

## ANALIZA CZASOWNIKOWO- RZECZOWNIKOWA

Tworzymy wirtualny świat na planszy typu Hex-a. W świecie będą organizmy posiadające swoje właściwości:

- Siła - mówiąca o zdolności bitewnej naszego organizmu.
- Inicjatywa - mówiąca o kolejności wykonywania ruchu podczas tury.
- Wiek - mówiący ile tur przeżył organizm.

Ponad to organizmy dzielimy na Rośliny i Zwierzęta. Rośliny nie mogą się poruszać ale mogą się rozmnażać samoczynnie.

Zwierzęta zaś mogą się poruszać a przy spotkaniu z innym zwierzęciem na jednym polu, dochodzi do interakcji:

- Zwierzę & zwierzę (różnych gatunków) - Walczy i wygrywa to o większej sile.
- Zwierzę & zwierzę (tego samego gatunku) - rozmnażają się i powstaje zwierzę tego gatunku.
- Zwierzę & Roślina - Jeśli zwierzę jest roślinożercą to zjada Roślinę.

Ponad to jest organizm 'człowiek' którym steruje aktor. Nasz człowiek posiada specjalne umiejętności takie jak:

- Z pokonanych przeciwników podnosi przedmiot.
- Może stworzyć z przedmiotów już posiadanych inne przedmioty kosztem jednej tury wytworzenia ich.

Przedmioty te służą do poprawy statystyk człowieka, oraz przyznania bonusów takich jak:

- magiczny napój (Zwiększa siłę o 10 na 5 tur)

- Błogosławieństwo Teutatesa (w przypadku konfrontacji z silniejszym przeciwnikiem nie umiera, a przesuwa się na sąsiednie wolne pole).

Tylko jeden bonus może być aktywny.

Wynikiem symulacji są :

- Wyniki starć
- Stan populacji

#### LEGENDA:

X – obiekty

Y – czynności

Z – pola

Ψ – warunki

Aktor – aktor

## KARTY CRC

World	
Extends: none	
Heritage: none	
Responsibilities	Collaborators
-organisms : List<List<organism>> +getCoords(enum direction) : (int,int) -generateWorld() : List<List<organism>> -makeOrganism(enum Species, (int,int)) : Organism +getField() : (int,int) +getPopulation() : List<organism,int>	Organism

UI	
Extends: none	
Heritage: none	
Responsibilities	Collaborators
+Log(string) : void +ExecCommand(enum Command) : void +Input() : enum Command -Commands : enum	World Human

GameObject	
Extends: none	
Heritage: none	
Responsibilities	Collaborators
-name : string -type : enum -description : string +getName() : string +getType() : enum type +getDescription() : string	Human Animal

<i>interface Interactions</i>	
Extends: none	
Heritage: none	
Responsibilities	Collaborators
+Interact() : void +Move() : void +Multiply() : void	Plant Animal Human

<i>Organism</i>	
Extends: none	
Heritage: Plant, Animal, Human	
Responsibilities	Collaborators
-strength : int -initiative : int -age : int -species : enum +get(enum value) : int +getSpecies() : enum -coords : (int,int) -dropTable : List<(gameObject,double)> +Drop() : gameObject +getCoords() : (int,int)	World GameObject

Animal	
Extends: Organism	
Heritage: none	
Responsibilities	Collaborators
-type : enum +Interact() : void +Move() : void +Multiply() : void +getType() : enum type	Interactions Plant Human World

Plant	
Extends: Organism	
Heritage: none	
Responsibilities	Collaborators
+Multiply()	Interactions Animal Human World

Human	
Extends: Organism	
Heritage: none	
Responsibilities	Collaborators
-Inventory : List <(gameObject,int)> +Craft(gameObject) : void -buff : enum +Take(gameObject) : void +getBuff(enum buff) +useItem(enum Item) : void	Interactions Animal Human World GameObject

Diagram przypadków użycia

