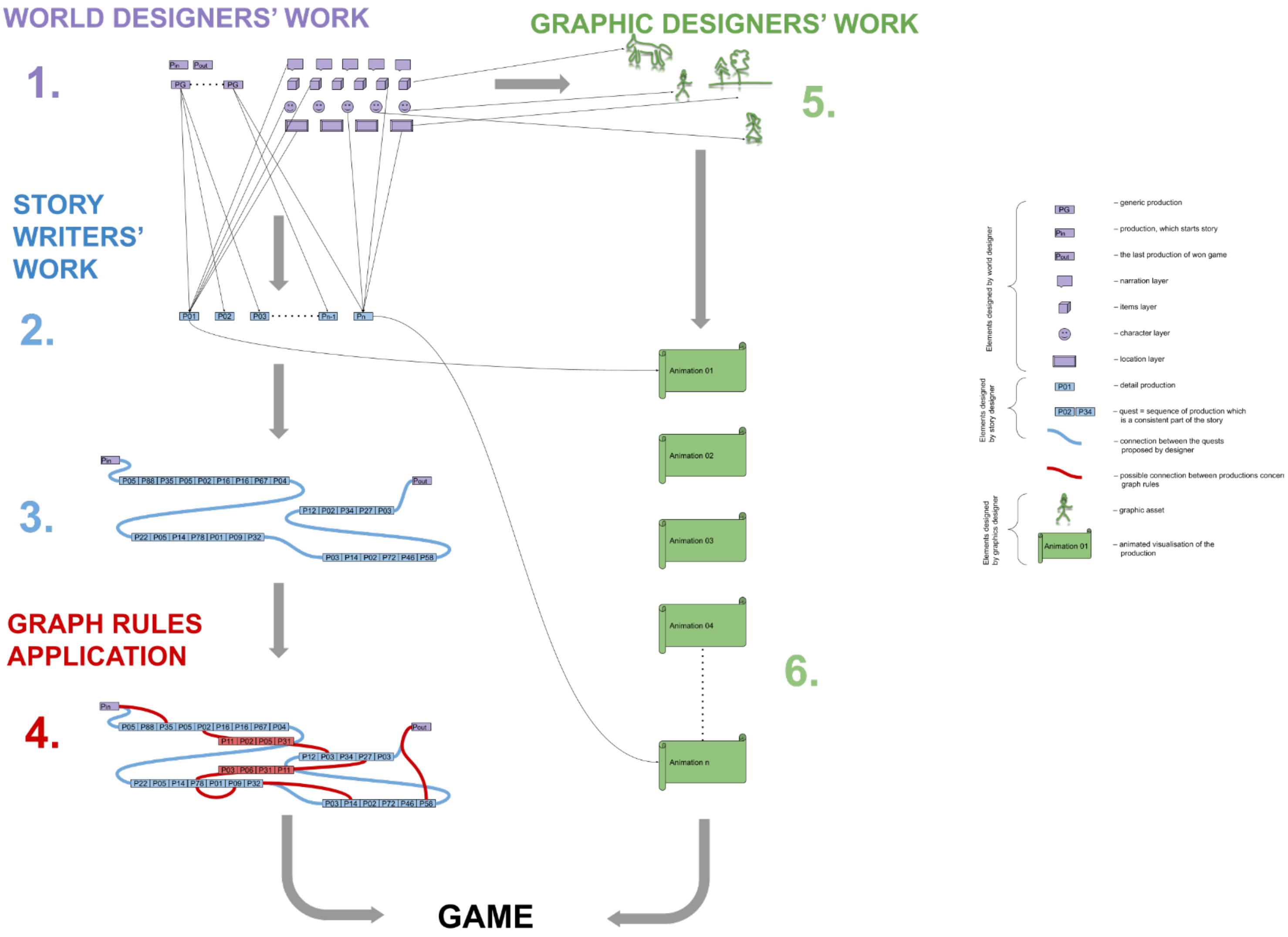
Stworzenie grupy z dwóch bohaterów: Main hero przejmuje kontrolę nad inną postacią Wizard.

Main hero oddaje drugiemu bohaterowi Wizard przedmiot Sword.

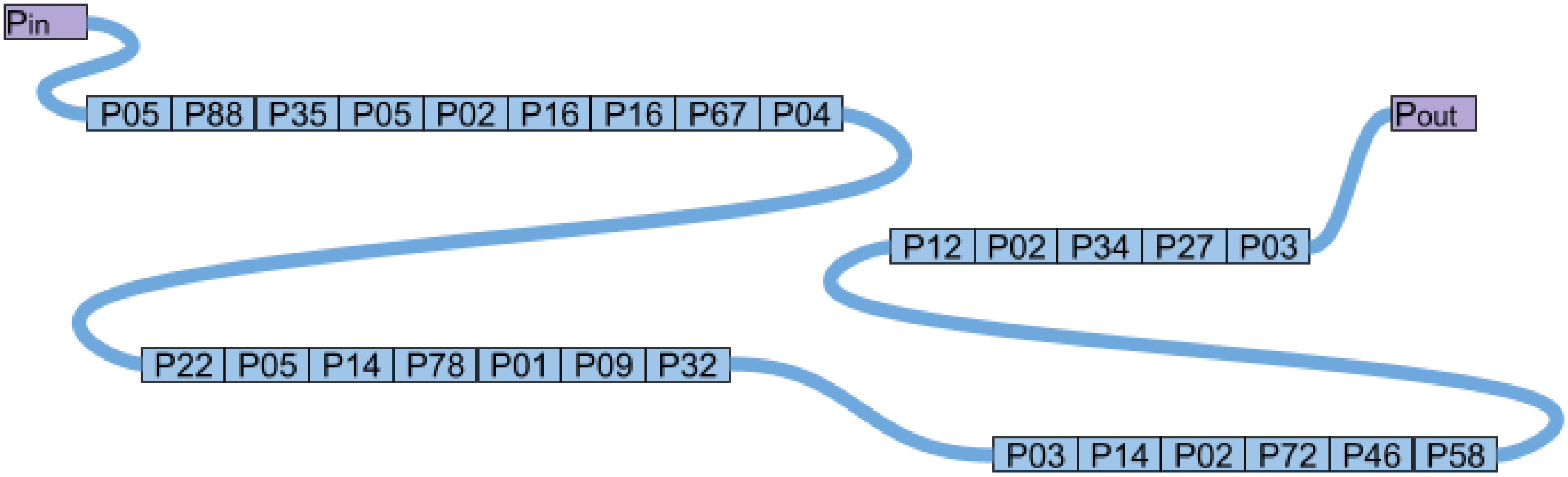
Walka z Wizard



Collaborative game developement


WORLD DESIGNERS' WORK
1.
Pin
Pout
PG
PG
GRAPHIC DESIGNERS' WORK
5.
STORY
WRITERS'
WORK
2.
P01
P02
P03
Pn-1
Pn
Animation 01
Animation 02
Animation 03
Animation 04
Animation n
6.
3.
Pin
P05 | P88 | P35 | P05 | P02 | P16 | P16 | P67 | P04
Pout
P12 | P02 | P34 | P27 | P03
P22 | P05 | P14 | P78 | P01 | P09 | P32
P03 | P14 | P02 | P72 | P46 | P58
GRAPH RULES
APPLICATION
4.
Pin
P05 | P88 | P35 | P05 | P02 | P16 | P16 | P67 | P04
P11 | P02 | P05 | P31
Pout
P12 | P03 | P34 | P27 | P03
P03 | P06 | P31 | P11
P22 | P05 | P14 | P78 | P01 | P09 | P32
P03 | P14 | P02 | P72 | P46 | P58
GAME
Elements designed by world designer
PG
– generic production
Pin
– production, which starts story
Pout
– the last production of won game
– narration layer
– items layer
– character layer
– location layer
Elements designed by story designer
P01
– detail production
P02 | P34
– quest = sequence of production which is a consistent part of the story
– connection between the quests proposed by designer
– possible connection between productions concern graph rules
Elements designed by graphics designer
– graphic asset
Animation 01
– animated visualisation of the production

