

Programowanie Obiektowe – Projekt w grupach

Dokumentacja

1. Wstęp

- Zajęcia i prowadzący:

- Wtorek, 18:55
- Mgr inż. Paweł Majewski

- Skład grupy:

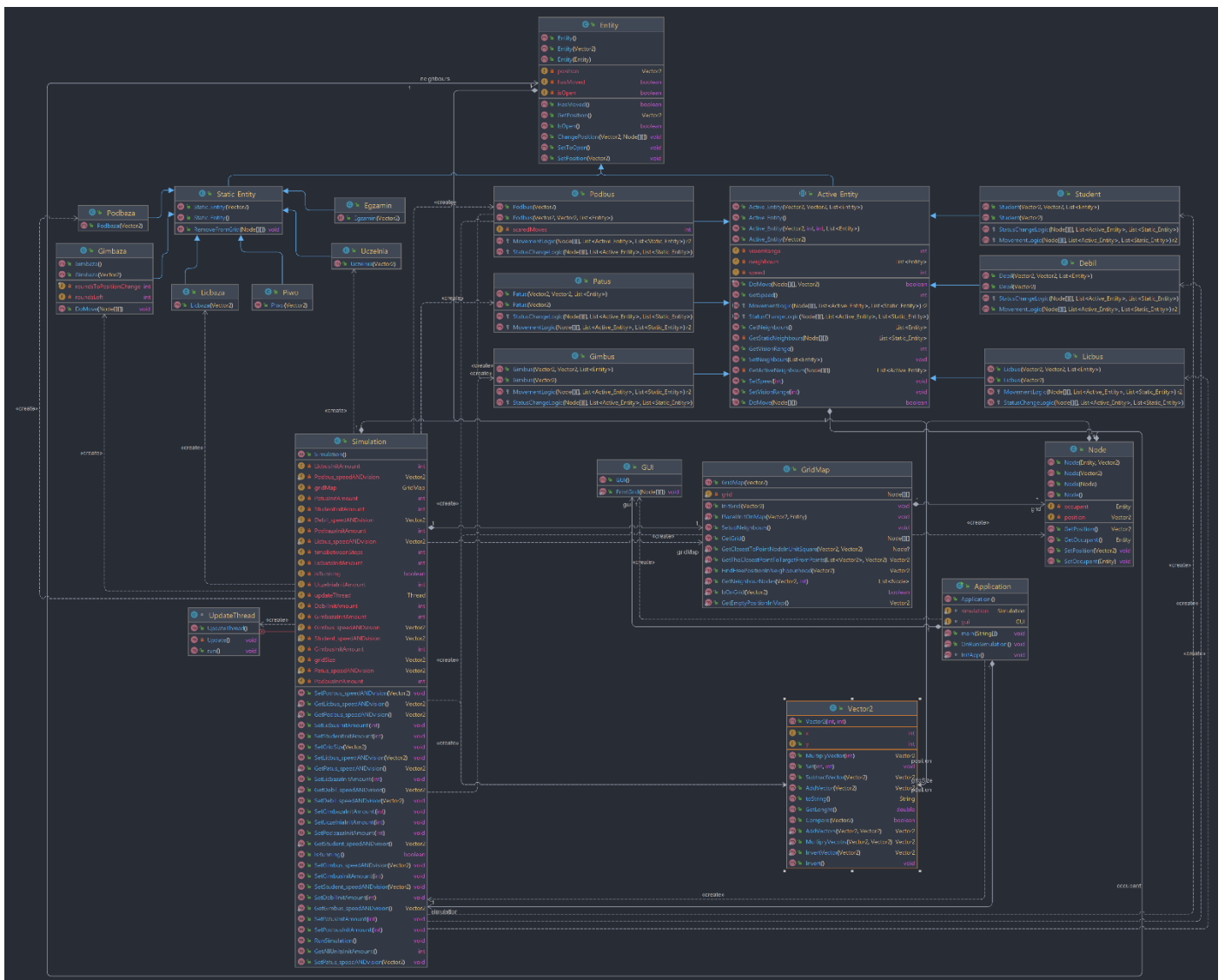
- Michał Dziedziak 263901 (lider)
- Karol Waszkiewicz 263918

- Opis projektu:

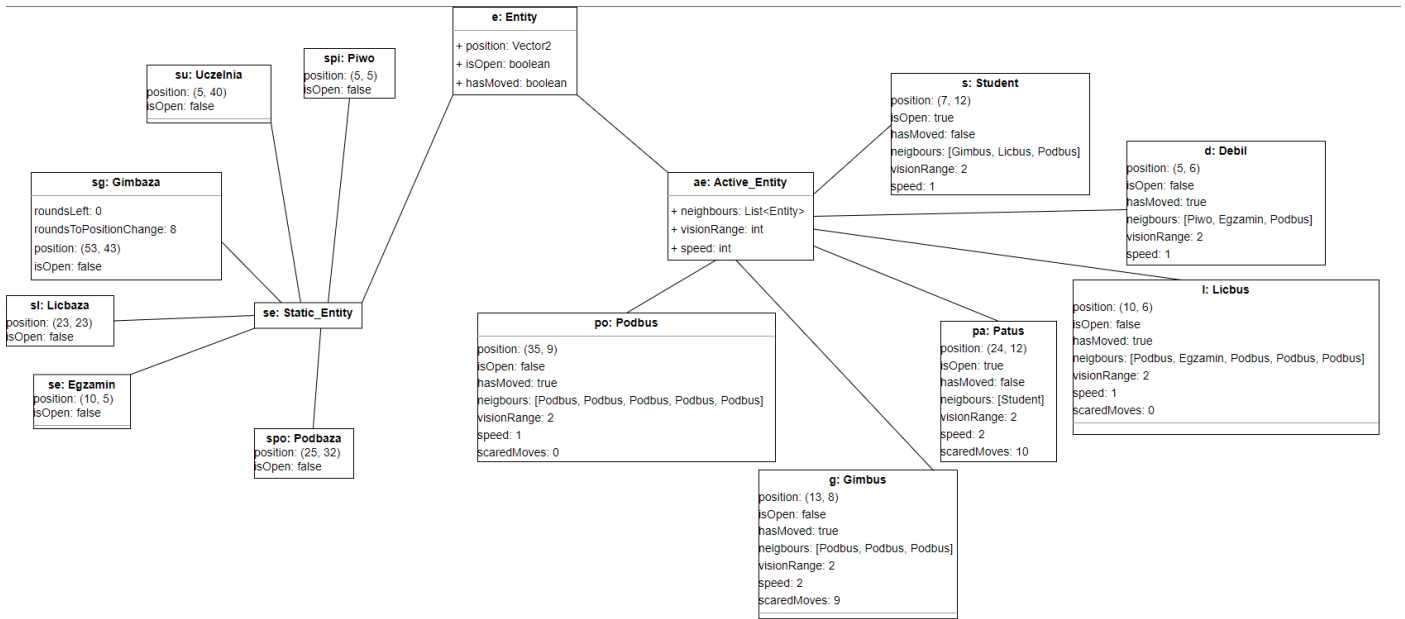
Projekt ten to aplikacja symulująca rozwój społeczności, w której każdy członek musi przejść przez pewien cykl. Celem samej symulacji jest dojście przez pewnego członka do końca cyklu, co ją zatrzymuje. Na prostokątnej planszy umiejscowione są dwa rodzaje bytów: ruchome (wcześniej wspomniani członkowie) oraz nieruchome. Zadaniem tych drugich jest umożliwienie kontynuacji symulacji, gdyż dzięki nim członkowie mogą awansować dalej, w głąb cyklu. Same byty nieruchome także dzielą się na takie, które istnieją od początku symulacji oraz na takie, które mogą pojawiać się i znikać przy pewnych okolicznościach. Wracając do bytów ruchomych, każdy z nich ma specjalne właściwości, co w pewien sposób może ułatwiać bądź utrudniać zakończenie symulacji. Dlatego też symulacja taka w dużej mierze oparta jest na losowości i generatorze liczb losowych – jej długości trwania mogą się diametralnie od siebie różnić.