Collaborative game developement



Pin

P05 P88 P35 P05 P02 P16 P16 P67 P04

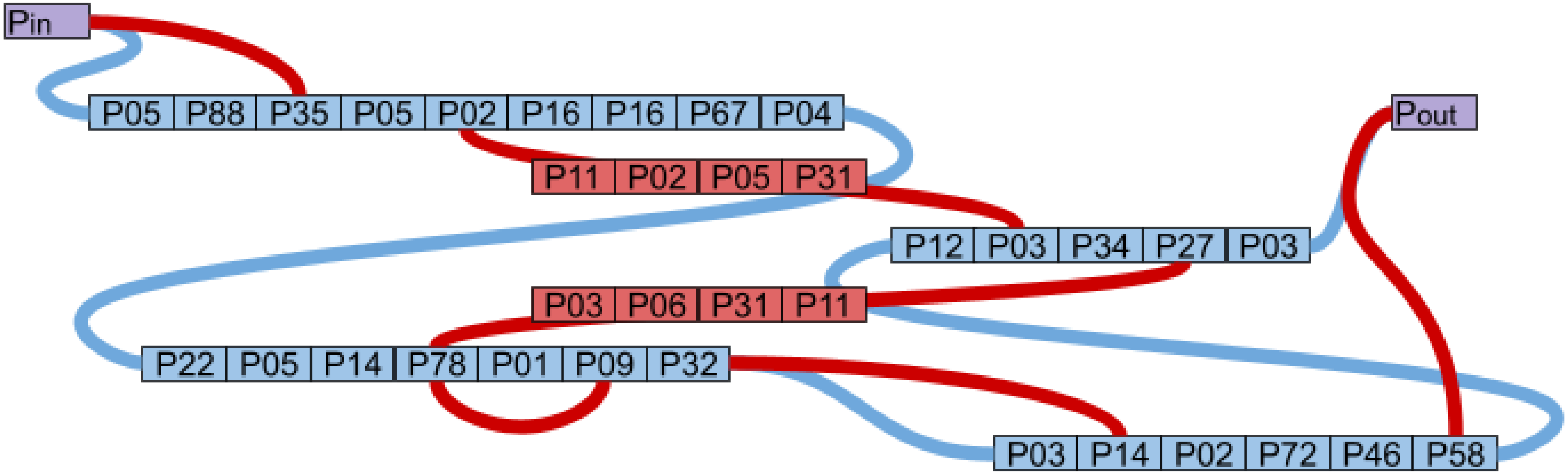
P22 P05 P14 P78 P01 P09 P32

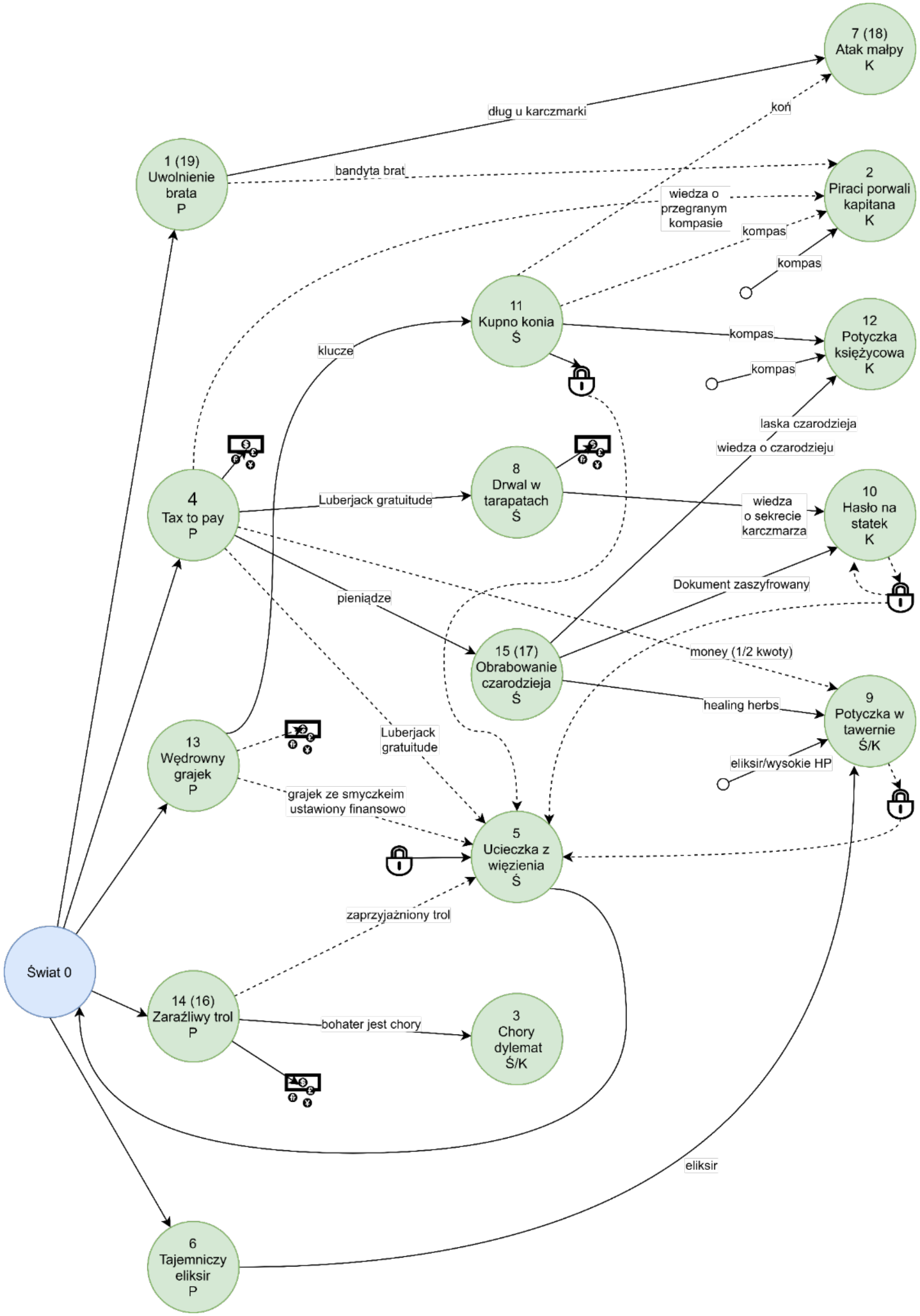
P03 P14 P02 P72 P46 P58

P12 P02 P34 P27 P03

Pout

Collaborative game developement




7 (18)
Atak małpy
K

dług u karczmarki

koń

1 (19)
Uwolnienie brata
P

bandyta brat

2
Piraci porwali kapitana
K

wiedza o przegranym kompasie

kompas

kompas

11
Kupno konia
Ś

kompas

12
Potyczka księżycowa
K

kompas

klucze

laska czarodzieja

wiedza o czarodzieju

8
Drwal w tarapatach
Ś

4
Tax to pay
P

Luberjack gratuitude

wiedza o sekrecie karczmarza

10
Hasło na statek
K

Dokument zaszyfrowany

pieniądze

15 (17)
Obrabowanie czarodzieja
Ś

money (1/2 kwoty)

healing herbs

9
Potyczka w tawernie
Ś/K

13
Wędrowny grajek
P

Luberjack gratuitude

eliksir/wysokie HP

grajek ze smyczkeim ustawiony finansowo

5
Ucieczka z więzienia
Ś

zaprzyjaźniony trol

Świat 0

14 (16)
Zaraźliwy trol
P

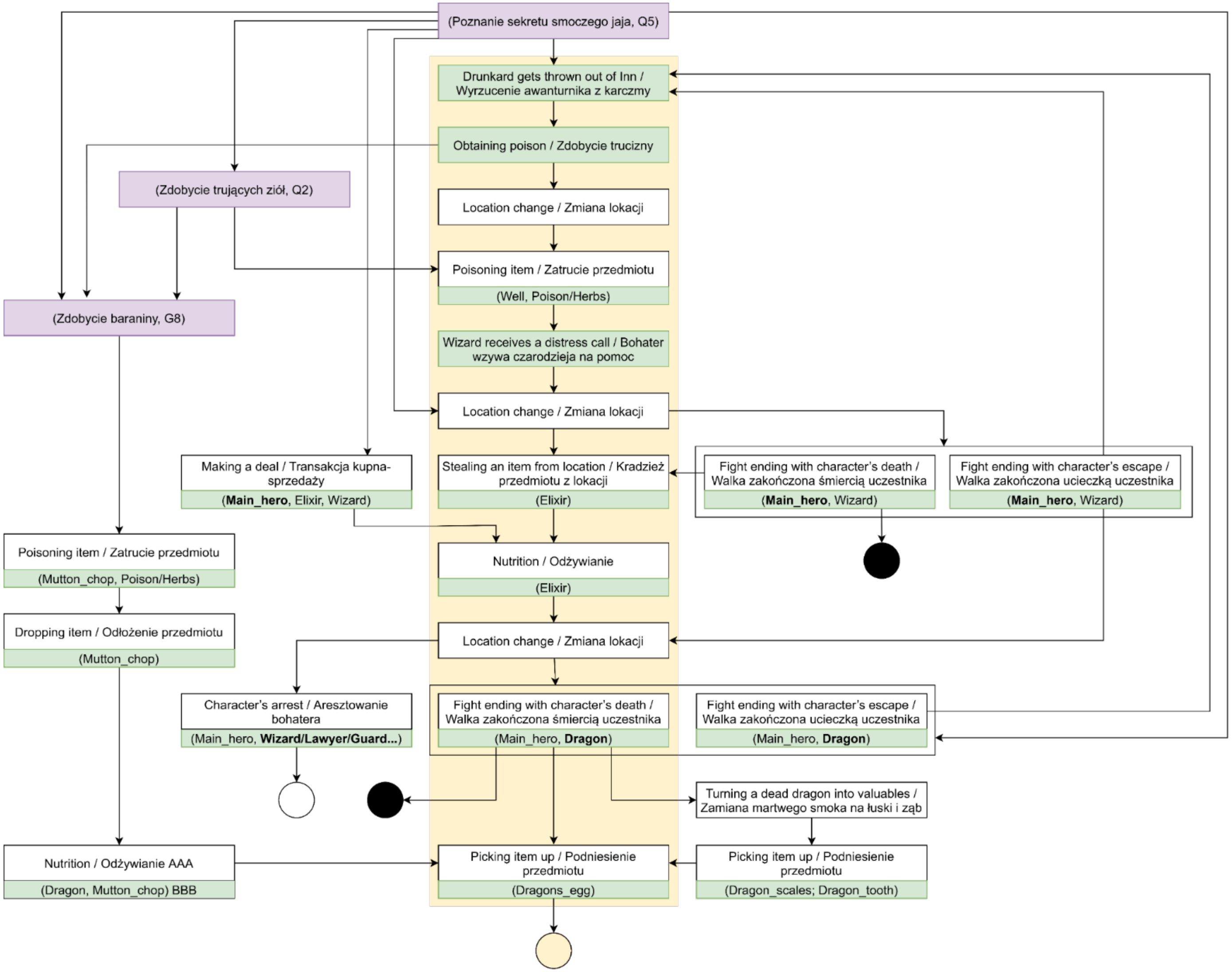
bohater jest chory

3
Chory dylemat
Ś/K

eliksir

6
Tajemniczy eliksir
P

Mission example



(Poznanie sekretu smoczego jaja, Q5)

Drunkard gets thrown out of Inn / Wyrzucenie awanturnika z karczmy

Obtaining poison / Zdobycie trucizny

(Zdobycie trujących ziół, Q2)

Location change / Zmiana lokacji

Poisoning item / Zatrucie przedmiotu

(Well, Poison/Herbs)

(Zdobycie baraniny, G8)

Wizard receives a distress call / Bohater wzywa czarodzieja na pomoc

Location change / Zmiana lokacji

Making a deal / Transakcja kupna-sprzedaży

(**Main_hero**, Elixir, Wizard)

Stealing an item from location / Kradzież przedmiotu z lokacji

(Elixir)

Fight ending with character's death / Walka zakończona śmiercią uczestnika

(**Main_hero**, Wizard)

Fight ending with character's escape / Walka zakończona ucieczką uczestnika

(**Main_hero**, Wizard)

Poisoning item / Zatrucie przedmiotu

(Mutton_chop, Poison/Herbs)

Nutrition / Odżywianie

(Elixir)

Dropping item / Odłożenie przedmiotu

(Mutton_chop)

Location change / Zmiana lokacji

Character's arrest / Aresztowanie bohatera

(Main_hero, **Wizard/Lawyer/Guard...**)

Fight ending with character's death / Walka zakończona śmiercią uczestnika

(Main_hero, **Dragon**)

Fight ending with character's escape / Walka zakończona ucieczką uczestnika

(Main_hero, **Dragon**)

Turning a dead dragon into valuables / Zamiana martwego smoka na łuski i ząb

Nutrition / Odżywianie AAA

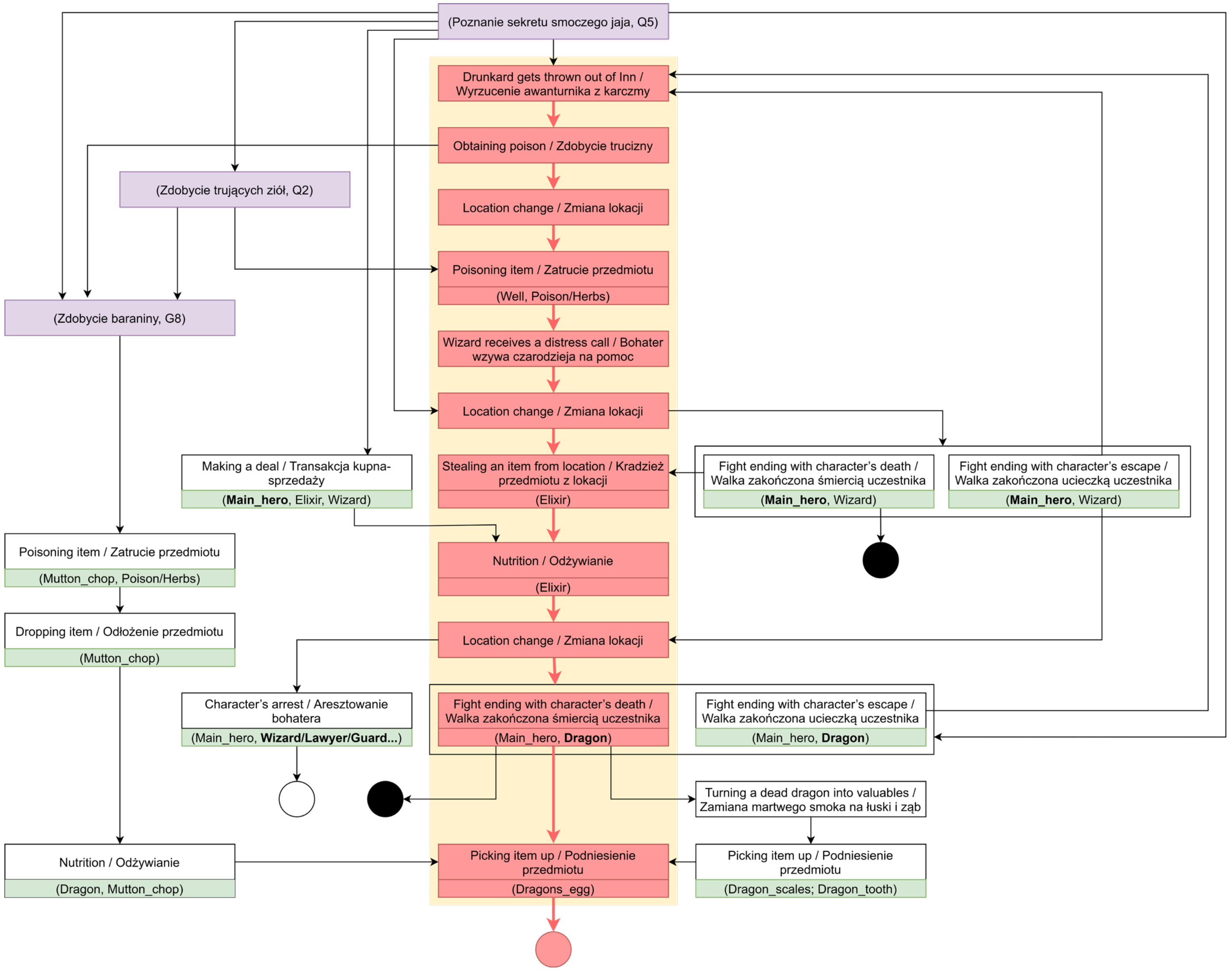
(Dragon, Mutton_chop) BBB

Picking item up / Podniesienie przedmiotu

(Dragons_egg)

Picking item up / Podniesienie przedmiotu

(Dragon_scales; Dragon_tooth)



(Poznanie sekretu smoczego jaja, Q5)

Drunkard gets thrown out of Inn / Wyrzucenie awanturnika z karczmy

Obtaining poison / Zdobycie trucizny

(Zdobycie trujących ziół, Q2)

Location change / Zmiana lokacji

Poisoning item / Zatrucie przedmiotu

(Well, Poison/Herbs)

(Zdobycie baraniny, G8)

Wizard receives a distress call / Bohater wzywa czarodzieja na pomoc

Location change / Zmiana lokacji

Making a deal / Transakcja kupna-sprzedaży

(**Main_hero**, Elixir, Wizard)

Stealing an item from location / Kradzież przedmiotu z lokacji

(Elixir)

Fight ending with character's death / Walka zakończona śmiercią uczestnika

(**Main_hero**, Wizard)

Fight ending with character's escape / Walka zakończona ucieczką uczestnika

(**Main_hero**, Wizard)

Poisoning item / Zatrucie przedmiotu

(Mutton_chop, Poison/Herbs)

Nutrition / Odżywianie

(Elixir)

Dropping item / Odłożenie przedmiotu

(Mutton_chop)

Location change / Zmiana lokacji

Character's arrest / Aresztowanie bohatera

(Main_hero, **Wizard/Lawyer/Guard...**)

Fight ending with character's death / Walka zakończona śmiercią uczestnika

(Main_hero, **Dragon**)

Fight ending with character's escape / Walka zakończona ucieczką uczestnika

(Main_hero, **Dragon**)

Turning a dead dragon into valuables / Zamiana martwego smoka na łuski i ząb

Nutrition / Odżywianie

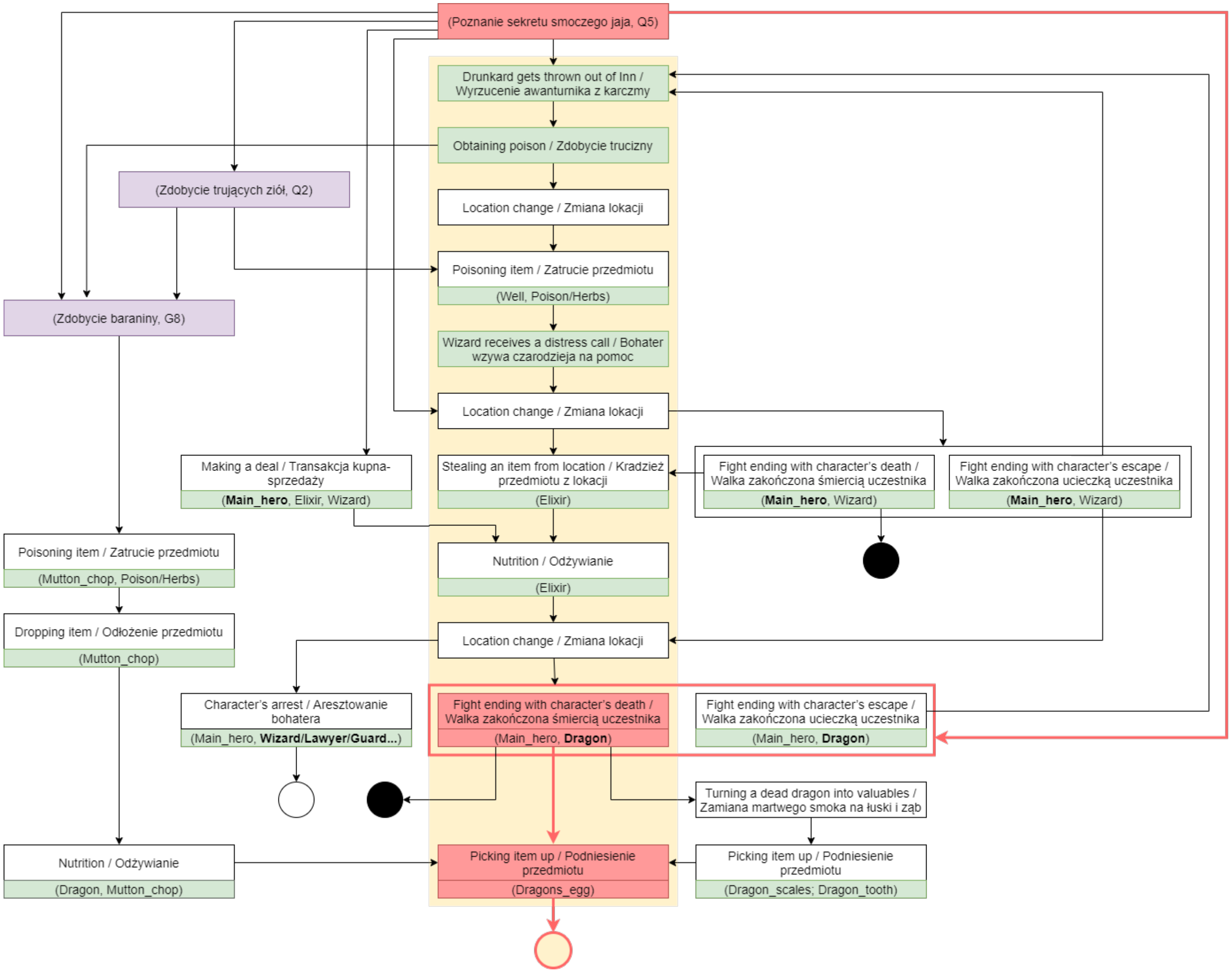
(Dragon, Mutton_chop)