2. Diagramy

- Diagram klas:



• Diagram obiektów:



• JavaDocs:

Wykonano opisy klas składających się na program, a następnie wygenerowano je. Drzew hierarchiczne wygląda następująco:

- main.**Application**
- main.GUI.**CustomActionListener** (implements [java.awt.event.ActionListener](#), [java.awt.event.FocusListener](#))
- main.DataGathering.**DataGathering**
- main.**Entity**
 - main.**Active_Entity**
 - main.ActiveSubclass.**Debil**
 - main.ActiveSubclass.**Gimbus**
 - main.ActiveSubclass.**Licbus**
 - main.ActiveSubclass.**Patus**
 - main.ActiveSubclass.**Podbus**
 - main.ActiveSubclass.**Student**
 - main.**Static_Entity**
 - main.StaticSubclass.**Egzamin**
 - main.StaticSubclass.**Gimbaza**
 - main.StaticSubclass.**Licbaza**
 - main.StaticSubclass.**Piwo**
 - main.StaticSubclass.**Uczelnia**
- main.**GridMap**
- main.GUI.**GUI**
- main.**Node**
- main.**Simulation**
- main.DataGathering.**SimulationResult**
- main.**Vector2**

Application:

Package `main`

Class Application

`java.lang.Object`
`main.Application`

Inheritance Tree

`public class Application`
`extends Object`

This class represents the application itself

Constructor Summary

Constructors

Constructor	Description
<code>Application()</code>	

Method Summary

All Methods Static Methods Concrete Methods

Modifier and Type	Method	Description
<code>static void</code>	<code>main(String[] args)</code>	

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

CustomActionListener:

Package `main.GUI`

Class CustomActionListener

`java.lang.Object`
`main.GUI.CustomActionListener`

All Implemented Interfaces:
`ActionListener`, `FocusListener`, `EventListener`

`public class CustomActionListener`
`extends Object`
`implements ActionListener`, `FocusListener`

This class makes it possible to customise simulation

Nested Class Summary

Nested Classes

Modifier and Type	Class	Description
<code>static enum</code>	<code>CustomActionListener.ActionType</code>	

Constructor Summary

Constructors

Constructor	Description
<code>CustomActionListener(CustomActionListener.ActionType type, Simulation simulation)</code>	A constructor regarding the simulation #3
<code>CustomActionListener(CustomActionListener.ActionType type, Simulation simulation, JTextField input)</code>	A constructor regarding the simulation #1
<code>CustomActionListener(CustomActionListener.ActionType type, Simulation simulation, JTextField amount, JTextField speed, JTextField visionRange)</code>	A constructor regarding the simulation #2

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
<code>void</code>	<code>actionPerformed(ActionEvent e)</code>	This method is used mostly for the pressing of the buttons
<code>void</code>	<code>focusGained(FocusEvent e)</code>	This method is used to do certain tasks when an object gains focus
<code>void</code>	<code>focusLost(FocusEvent e)</code>	This method is used to do certain tasks when an object loses focus

DataGathering:

Package `main.DataGathering`

Class DataGathering

`java.lang.Object`¹
`main.DataGathering.DataGathering`

```
public class DataGathering
extends Object1
```

This is a raw class designed for generating data for statistics

Constructor Summary

Constructors

Constructor	Description
<code>DataGathering(Simulation sim, boolean doStatisticGathering)</code>	This method is used to gather generated data

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
<code>void</code>	<code>PodbusScenario()</code>	This method generates data for scenario with growing Podbus amount

Methods inherited from class `java.lang.Object`¹

`clone`¹, `equals`¹, `finalize`¹, `getClass`¹, `hashCode`¹, `notify`¹, `notifyAll`¹, `toString`¹, `wait`¹, `wait`¹, `wait`¹

Constructor Details

DataGathering

```
public DataGathering(Simulation sim,
                    boolean doStatisticGathering)
```

This method is used to gather generated data

Parameters:

`sim` - This parameter is used to execute commands regarding simulation

`doStatisticGathering` - This parameter sets a boolean value for gathering of the statistics. True allows it.