Picking item up / Podniesienie przedmiotu

(Dragons_egg)

Picking item up / Podniesienie przedmiotu

(Dragon_scales; Dragon_tooth)



(Poznanie sekretu smoczego jaja, Q5)

Drunkard gets thrown out of Inn / Wyrzucenie awanturnika z karczmy

Obtaining poison / Zdobycie trucizny

(Zdobycie trujących ziół, Q2)

Location change / Zmiana lokacji

Poisoning item / Zatrucie przedmiotu

(Well, Poison/Herbs)

(Zdobycie baraniny, G8)

Wizard receives a distress call / Bohater wzywa czarodzieja na pomoc

Location change / Zmiana lokacji

Making a deal / Transakcja kupna-sprzedaży

(**Main_hero**, Elixir, Wizard)

Stealing an item from location / Kradzież przedmiotu z lokacji

(Elixir)

Fight ending with character's death / Walka zakończona śmiercią uczestnika

(**Main_hero**, Wizard)

Fight ending with character's escape / Walka zakończona ucieczką uczestnika

(**Main_hero**, Wizard)

Poisoning item / Zatrucie przedmiotu

(Mutton_chop, Poison/Herbs)

Nutrition / Odżywianie

(Elixir)

Dropping item / Odłożenie przedmiotu

(Mutton_chop)

Location change / Zmiana lokacji

Character's arrest / Aresztowanie bohatera

(Main_hero, **Wizard/Lawyer/Guard...**)

Fight ending with character's death / Walka zakończona śmiercią uczestnika

(Main_hero, **Dragon**)

Fight ending with character's escape / Walka zakończona ucieczką uczestnika

(Main_hero, **Dragon**)

Turning a dead dragon into valuables / Zamiana martwego smoka na łuski i ząb

Nutrition / Odżywianie

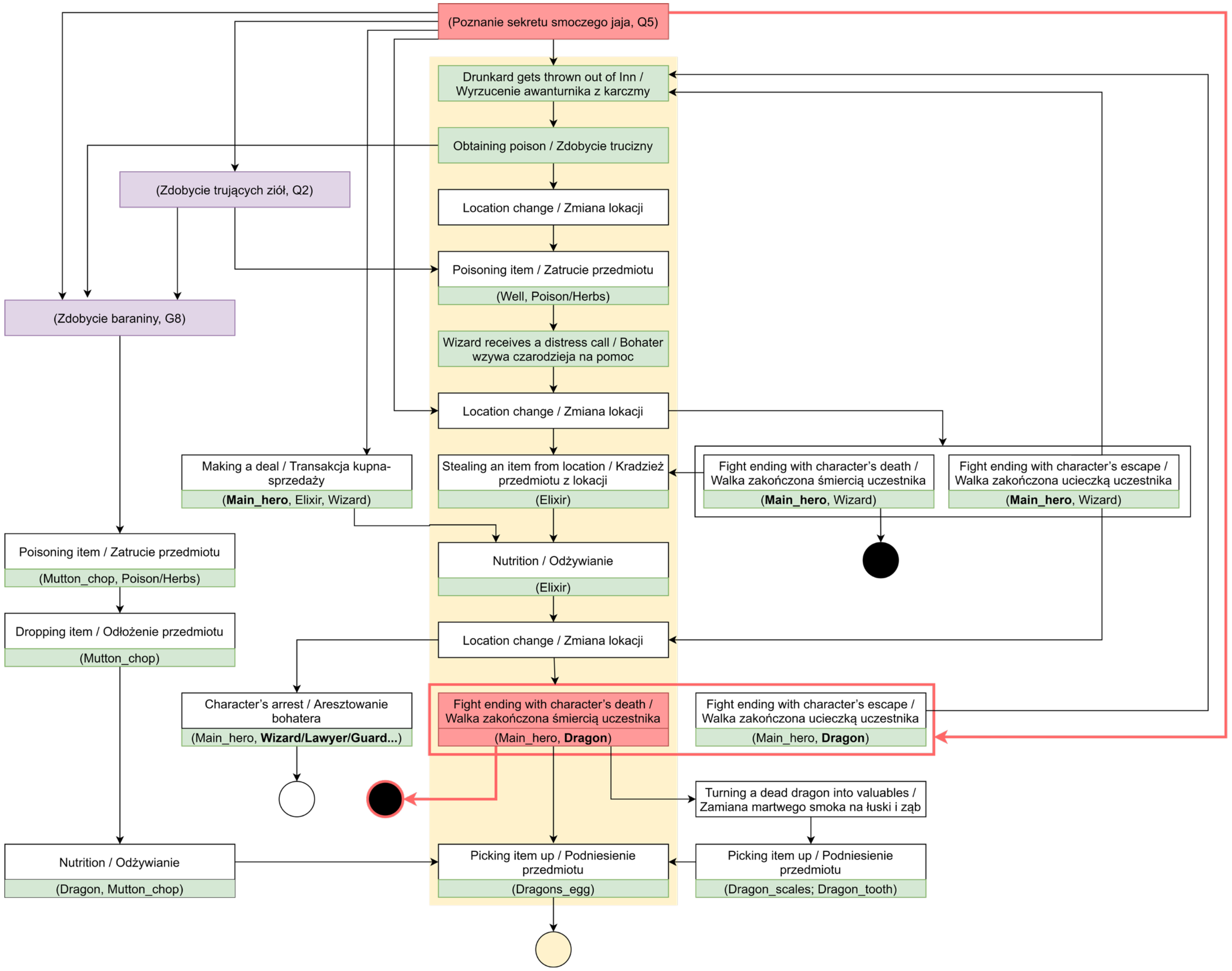
(Dragon, Mutton_chop)

Picking item up / Podniesienie przedmiotu

(Dragons_egg)

Picking item up / Podniesienie przedmiotu

(Dragon_scales; Dragon_tooth)



(Poznanie sekretu smoczego jaja, Q5)

Drunkard gets thrown out of Inn / Wyrzucenie awanturnika z karczmy

Obtaining poison / Zdobycie trucizny

(Zdobycie trujących ziół, Q2)

Location change / Zmiana lokacji

Poisoning item / Zatrucie przedmiotu
(Well, Poison/Herbs)

(Zdobycie baraniny, G8)

Wizard receives a distress call / Bohater wzywa czarodzieja na pomoc

Location change / Zmiana lokacji

Making a deal / Transakcja kupna-sprzedaży
(**Main_hero**, Elixir, Wizard)

Stealing an item from location / Kradzież przedmiotu z lokacji
(Elixir)

Fight ending with character's death / Walka zakończona śmiercią uczestnika
(**Main_hero**, Wizard)

Fight ending with character's escape / Walka zakończona ucieczką uczestnika
(**Main_hero**, Wizard)

Poisoning item / Zatrucie przedmiotu
(Mutton_chop, Poison/Herbs)

Nutrition / Odżywianie
(Elixir)

Dropping item / Odłożenie przedmiotu
(Mutton_chop)

Location change / Zmiana lokacji

Character's arrest / Aresztowanie bohatera
(Main_hero, **Wizard/Lawyer/Guard...**)

Fight ending with character's death / Walka zakończona śmiercią uczestnika
(Main_hero, **Dragon**)

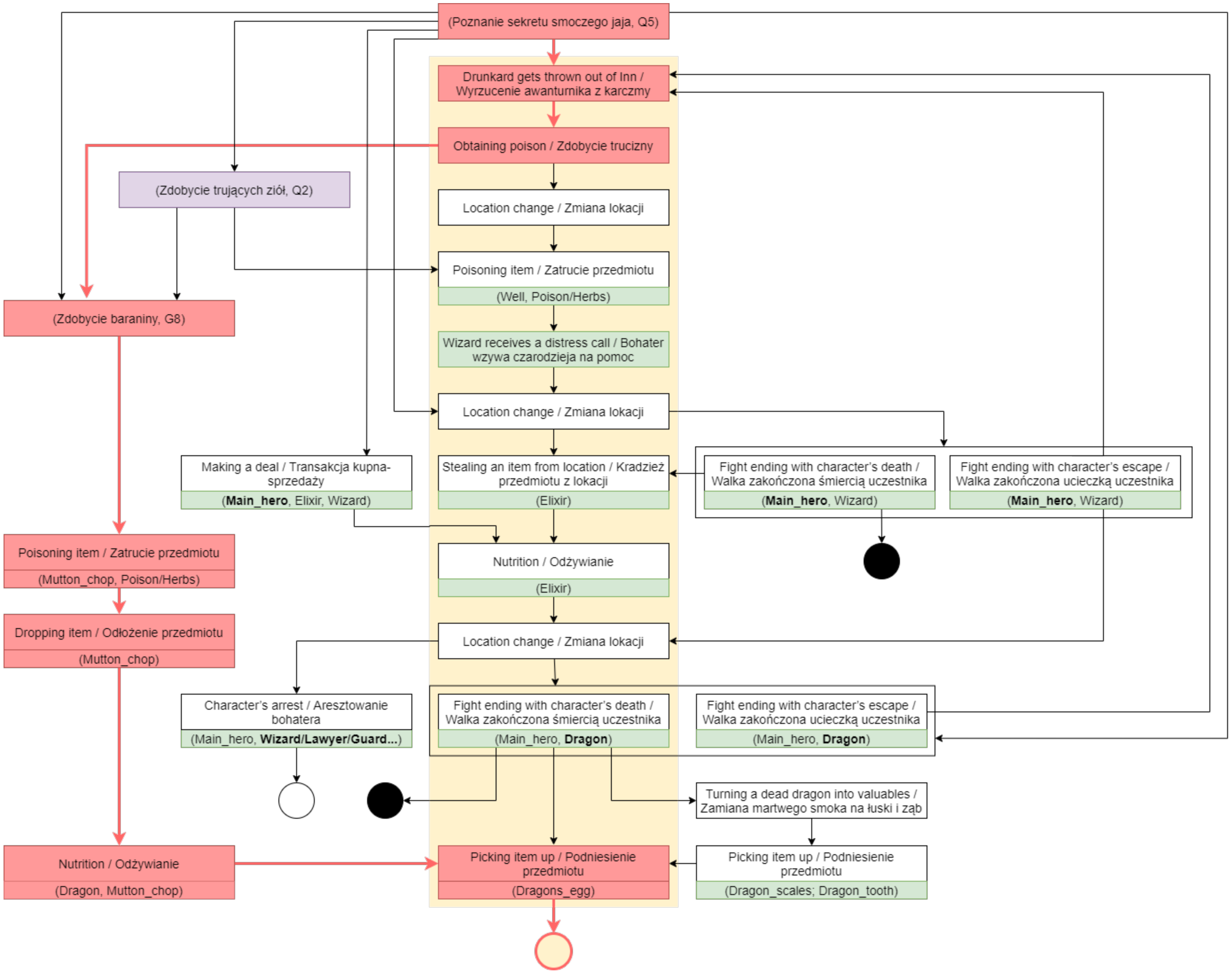
Fight ending with character's escape / Walka zakończona ucieczką uczestnika
(Main_hero, **Dragon**)

Turning a dead dragon into valuables / Zamiana martwego smoka na łuski i ząb

Nutrition / Odżywianie
(Dragon, Mutton_chop)

Picking item up / Podniesienie przedmiotu
(Dragons_egg)

Picking item up / Podniesienie przedmiotu
(Dragon_scales; Dragon_tooth)



(Poznanie sekretu smoczego jaja, Q5)

Drunkard gets thrown out of Inn / Wyrzucenie awanturnika z karczmy

Obtaining poison / Zdobycie trucizny

(Zdobycie trujących ziół, Q2)

Location change / Zmiana lokacji

Poisoning item / Zatrucie przedmiotu

(Well, Poison/Herbs)

(Zdobycie baraniny, G8)

Wizard receives a distress call / Bohater wzywa czarodzieja na pomoc

Location change / Zmiana lokacji

Making a deal / Transakcja kupna-sprzedaży

(**Main_hero**, Elixir, Wizard)

Stealing an item from location / Kradzież przedmiotu z lokacji

(Elixir)

Fight ending with character's death / Walka zakończona śmiercią uczestnika

(**Main_hero**, Wizard)

Fight ending with character's escape / Walka zakończona ucieczką uczestnika

(**Main_hero**, Wizard)

Poisoning item / Zatrucie przedmiotu

(Mutton_chop, Poison/Herbs)

Nutrition / Odżywianie

(Elixir)

Dropping item / Odłożenie przedmiotu

(Mutton_chop)

Location change / Zmiana lokacji

Character's arrest / Aresztowanie bohatera

(Main_hero, **Wizard/Lawyer/Guard…**)

Fight ending with character's death / Walka zakończona śmiercią uczestnika

(Main_hero, **Dragon**)

Fight ending with character's escape / Walka zakończona ucieczką uczestnika

(Main_hero, **Dragon**)

Turning a dead dragon into valuables / Zamiana martwego smoka na łuski i ząb

Nutrition / Odżywianie

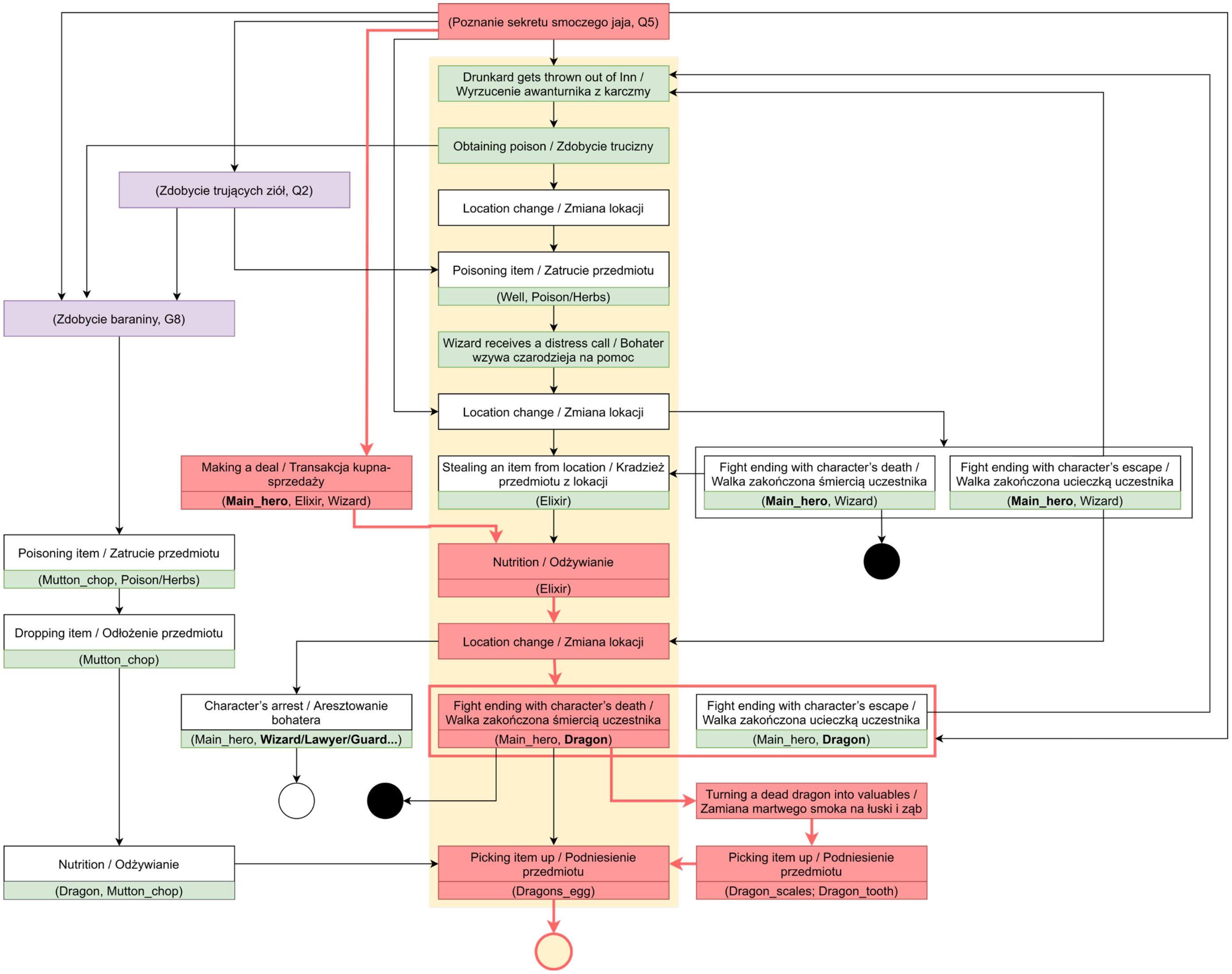
(Dragon, Mutton_chop)

Picking item up / Podniesienie przedmiotu

(Dragons_egg)

Picking item up / Podniesienie przedmiotu

(Dragon_scales; Dragon_tooth)



(Poznanie sekretu smoczego jaja, Q5)

Drunkard gets thrown out of Inn / Wyrzucenie awanturnika z karczmy

Obtaining poison / Zdobycie trucizny

(Zdobycie trujących ziół, Q2)

Location change / Zmiana lokacji

Poisoning item / Zatrucie przedmiotu

(Well, Poison/Herbs)

(Zdobycie baraniny, G8)

Wizard receives a distress call / Bohater wzywa czarodzieja na pomoc

Location change / Zmiana lokacji

Making a deal / Transakcja kupna-sprzedaży

(**Main_hero**, Elixir, Wizard)

Stealing an item from location / Kradzież przedmiotu z lokacji

(Elixir)

Fight ending with character's death / Walka zakończona śmiercią uczestnika

(**Main_hero**, Wizard)

Fight ending with character's escape / Walka zakończona ucieczką uczestnika

(**Main_hero**, Wizard)

Poisoning item / Zatrucie przedmiotu

(Mutton_chop, Poison/Herbs)

Nutrition / Odżywianie

(Elixir)

Dropping item / Odłożenie przedmiotu

(Mutton_chop)

Location change / Zmiana lokacji

Character's arrest / Aresztowanie bohatera

(Main_hero, **Wizard/Lawyer/Guard...**)

Fight ending with character's death / Walka zakończona śmiercią uczestnika

(Main_hero, **Dragon**)

Fight ending with character's escape / Walka zakończona ucieczką uczestnika

(Main_hero, **Dragon**)

Turning a dead dragon into valuables / Zamiana martwego smoka na łuski i ząb

Nutrition / Odżywianie

(Dragon, Mutton_chop)