Method Details

PodbusScenario

```
public void PodbusScenario()
```

This method generates data for scenario with growing Podbus amount

Entity:

Package main

Class Entity

java.lang.Object[Ⓐ]
main.Entity

Direct Known Subclasses:

Active_Entity, Static_Entity

```
public class Entity  
extends ObjectⒶ
```

This class represents all entities, so the things which are placed on the GridMap

Constructor Summary

Constructors

Constructor	Description
<code>Entity()</code>	This constructor creates a new entity
<code>Entity(Entity copy)</code>	This constructor creates a new entity
<code>Entity(Vector2 position)</code>	This constructor creates a new entity

Method Summary

All Methods	Static Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description	
static void	<code>ChangeAmountOfGivenSubclass(Entity ent, int change)</code>	This method changes the amount of given subclass by a given amount	
void	<code>ChangePosition(Vector2 pos, Node[][] grid)</code>	This method changes position of an entity	
Vector2	<code>GetPosition()</code>	This getter is used to return a position value of an entity	
boolean	<code>IsOpen()</code>	This boolean returns true or false statements regarding the entity's status of being opened or not	
void	<code>RemoveFromGrid(Node[][] grid)</code>	This method removes an entity from grid	
static void	<code>ResetAmountOfAllSubclasses()</code>	This method changes amount of all subclasses to 0	
void	<code>SetPosition(Vector2 pos)</code>	This setter is used to set a certain position to an entity	
void	<code>SetToOpen()</code>	This sets the entity's status of being opened to true	

Methods inherited from class java.lang.Object[Ⓐ]

`cloneⒶ`, `equalsⒶ`, `finalizeⒶ`, `getClassⒶ`, `hashCodeⒶ`, `notifyⒶ`, `notifyAllⒶ`, `toStringⒶ`, `waitⒶ`, `waitⒶ`, `waitⒶ`

Active_Entity:

Package main

Class Active_Entity

java.lang.Object¹
main.Entity
main.Active_Entity

Direct Known Subclasses:

Debil, Gimbus, Licbus, Patus, Podbus, Student

```
public abstract class Active_Entity  
extends Entity
```

This class is used to program entities capable of moving on the map / grid

Constructor Summary

Constructors

Constructor	Description
<code>Active_Entity()</code>	
<code>Active_Entity(Vector2 speedANDvision)</code>	This constructor is used to create a new active entity
<code>Active_Entity(Vector2 position, int speed, int visionRange)</code>	This constructor is used to create a new active entity
<code>Active_Entity(Vector2 position, Vector2 speedANDvision)</code>	This constructor is used to create a new active entity

Method Summary

All Methods	Instance Methods	Abstract Methods	Concrete Methods
Modifier and Type	Method	Description	
void	<code>AddMoves(int amount)</code>	This setter adds a given number of moves to an entity	
final boolean	<code>DoMove(Node[][] grid, Vector2 forcedDir)</code>	This method initialises an entity's movement.	
protected Vector2	<code>GetMovementVectorToStaticEntity(List¹<Static_Entity> staticNeigh, Class¹<? extends Static_Entity> targetClass)</code>	This method returns a local vector to given static entity.	
int	<code>GetSpeed()</code>	This getter returns the speed value of an entity	
int	<code>GetVisionRange()</code>	This getter returns the range of an entity's vision	
protected boolean	<code>IsStaticEntityInNeighborhood(List¹<Static_Entity> staticNeigh, Class¹<? extends Static_Entity> target)</code>	This method returns true if there is a given static entity in any of surrounding nodes	
protected abstract Vector2	<code>MovementLogic(Node[][] grid, List¹<Active_Entity> activeNeigh, List¹<Static_Entity> staticNeigh)</code>	This method returns local movement vector which indicates direction to which will entity go in given round	
void	<code>SetSpeed(int speed)</code>	This setter sets a given speed to an entity	
void	<code>SetVisionRange(int visionRange)</code>	This setter sets a given amount of vision range for a certain entity	
protected abstract boolean	<code>StatusChangeLogic(Node[][] grid, List¹<Active_Entity> activeNeigh, List¹<Static_Entity> staticNeigh)</code>	This method returns true if any of end conditions is met	

Methods inherited from class main.Entity

`ChangeAmountOfGivenSubclass`, `ChangePosition`, `GetPosition`, `IsOpen`, `RemoveFromGrid`, `ResetAmountOfAllSubclasses`, `SetPosition`, `SetToOpen`

Methods inherited from class java.lang.Object¹

`clone1`, `equals1`, `finalize1`, `getClass1`, `hashCode1`, `notify1`, `notifyAll1`, `toString1`, `wait1`, `wait1`, `wait1`

Debil:

Package `main.ActiveSubclass`

Class `Debil`

```
java.lang.ObjectⒶ
  main.Entity
    main.Active_Entity
      main.ActiveSubclass.Debil
```

Class `Debil`

```
public class Debil
  extends Active_Entity
```

This class represents `Debil` entity. It is a special entity which `Student` entity can become randomly every round (10%). `Debil` scares every active entity, except other `Debil` entities or `Student` entities. It behaves similarly to `Student`, except it cannot consume `Piwo`. In order to become `Student` again, `Debil` needs to stand next to `Egzamin`. This action turns `Debil` back to `Student` and removes `Egzamin`. It can alternatively become `Student` every new round with 10% chance of this happening.

Field Summary

Fields

Modifier and Type	Field	Description
static int	<code>amount</code>	

Constructor Summary

Constructors

Constructor	Description
<code>Debil(Vector2 speedANDvision)</code>	This constructor behaves the same as the previous one, except it takes fewer parameters
<code>Debil(Vector2 position, Vector2 speedANDvision)</code>	This constructor creates a new <code>Debil</code> entity.

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
<code>Vector2</code>	<code>MovementLogic(Node[][] grid, List[Ⓐ]<Active_Entity> activeNeigh, List[Ⓐ]<Static_Entity> staticNeigh)</code>	This method defines the logic of <code>Debil</code> 's movement.
protected boolean	<code>StatusChangeLogic(Node[][] grid, List[Ⓐ]<Active_Entity> activeNeigh, List[Ⓐ]<Static_Entity> staticNeigh)</code>	This method defines possible changes in the <code>Debil</code> entity itself.

Methods inherited from class `main.Active_Entity`

`AddMoves`, `DoMove`, `GetMovementVectorToStaticEntity`, `GetSpeed`, `GetVisionRange`, `IsStaticEntityInNeighborhood`, `SetSpeed`, `SetVisionRange`

Methods inherited from class `main.Entity`

`ChangeAmountOfGivenSubclass`, `ChangePosition`, `GetPosition`, `IsOpen`, `RemoveFromGrid`, `ResetAmountOfAllSubclasses`, `SetPosition`, `SetToOpen`

Methods inherited from class `java.lang.ObjectⒶ`

`cloneⒶ`, `equalsⒶ`, `finalizeⒶ`, `getClassⒶ`, `hashCodeⒶ`, `notifyⒶ`, `notifyAllⒶ`, `toStringⒶ`, `waitⒶ`, `waitⒶ`, `waitⒶ`

Gimbus:

Package main.ActiveSubclass

Class Gimbus

java.lang.Object[Ⓙ]
main.Entity
main.Active_Entity
main.ActiveSubclass.Gimbus

```
public class Gimbus  
extends Active_Entity
```

This class represents Gimbus entity. It's an entity which Podbus can change into after standing next to Gimbaza static entity. It scares groups of Podbus entities up to 3 members. Higher amounts will scare Gimbus instead. It is always going to be scared by either Student or Debil. If two Gimbus entities stand next to eachother, they can either both become Patus entities; eliminate eachother; or nothing could happen. Each of these events has 33% chance of occuring. In order to become Licbus, Gimbus must stand next to Licbaza static entity.