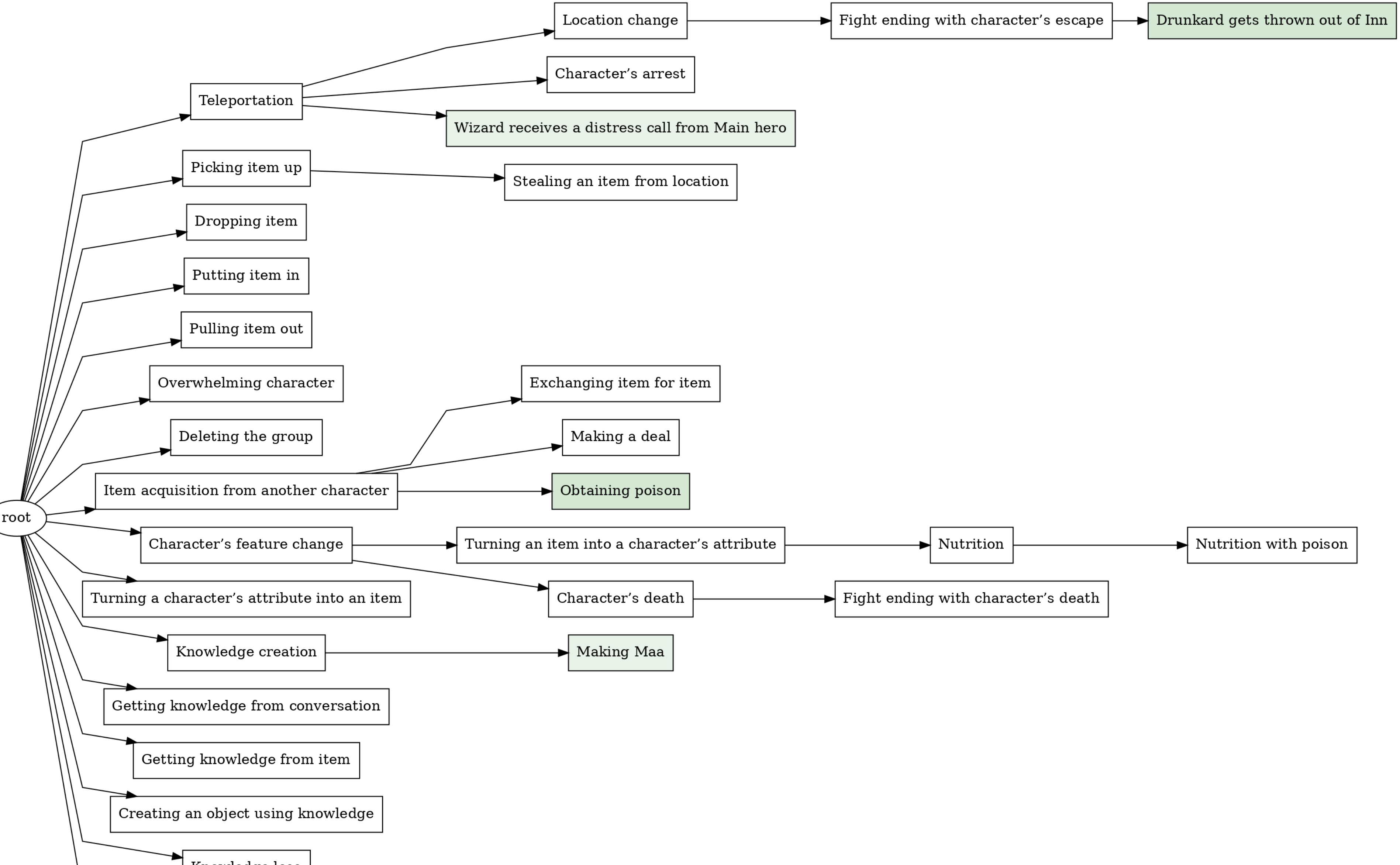
Picking item up / Podniesienie przedmiotu

(Dragons_egg)

Picking item up / Podniesienie przedmiotu

(Dragon_scales; Dragon_tooth)

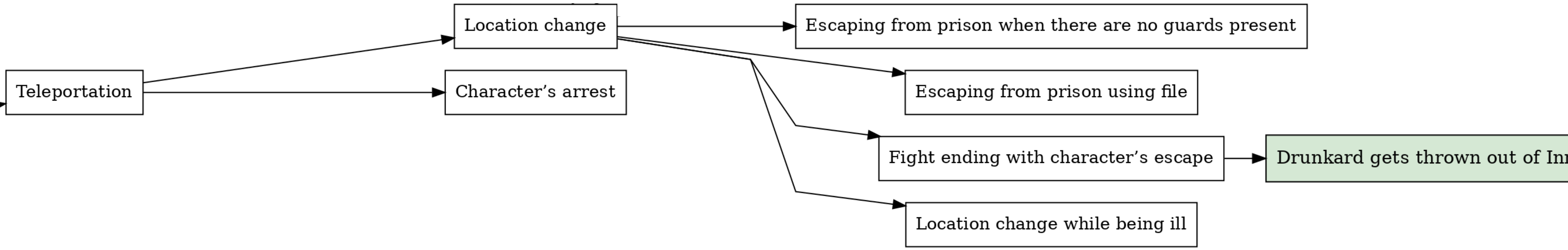
Production hierarchy tree



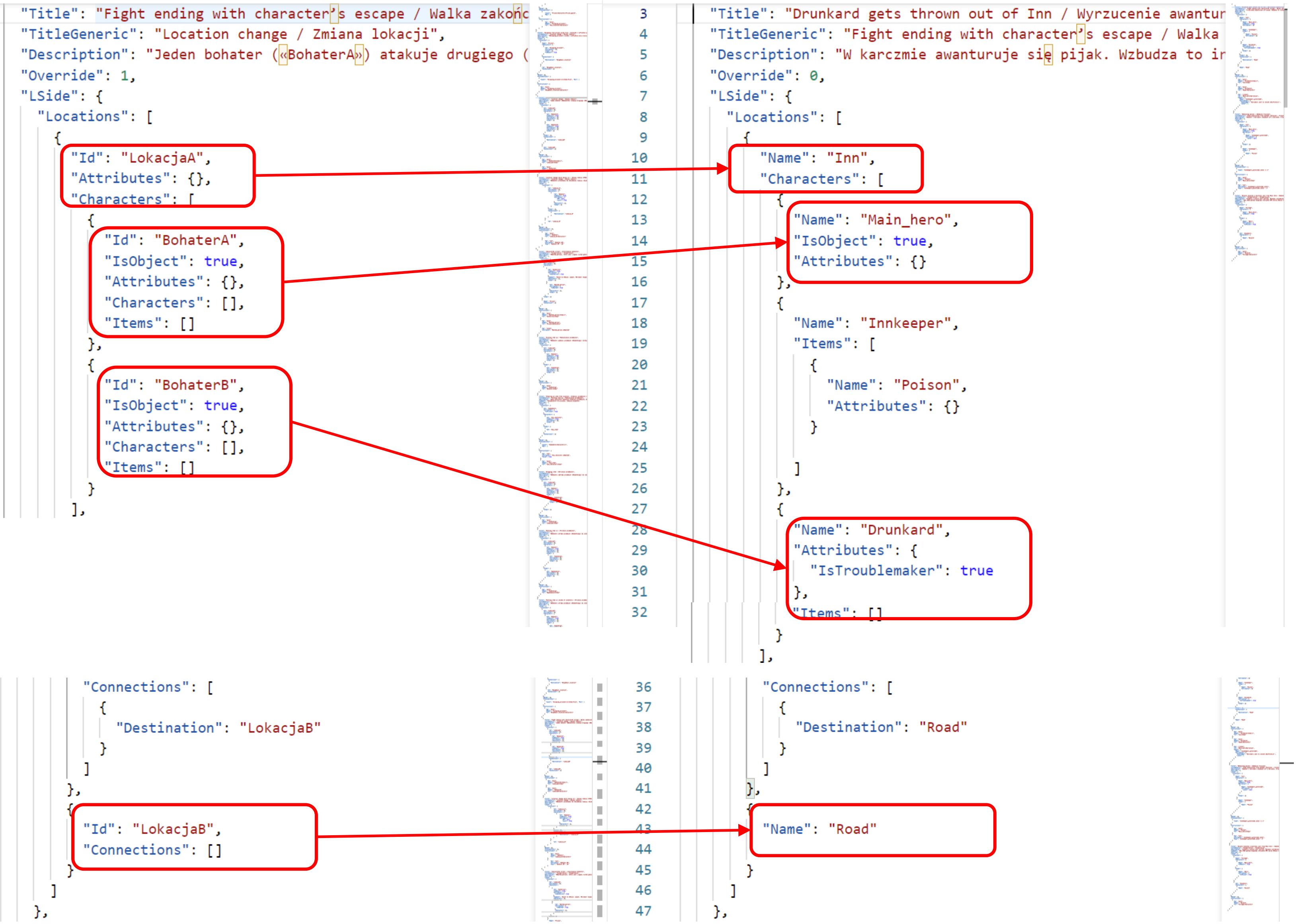
- root
  - Teleportation
    - Location change
      - Fight ending with character’s escape
        - Drunkard gets thrown out of Inn
    - Character’s arrest
    - Wizard receives a distress call from Main hero
  - Picking item up
    - Stealing an item from location
  - Dropping item
  - Putting item in
  - Pulling item out
  - Overwhelming character
  - Deleting the group
  - Item acquisition from another character
    - Exchanging item for item
    - Making a deal
    - Obtaining poison
  - Character’s feature change
    - Turning an item into a character’s attribute
      - Nutrition
        - Nutrition with poison
    - Character’s death
      - Fight ending with character’s death
  - Turning a character’s attribute into an item
  - Knowledge creation
    - Making Maa
  - Getting knowledge from conversation
  - Getting knowledge from item
  - Creating an object using knowledge
  - Knowledge loss

Production hierarchy tree

- The goal is to unify the game mechanics.


Location change
Escaping from prison when there are no guards present
Teleportation
Character’s arrest
Escaping from prison using file
Fight ending with character’s escape
Drunkard gets thrown out of Inn
Location change while being ill

Hierarchy of productions



```
"Title": "Fight ending with character’s escape / Walka zakońc
"TitleGeneric": "Location change / Zmiana lokacji",
"Description": "Jeden bohater («BohaterA») atakuje drugiego (
"Override": 1,
"LSide": {
  "Locations": [
    {
      "Id": "LokacjaA",
      "Attributes": {},
      "Characters": [
        {
          "Id": "BohaterA",
          "IsObject": true,
          "Attributes": {},
          "Characters": [],
          "Items": []
        },
        {
          "Id": "BohaterB",
          "IsObject": true,
          "Attributes": {},
          "Characters": [],
          "Items": []
        }
      ],
```

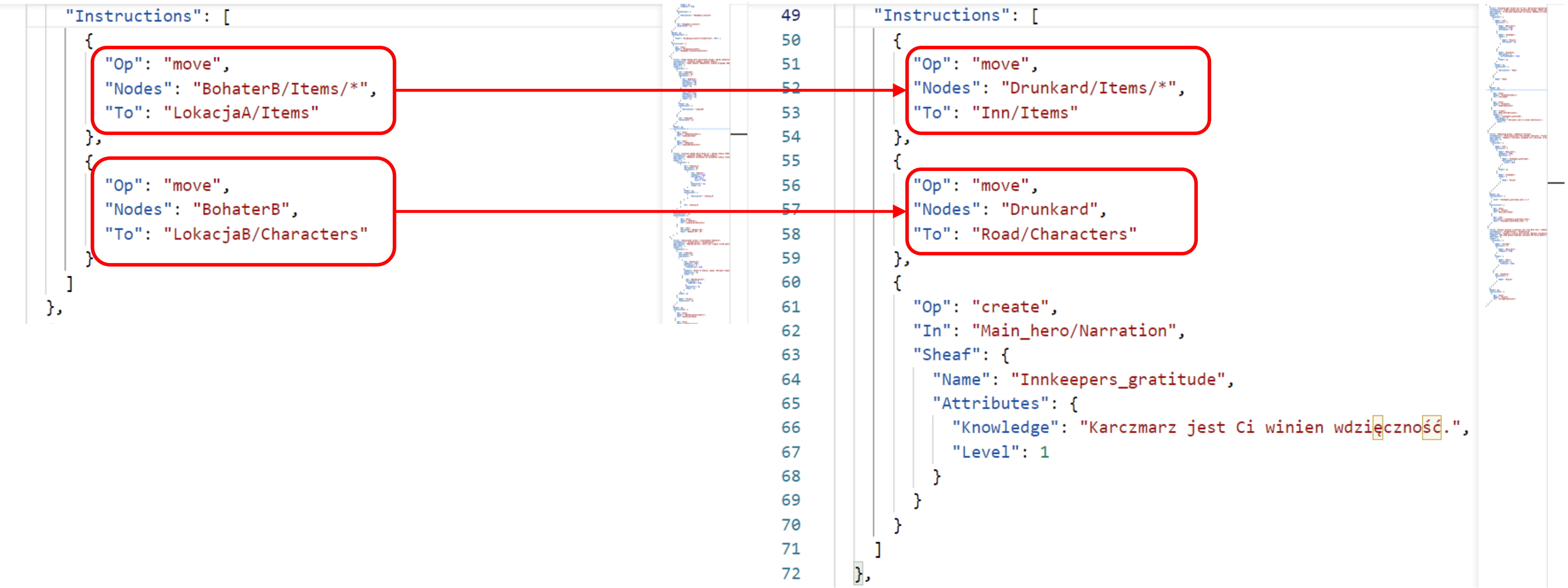
```
      "Connections": [
        {
          "Destination": "LokacjaB"
        }
      ]
    },
    {
      "Id": "LokacjaB",
      "Connections": []
    }
  ]
},
```

```
"Title": "Drunkard gets thrown out of Inn / Wyrzucenie awantur
"TitleGeneric": "Fight ending with character’s escape / Walka
"Description": "W karczmie awanturuje się pijak. Wzbudza to ir
"Override": 0,
"LSide": {
  "Locations": [
    {
      "Name": "Inn",
      "Characters": [
        {
          "Name": "Main_hero",
          "IsObject": true,
          "Attributes": {}
        },
        {
          "Name": "Innkeeper",
          "Items": [
            {
              "Name": "Poison",
              "Attributes": {}
            }

          ]
        },
        {
          "Name": "Drunkard",
          "Attributes": {
            "IsTroublemaker": true
          },
          "Items": []
        }
      ],
```

```
      "Connections": [
        {
          "Destination": "Road"
        }
      ]
    },
    {
      "Name": "Road"
    }
  ]
},
```

Hierarchy of productions



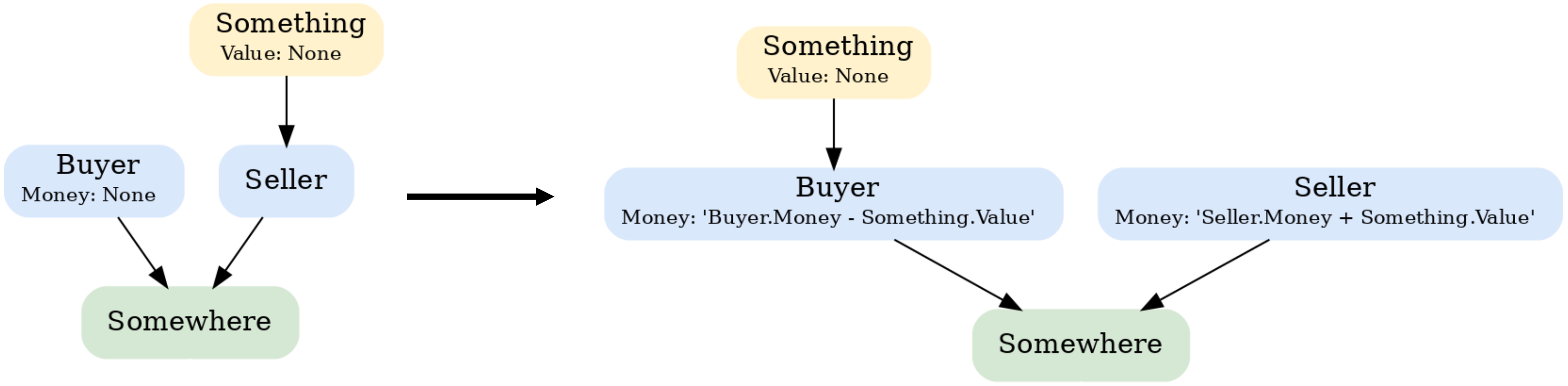
```
  "Instructions": [
    {
      "Op": "move",
      "Nodes": "BohaterB/Items/*",
      "To": "LokacjaA/Items"
    },
    {
      "Op": "move",
      "Nodes": "BohaterB",
      "To": "LokacjaB/Characters"
    }
  ]
},
```

```
  "Instructions": [
    {
      "Op": "move",
      "Nodes": "Drunkard/Items/*",
      "To": "Inn/Items"
    },
    {
      "Op": "move",
      "Nodes": "Drunkard",
      "To": "Road/Characters"
    },
    {
      "Op": "create",
      "In": "Main_hero/Narration",
      "Sheaf": {
        "Name": "Innkeepers_gratitude",
        "Attributes": {
          "Knowledge": "Karczmarz jest Ci winien wdzięczność.",
          "Level": 1
        }
      }
    }
  ]
},
```

Visualisation of productions

Making a deal

«Seller» sprzedaje kupującemu («Buyer») przedmiot («Something»).



Something
Value: None

Buyer
Money: None

Seller

Somewhere

Something
Value: None

Buyer
Money: 'Buyer.Money - Something.Value'

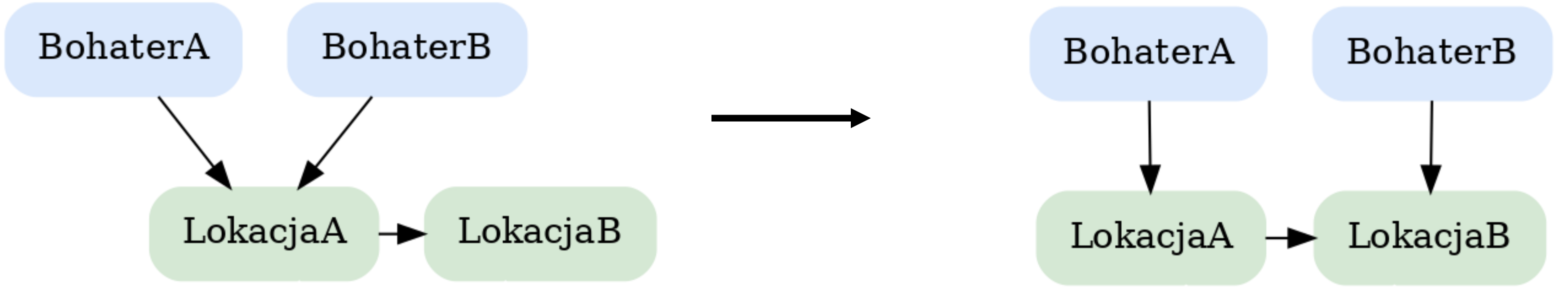
Seller
Money: 'Seller.Money + Something.Value'