Field Summary

Fields

Modifier and Type	Field	Description
static int	amount	

Constructor Summary

Constructors

Constructor	Description
Gimbus(Vector2 speedANDvision)	This constructor behaves the same as the previous one, except it takes fewer parameters
Gimbus(Vector2 position, Vector2 speedANDvision)	This constructor creates a new Gimbus entity

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
protected Vector2	MovementLogic(Node[][] grid, List [Ⓙ] <Active_Entity> activeNeigh, List [Ⓙ] <Static_Entity> staticNeigh)	This method defines the logic of Gimbus' movement.
void	SetScaredMoves(int val)	This setter sets the scared moves if right conditions are met
protected boolean	StatusChangeLogic(Node[][] grid, List [Ⓙ] <Active_Entity> activeNeigh, List [Ⓙ]	This method defines possible changes in the Gimbus entity itself.

Licbus:

Package main.ActiveSubclass

Class Licbus

```
java.lang.ObjectⒶ
  main.Entity
    main.Active_Entity
      main.ActiveSubclass.Licbus
```

```
public class Licbus
  extends Active_Entity
```

This class represents Licbus entity. It's an entity which Gimbus can change into after standing next to Licbaza or Patus, after consuming Egzamin. It is always going to be scared by either Student or Debil. Licbus entities have 20% chance of leaving an Egzamin entity behind every round. In order to become Student, it needs to place itself next to Uczelnia static entity.

Field Summary

Fields

Modifier and Type	Field	Description
static int	amount	

Constructor Summary

Constructors

Constructor	Description
Licbus(Vector2 speedANDvision)	This constructor behaves the same as the previous one, except it takes fewer parameters
Licbus(Vector2 position, Vector2 speedANDvision)	This constructor creates a new Licbus entity

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
protected Vector2	MovementLogic(Node[][] grid, List [Ⓐ] <Active_Entity> activeNeigh, List [Ⓐ] <Static_Entity> staticNeigh)	This method defines the logic of Licbus' movement.
protected boolean	StatusChangeLogic(Node[][] grid, List [Ⓐ] <Active_Entity> activeNeigh, List [Ⓐ] <Static_Entity> staticNeigh)	This method defines possible changes in the Licbus entity itself.

Methods inherited from class main.Active_Entity

AddMoves, DoMove, GetMovementVectorToStaticEntity, GetSpeed, GetVisionRange, IsStaticEntityInNeighborhood, SetSpeed, SetVisionRange

Methods inherited from class main.Entity

ChangeAmountOfGivenSubclass, ChangePosition, GetPosition, IsOpen, RemoveFromGrid, ResetAmountOfAllSubclasses, SetPosition, SetToOpen

Methods inherited from class java.lang.Object[Ⓐ]

clone[Ⓐ], equals[Ⓐ], finalize[Ⓐ], getClass[Ⓐ], hashCode[Ⓐ], notify[Ⓐ], notifyAll[Ⓐ], toString[Ⓐ], wait[Ⓐ], wait[Ⓐ], wait[Ⓐ]

Patus:

Class Patus

```
java.lang.ObjectⒶ
  main.Entity
    main.Active_Entity
      main.ActiveSubclass.Patus
```

```
public class Patus
  extends Active_Entity
```

This class represents Patus entity. It's a special entity which two Gimbus entities can turn into after meeting each other (33% chance). It scares groups of Podbus entities up to 3 members. Higher amounts will scare Patus instead. It is always going to be scared by either Student or Debil. Patus entities have 20% chance of leaving a Piwo entity behind them every round. In order to become Licbus, Podbus entity needs to consume Egzamin.

Field Summary

Fields

Modifier and Type	Field	Description
static int	amount	

Constructor Summary

Constructors

Constructor	Description
<code>Patus(Vector2 speedANDvision)</code>	This constructor behaves the same as the previous one, except it takes fewer parameters
<code>Patus(Vector2 position, Vector2 speedANDvision)</code>	This constructor creates a new Patus entity

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
protected Vector2	<code>MovementLogic(Node[][] grid, List[Ⓐ]<Active_Entity> activeNeigh, List[Ⓐ]<Static_Entity> staticNeigh)</code>	This method defines the logic of Patus' movement.
void	<code>SetScaredMoves(int val)</code>	This setter sets the scared moves if right conditions are met
protected boolean	<code>StatusChangeLogic(Node[][] grid, List[Ⓐ]<Active_Entity> activeNeigh, List[Ⓐ]<Static_Entity> staticNeigh)</code>	This method defines possible changes in the Patus entity itself.

Methods inherited from class main.Active_Entity

AddMoves, DoMove, GetMovementVectorToStaticEntity, GetSpeed, GetVisionRange, IsStaticEntityInNeighborhood, SetSpeed, SetVisionRange

Methods inherited from class main.Entity

ChangeAmountOfGivenSubclass, ChangePosition, GetPosition, IsOpen, RemoveFromGrid, ResetAmountOfAllSubclasses, SetPosition, SetToOpen

Methods inherited from class java.lang.Object[Ⓐ]

clone[Ⓐ], equals[Ⓐ], finalize[Ⓐ], getClass[Ⓐ], hashCode[Ⓐ], notify[Ⓐ], notifyAll[Ⓐ], toString[Ⓐ], wait[Ⓐ], wait[Ⓐ], wait[Ⓐ]

Podbus:

Package main.ActiveSubclass

Class Podbus

java.lang.Object[Ⓔ]
main.Entity
main.Active_Entity
main.ActiveSubclass.Podbus

```
public class Podbus  
extends Active_Entity
```

This class represents Podbus entity. It's the entity lowest in hierarchy, it cannot be created by other entities doing something. It is capable of grouping up with other Podbus entities, making them travel together as a single entity. Groups of Podbus entities up to 3 members will be scared by nearby Gimbus and Patus entities. Groups of Podbus entities over 3 members will scare nearby Gimbus and Patus entities. It is always going to be scared by either Student or Debil entities. In order to become Gimbus, Patus entity needs to stand next to Gimbaza static entity.

Field Summary

Fields

Modifier and Type	Field	Description
static int	amount	

Constructor Summary

Constructors

Constructor	Description
<code>Podbus(Vector2 speedANDvision)</code>	This constructor behaves the same as the previous one, except it takes fewer parameters
<code>Podbus(Vector2 position, Vector2 speedANDvision)</code>	This constructor creates a new Podbus entity

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
int	<code>GetGroupSize()</code>	This method returns size of List group in Podbus class
protected Vector2	<code>MovementLogic(Node[][] grid, List<Active_Entity> activeNeigh, List<Static_Entity> staticNeigh)</code>	This method defines the logic of Podbus' movement.
void	<code>SetScaredMoves(int val)</code>	This setter sets the scared moves if right conditions are met
protected boolean	<code>StatusChangeLogic(Node[][] grid, List<Active_Entity> activeNeigh, List<Static_Entity> staticNeigh)</code>	This method defines possible changes in the Podbus entity itself.

Methods inherited from class main.Active_Entity

AddMoves, DoMove, GetMovementVectorToStaticEntity, GetSpeed, GetVisionRange, IsStaticEntityInNeighborhood, SetSpeed, SetVisionRange

Methods inherited from class main.Entity

ChangeAmountOfGivenSubclass, ChangePosition, GetPosition, IsOpen, RemoveFromGrid, ResetAmountOfAllSubclasses, SetPosition, SetToOpen

Student:

Package main.ActiveSubclass

Class Student

```
java.lang.ObjectⒺ
  main.Entity
    main.Active_Entity
      main.ActiveSubclass.Student
```

```
public class Student
extends Active_Entity
```

This class represents Student entity. It's an entity which Licbus can change into after standing next to Uczelnia or Debil, after either consuming Egzamin or at the start of a new round (10% chance). Student entity has 10% chance of turning into Debil each round. Student scares every active entity, except other Student entities or Debil entities. Every Student will wander randomly until they find Piwo entity, which attracts them. Consuming it finishes the simulation.

Field Summary

Fields

Modifier and Type	Field	Description
static int	amount	

Constructor Summary

Constructors

Constructor	Description
Student(Vector2 speedANDvision)	This constructor behaves the same as the previous one, except it takes fewer parameters
Student(Vector2 position, Vector2 speedANDvision)	This constructor creates a new Student entity

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
protected Vector2	MovementLogic(Node[][] grid, List [Ⓔ] <Active_Entity> activeNeigh, List [Ⓔ] <Static_Entity> staticNeigh)	This method defines the logic of Student's movement.
protected boolean	StatusChangeLogic(Node[][] grid, List [Ⓔ] <Active_Entity> activeNeigh, List [Ⓔ] <Static_Entity> staticNeigh)	This method defines possible changes in the Student entity itself.

Methods inherited from class main.Active_Entity

AddMoves, DoMove, GetMovementVectorToStaticEntity, GetSpeed, GetVisionRange, IsStaticEntityInNeighborhood, SetSpeed, SetVisionRange

Methods inherited from class main.Entity

ChangeAmountOfGivenSubclass, ChangePosition, GetPosition, IsOpen, RemoveFromGrid, ResetAmountOfAllSubclasses, SetPosition, SetToOpen

Methods inherited from class java.lang.Object[Ⓔ]

clone[Ⓔ], equals[Ⓔ], finalize[Ⓔ], getClass[Ⓔ], hashCode[Ⓔ], notify[Ⓔ], notifyAll[Ⓔ], toString[Ⓔ], wait[Ⓔ], wait[Ⓔ], wait[Ⓔ]

Static_Entity:

Package main

Class Static_Entity

java.lang.Object[Ⓐ]
main.Entity
main.Static_Entity

Direct Known Subclasses:

Egzamin, Gimbaza, Licbaza, Piwo, Uczelnia

```
public class Static_Entity  
extends Entity
```

This class represents static entities. Their purpose is to change active entites into other variants or to finish the simulation (Piwo).

Constructor Summary

Constructors

Constructor	Description
<code>Static_Entity()</code>	
<code>Static_Entity(Vector2 position)</code>	

Method Summary

Methods inherited from class main.Entity

`ChangeAmountOfGivenSubclass`, `ChangePosition`, `GetPosition`, `IsOpen`, `RemoveFromGrid`, `ResetAmountOfAllSubclasses`, `SetPosition`, `SetToOpen`

Methods inherited from class java.lang.Object[Ⓐ]

`cloneⒶ`, `equalsⒶ`, `finalizeⒶ`, `getClassⒶ`, `hashCodeⒶ`, `notifyⒶ`, `notifyAllⒶ`, `toStringⒶ`, `waitⒶ`, `waitⒶ`, `waitⒶ`

Egzamin:

Package main.StaticSubclass

Class Egzamin

java.lang.Object[Ⓐ]
main.Entity
main.Static_Entity
main.StaticSubclass.Egzamin

```
public class Egzamin  
extends Static_Entity
```

This class represents an Egzamin entity. This entity has 20% chance of being dropped by a Licbus entity. It is used for converting Patus entities to Licbus entities and Debil entities to Student entities.

Field Summary

Fields

Modifier and Type	Field	Description
static int	amount	

Constructor Summary

Constructors

Constructor	Description
<code>Egzamin(Vector2 position)</code>	This method defines the position of Egzamin.

Method Summary

Methods inherited from class main.Entity

`ChangeAmountOfGivenSubclass`, `ChangePosition`, `GetPosition`, `IsOpen`, `RemoveFromGrid`, `ResetAmountOfAllSubclasses`, `SetPosition`, `SetToOpen`

Methods inherited from class java.lang.Object[Ⓐ]

`cloneⒶ`, `equalsⒶ`, `finalizeⒶ`, `getClassⒶ`, `hashCodeⒶ`, `notifyⒶ`, `notifyAllⒶ`, `toStringⒶ`, `waitⒶ`, `waitⒶ`, `waitⒶ`

Gimbaza:

Package `main.StaticSubclass`

Class Gimbaza

`java.lang.Object`
`main.Entity`
`main.Static_Entity`
`main.StaticSubclass.Gimbaza`

```
public class Gimbaza
extends Static_Entity
```

This class represents a Gimbaza entity. This entity is crucial for Podbus entities to turn into Gimbus entities. Every Gimbaza randomly changes its position after a given number of rounds.

Constructor Summary

Constructors

Constructor	Description
<code>Gimbaza()</code>	
<code>Gimbaza(Vector2 position)</code>	This method defines the position of Gimbaza.