Somewhere

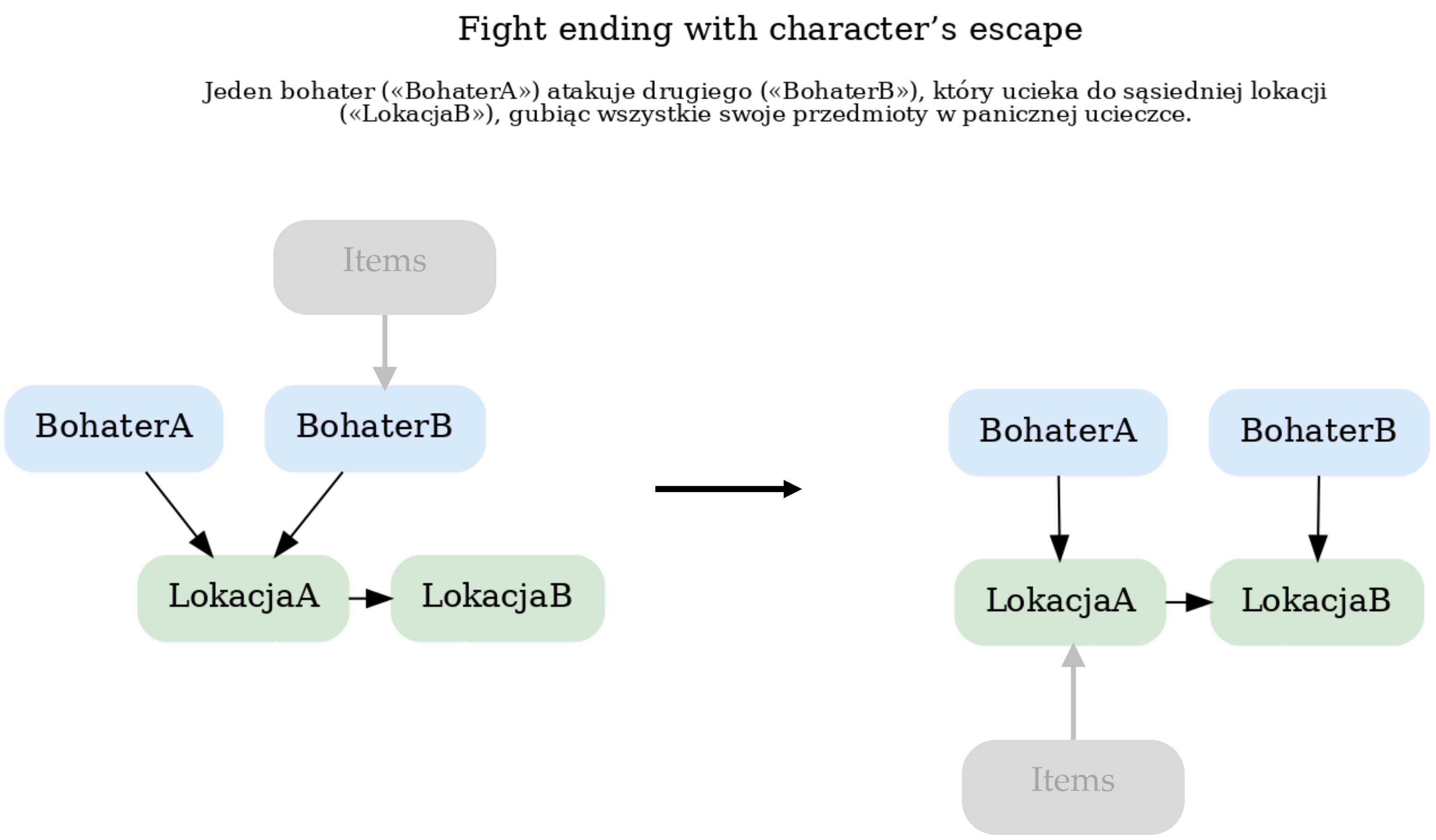
Visualisation of productions

Fight ending with character’s escape

Jeden bohater («BohaterA») atakuje drugiego («BohaterB»), który ucieka do sąsiedniej lokacji («LokacjaB»),



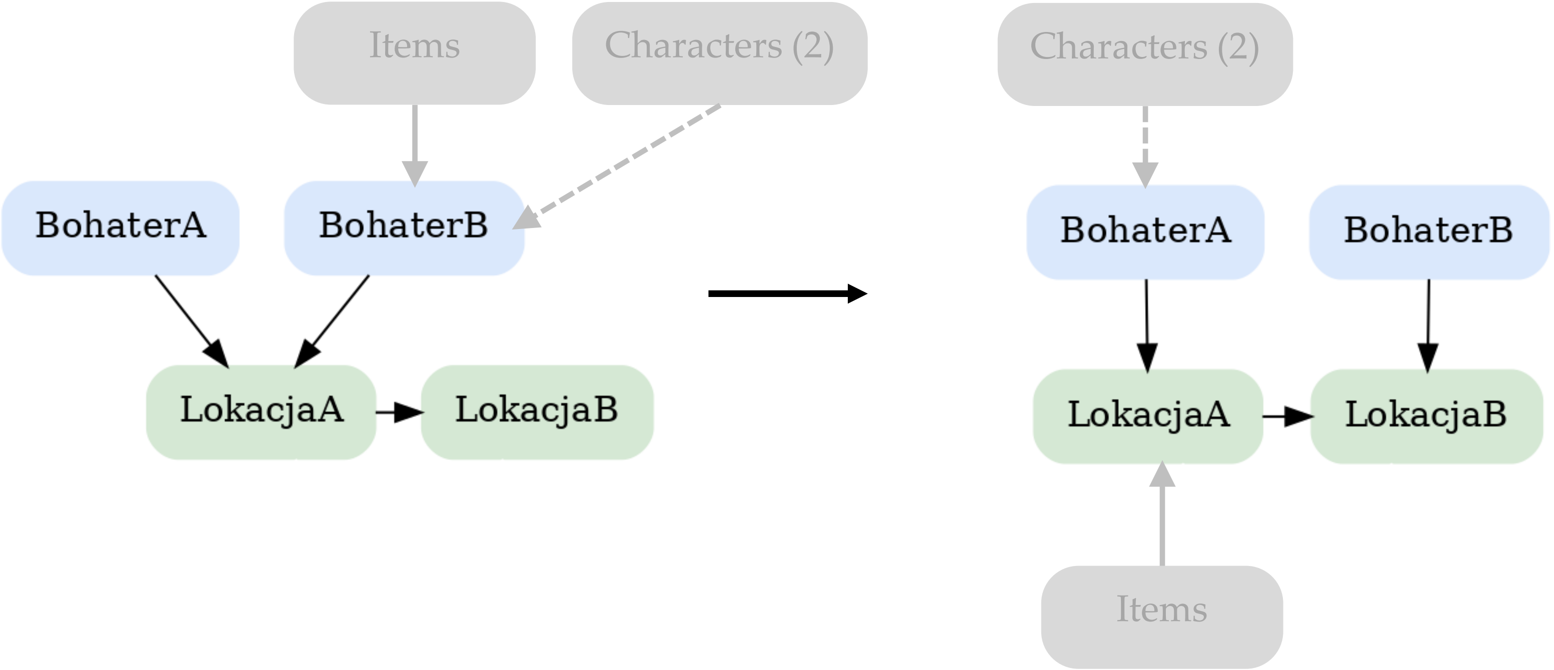
Visualisation of productions


Fight ending with character's escape
Jeden bohater («BohaterA») atakuje drugiego («BohaterB»), który ucieka do sąsiedniej lokacji («LokacjaB»), gubiąc wszystkie swoje przedmioty w panicznej ucieczce.
Items
BohaterA
BohaterB
LokacjaA
LokacjaB
BohaterA
BohaterB
LokacjaA
LokacjaB
Items

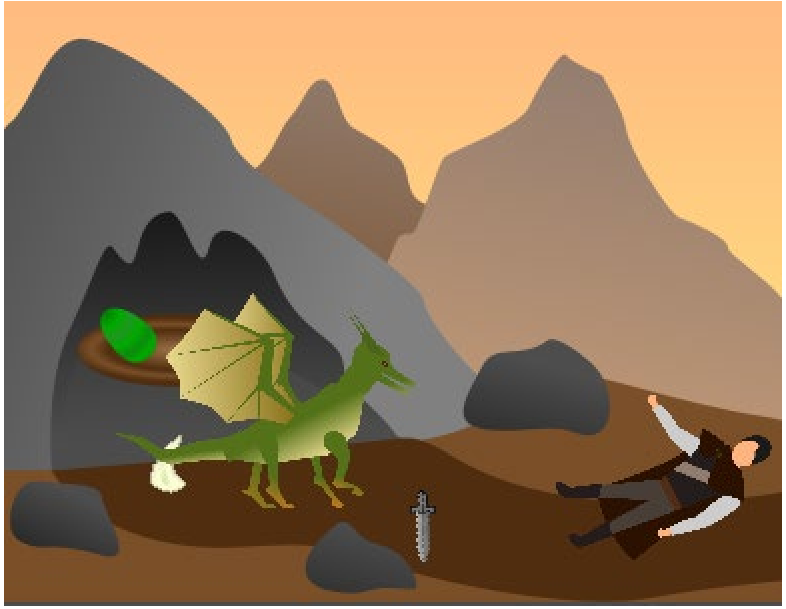
Visualisation of productions

Fight ending with character’s escape

Jeden bohater («BohaterA») atakuje drugiego («BohaterB»), który ucieka do sąsiedniej lokacji («LokacjaB»), gubiąc wszystkie swoje przedmioty w panicznej ucieczce.
«BohaterA» przejmuje kontrolę nad dwoma postaciami kontrolowanymi przez pokonanego.



Visualisation example



Students’ cooperation:
Mikołaj Wrona, Krzysztof Mańka, Karolina Szypura, Dominik Urban

Visualisation example

Students’ cooperation:
Mikołaj Wrona, Krzysztof Mańka, Karolina Szypura, Dominik Urban

Visualisation example

Students' cooperation:
Mikołaj Wrona, Krzysztof Mańka, Karolina Szypura, Dominik Urban