Method Summary

All Methods

Instance Methods

Concrete Methods

Modifier and Type	Method	Description
<code>void</code>	<code>DoMove(Node[][] grid)</code>	This method is used in regard of Gimbaza entity changing its position every given amount of rounds

Methods inherited from class `main.Entity`

`ChangeAmountOfGivenSubclass`, `ChangePosition`, `GetPosition`, `IsOpen`, `RemoveFromGrid`, `ResetAmountOfAllSubclasses`, `SetPosition`, `SetToOpen`

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Licbaza:

Package `main.StaticSubclass`

Class Licbaza

`java.lang.Object`
`main.Entity`
`main.Static_Entity`
`main.StaticSubclass.Licbaza`

```
public class Licbaza  
extends Static_Entity
```

This class represents a Licbaza entity. Its use is to allow Gimbus entities turn into Licbus entities.

Constructor Summary

Constructors

Constructor	Description
<code>Licbaza(Vector2 position)</code>	This method defines the position of Licbaza

Method Summary

Methods inherited from class `main.Entity`

`ChangeAmountOfGivenSubclass`, `ChangePosition`, `GetPosition`, `IsOpen`, `RemoveFromGrid`, `ResetAmountOfAllSubclasses`, `SetPosition`, `SetToOpen`

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Piwo:

Package `main.StaticSubclass`

Class Piwo

`java.lang.Object`
`main.Entity`
`main.Static_Entity`
`main.StaticSubclass.Piwo`

```
public class Piwo
extends Static_Entity
```

This class represents a Piwo entity. It has 20% of being dropped by Patus entities. Its use is to be consumed by Student entities.

Field Summary

Fields

Modifier and Type	Field	Description
static int	<code>amount</code>	

Constructor Summary

Constructors

Constructor	Description
<code>Piwo(Vector2 position)</code>	This method defines the position of Piwo

Method Summary

Methods inherited from class `main.Entity`

`ChangeAmountOfGivenSubclass`, `ChangePosition`, `GetPosition`, `IsOpen`, `RemoveFromGrid`, `ResetAmountOfAllSubclasses`, `SetPosition`, `SetToOpen`

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Uczelnia:

Package `main.StaticSubclass`

Class Uczelnia

`java.lang.Object`[Ⓔ]
`main.Entity`
`main.Static_Entity`
`main.StaticSubclass.Uczelnia`

```
public class Uczelnia
extends Static_Entity
```

This class represents an Uczelnia entity. Its use is to allow Licbus entities turn into Student entities

Constructor Summary

Constructors

Constructor	Description
<code>Uczelnia(Vector2 position)</code>	This method defines the position of Uczelnia

Method Summary

Methods inherited from class `main.Entity`

`ChangeAmountOfGivenSubclass`, `ChangePosition`, `GetPosition`, `IsOpen`, `RemoveFromGrid`, `ResetAmountOfAllSubclasses`, `SetPosition`, `SetToOpen`

Methods inherited from class `java.lang.Object`[Ⓔ]

`clone`[Ⓔ], `equals`[Ⓔ], `finalize`[Ⓔ], `getClass`[Ⓔ], `hashCode`[Ⓔ], `notify`[Ⓔ], `notifyAll`[Ⓔ], `toString`[Ⓔ], `wait`[Ⓔ], `wait`[Ⓔ], `wait`[Ⓔ]

GridMap:

Package main

Class GridMap

java.lang.Object[Ⓙ]
main.GridMap

```
public class GridMap  
extends ObjectⒿ
```

This class defines the gridMap used for the simulation

Constructor Summary

Constructors

Constructor	Description
<code>GridMap(Vector2 size)</code>	This constructor is used to generate a new gridMap

Method Summary

All Methods	Static Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description	
static Node	<code>GetClosestToPointNodeInUnitsSquare(Vector2 center, Vector2 targetPos)</code>	This method returns closest (free) node to target position from nodes which surrounds given point (center).	
static Vector2	<code>GetEmptyPositionInMap()</code>	This method returns global position of random node (without any occupant) in grid	
static Vector2	<code>GetFreePositionInNeighbourhood(Vector2 position)</code>	This method returns position of random free node from nodes in surrounding of node with given position.	
Node[][]	<code>GetGrid()</code>	This getter is used to return the grid property	
static List [Ⓙ] <Node>	<code>GetNeighbourNodes(Vector2 center, int deepness)</code>	This method returns a list of nodes which surround the node with given coordinates (center).	
static Vector2	<code>GetTheClosestPointToTargetFromPoints(List[Ⓙ]<Vector2> points, Vector2 target)</code>	This method returns the closest position to given target from positions list	
void	<code>InitGrid(Vector2 gridSize)</code>	This method initiates grid size and initialises each node	
static boolean	<code>IsOnGrid(Vector2 pos)</code>	This method checks whether position is inside a grid's boundaries	
static void	<code>PlaceUnitOnMap(Vector2 pos, Entity unit)</code>	This method sets a unit on a given position	

Methods inherited from class java.lang.Object[Ⓙ]

`cloneⒿ`, `equalsⒿ`, `finalizeⒿ`, `getClassⒿ`, `hashCodeⒿ`, `notifyⒿ`, `notifyAllⒿ`, `toStringⒿ`, `waitⒿ`, `waitⒿ`, `waitⒿ`

GUI:

Package `main.GUI`

Class GUI

`java.lang.Object`[↗]
`main.GUI.GUI`

```
public class GUI  
extends Object↗
```

This class is used for creating and handling GUI

Constructor Summary

Constructors

Constructor	Description
<code>GUI(Simulation sim)</code>	

Method Summary

All Methods

Static Methods

Concrete Methods

Modifier and Type	Method	Description
static void	<code>InitalizeNodeGridGui(Node[][] gridMap, int roundNumber)</code>	This method initiates GUI part which handles showing the grid
static void	<code>PrintGridInConsole(Node[][] gridMap, int timeBetweenSteps)</code>	This method prints grid in console and put thread to sleep for given amount
static void	<code>SetSimulationStatus(String[↗] status)</code>	
static void	<code>UpdateGridGui(Node[][] gridMap, int roundNumber)</code>	This method updates grid in GUI

Methods inherited from class `java.lang.Object`[↗]

`clone`[↗], `equals`[↗], `finalize`[↗], `getClass`[↗], `hashCode`[↗], `notify`[↗], `notifyAll`[↗], `toString`[↗], `wait`[↗], `wait`[↗], `wait`[↗]

Node:

Package `main`

Class Node

`java.lang.Object`
`main.Node`

```
public class Node
extends Object
```

This class represents a single node on which an entity can be placed

Constructor Summary

Constructors

Constructor	Description
<code>Node()</code>	This constructor creates a node
<code>Node(Entity occupant, Vector2 position)</code>	This constructor creates a node
<code>Node(Node node)</code>	This constructor creates a node
<code>Node(Vector2 pos)</code>	This constructor creates a node

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
<code>Entity</code>	<code>GetOccupant()</code>	This getter is used to return an information about what entity is placed on the node
<code>Vector2</code>	<code>GetPosition()</code>	This getter gets the position of a node
<code>void</code>	<code>SetOccupant(Entity occupant)</code>	This setter sets a chosen entity on the node
<code>void</code>	<code>SetPosition(Vector2 pos)</code>	This setter sets the position of a node

Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Simulation #1:

Package main

Class Simulation

java.lang.Object¹²
main.Simulation

```
public class Simulation  
extends Object12
```

This class represents the simulation process of the program

Field Summary

Fields

Modifier and Type	Field	Description
static int	RoundCount	

Constructor Summary

Constructors

Constructor	Description
Simulation()	This constructor creates the simulation

Method Summary

All Methods	Static Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description	
int	GetAllUnitsInitAmount()	This getter returns the amount of all entities	
static Vector2	GetDebil_speedANDvision()	This getter returns the values of Debil's speed and vision	
int	GetDebilInitAmount()	This getter returns the amount of Debil entities	
int	GetEgzaminInitAmount()	This getter returns the amount of Egzamin entities	
int	GetGimbazaInitAmount()	This getter returns the amount of Gimbaza entities	
static Vector2	GetGimbus_speedANDvision()	This getter returns the values of Gimbus' speed and vision	
int	GetGimbusInitAmount()	This getter returns the amount of Gimbus entities	
GridMap	GetGridMap()	This getter returns the gridMap	
Vector2	GetGridSize()	This getter returns the gridMap's size	
int	GetLicbazaInitAmount()	This getter returns the amount of Licbaza entities	
static Vector2	GetLicbus_speedANDvision()	This getter returns the values of Licbus' speed and vision	
int	GetLicbusInitAmount()	This getter returns the amount of Licbus entities	
static Vector2	GetPatus_speedANDvision()	This getter returns the values of Patus' speed and vision	

Simulation #2:

int	GetPatusInitAmount()	This getter returns the amount of Patus entities
int	GetPiwoInitAmount()	This getter returns the amount of Piwo entities
static Vector2	GetPodbus_speedANDvision()	This getter returns the values of Podbus' speed and vision
int	GetPodbusInitAmount()	This getter returns the amount of Podbus entities
SimulationResult	GetResult()	This getter returns the results
static Vector2	GetStudent_speedANDvision()	This getter returns the values of Student's speed and vision
int	GetStudentInitAmount()	This getter returns the amount of Student entities
static int	GetTimeBetweenSteps()	This getter returns the time between steps of an entity
int	GetUczelniaInitAmount()	This getter returns the amount of Uczelnia entities
void	InitSimulation()	This method initiates grid for simulation
boolean	IsRunning()	This boolean returns true or false whether the simulation is running
void	PauseSimulation()	This method will pause simulation if any is currently running
void	RunSimulation()	This method runs simulation in new thread
void	RunSimulationWithoutNewThred()	This method runs simulation without awaking new thread
void	SetDebil_speedANDvision(Vector2 speedANDvision)	This setter sets the speed and vision values of Debil entity
void	SetDebilInitAmount(int amount)	This setter sets the initial amount of Debil entities on the gridMap
void	SetEgzaminInitAmount(int amount)	This setter sets the initial amount of Egzamin entities on the gridMap
void	SetGimbazaInitAmount(int amount)	This setter sets the initial amount of Gimbaza entities on the gridMap
void	SetGimbus_speedANDvision(Vector2 speedANDvision)	This setter sets the speed and vision values of Gimbus entity
void	SetGimbusInitAmount(int amount)	This setter sets the initial amount of Gimbus entities on the gridMap
void	SetGridSize(Vector2 size)	This setter sets the size of the gridMap
static void	SetIsPrintingGrid(boolean val)	This setter sets the boolean value of the isPrintingGrid property
void	SetLicbazaInitAmount(int amount)	This setter sets the initial amount of Licbaza entities on the gridMap
void	SetLicbus_speedANDvision(Vector2 speedANDvision)	This setter sets the speed and vision values of Licbus entity
void	SetLicbusInitAmount(int amount)	This setter sets the initial amount of Licbus entities on the gridMap
void	SetPatus_speedANDvision(Vector2 speedANDvision)	This setter sets the speed and vision values of Patus entity
void	SetPatusInitAmount(int amount)	This setter sets the initial amount of Patus entities on the gridMap
void	SetPiwoInitAmount(int amount)	This setter sets the initial amount of Piwo entities on the gridMap
void	SetPodbus_speedANDvision(Vector2 speedANDvision)	This setter sets the speed and vision values of Podbus entity
void	SetPodbusInitAmount(int amount)	This setter sets the initial amount of Podbus entities on the gridMap
void	SetStudent_speedANDvision(Vector2 speedANDvision)	This setter sets the speed and vision values of Student entity
void	SetStudentInitAmount(int amount)	This setter sets the initial amount of Student entities on the gridMap

Simulation #3:

static void	SetTimeBetweenSteps(int val)	This setter sets time between steps of an entity
void	SetUczelniaInitAmount(int amount)	This setter sets the initial amount of Uczelnia entities on the gridMap
void	SetupSimulationProperties(Vector2 gridSize, Vector2 debil_speedANDvision, Vector2 gimbus_speedANDvision, Vector2 licbus_speedANDvision, Vector2 patus_speedANDvision, Vector2 podbus_speedANDvision, Vector2 student_speedANDvision, int debilInitAmount, int gimbusInitAmount, int licbusInitAmount, int patusInitAmount, int podbusInitAmount, int studentInitAmount, int gimbazaInitAmount, int licbazaInitAmount, int uczeIniaInitAmount, int piwoInitAmount, int egzaminInitAmount)	This method will setup values used as begin conditions for simulation
Methods inherited from class java.lang.Object ¹		
clone ¹ , equals ¹ , finalize ¹ , getClass ¹ , hashCode ¹ , notify ¹ , notifyAll ¹ , toString ¹ , wait ¹ , wait ¹ , wait ¹		

SimulationResult:

Package main.DataGathering

Class SimulationResult

java.lang.Object¹
main.DataGathering.SimulationResult

public class SimulationResult
extends Object¹

This class is used for gathering data after simulation

Field Summary

Fields

Modifier and Type	Field	Description
int	finNumberOFdebil	
int	finNumberOFegzamin	
int	finNumberOFgimbus	
int	finNumberOFLicbus	
int	finNumberOFPatus	
int	finNumberOFpiwo	
int	finNumberOFpodbus	
int	finNumberStudent	
Vector2	gridSize	
int	inicialNumberOFdebil	
int	inicialNumberOFegzamin	
int	inicialNumberOFgimbaza	
int	inicialNumberOFgimbus	
int	inicialNumberOFLicbaza	
int	inicialNumberOFLicbus	
int	inicialNumberOFPatus	
int	inicialNumberOFpiwo	
int	inicialNumberOFpodbus	
int	inicialNumberOFuczeInia	
int	inicialNumberStudent	
int	numberOFendCondition	
int	numberOFrounds	

Constructor Summary

Constructors

Constructor	Description
SimulationResult()	

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
String ¹	toString()	This method returns all the crucial data after finishing the simulation

Methods Inherited from class java.lang.Object¹

clone¹, equals¹, finalize¹, getClass¹, hashCode¹, notify¹, notifyAll¹, wait¹, wait¹, wait¹

Vector2:

Package main

Class Vector2

java.lang.Object[Ⓔ]
main.Vector2

```
public class Vector2  
extends ObjectⒺ
```

This class is used to make entities be able to move

Field Summary

Fields

Modifier and Type	Field	Description
int	x	
int	y	

Constructor Summary

Constructors

Constructor	Description
Vector2 (int x, int y)	This constructor creates the vector for the entity

Method Summary

All Methods

Static Methods

Instance Methods

Concrete Methods

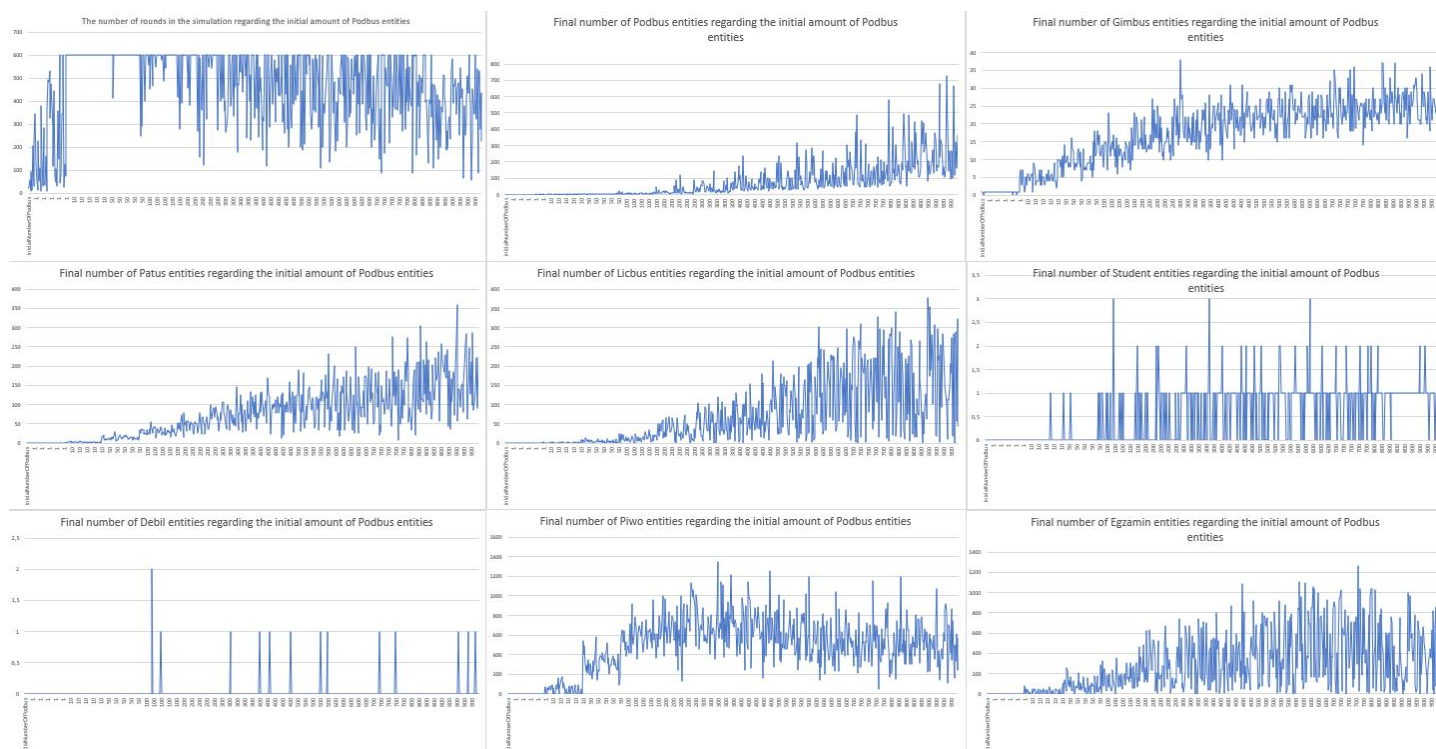
Modifier and Type	Method	Description
Vector2	AddVector (Vector2 vec)	This method returns current vector which x and y elements are enlarges with given x and y element of given vector
static Vector2	AddVectors (Vector2 first, Vector2 second)	This method returns vector which x and y elements are sum of x and y elements of two vectors
boolean	Compare (Vector2 vec)	This method returns true if current x and y values matches x and y values of given vector
double	GetLenght ()	This method Returns length of a vector
void	Invert ()	This method swaps x and y elements with each other
Vector2	MultiplyVector (int multi)	This method returns current vector multiplied by given values
static Vector2	One ()	This method returns Vector2 with x = 1 and y = 1
void	Set (int x, int y)	This method sets vector x and y values
Vector2	SubtractVector (Vector2 vec)	This method returns current vector which x and y elements are reduced with given x and y element of given vector
String [Ⓔ]	toString ()	
static Vector2	Zero ()	This method returns Vector2 with x = 0 and y = 0

Methods inherited from class java.lang.Object[Ⓔ]

clone[Ⓔ], equals[Ⓔ], finalize[Ⓔ], getClass[Ⓔ], hashCode[Ⓔ], notify[Ⓔ], notifyAll[Ⓔ], wait[Ⓔ], wait[Ⓔ], wait[Ⓔ]

- Testy jednostkowe

Wykonano także testy w programie i odnotowano różne zmiany w zależności od zmiany wartości początkowej pewnego parametru. Ustalono, że byłyby to liczba Podbusów. Wyniki testów jednostkowych, przedstawiające zależności końcowych ilości Rund oraz Entitów, odnośnie początkowej liczby Podbusów, wyglądają następująco:

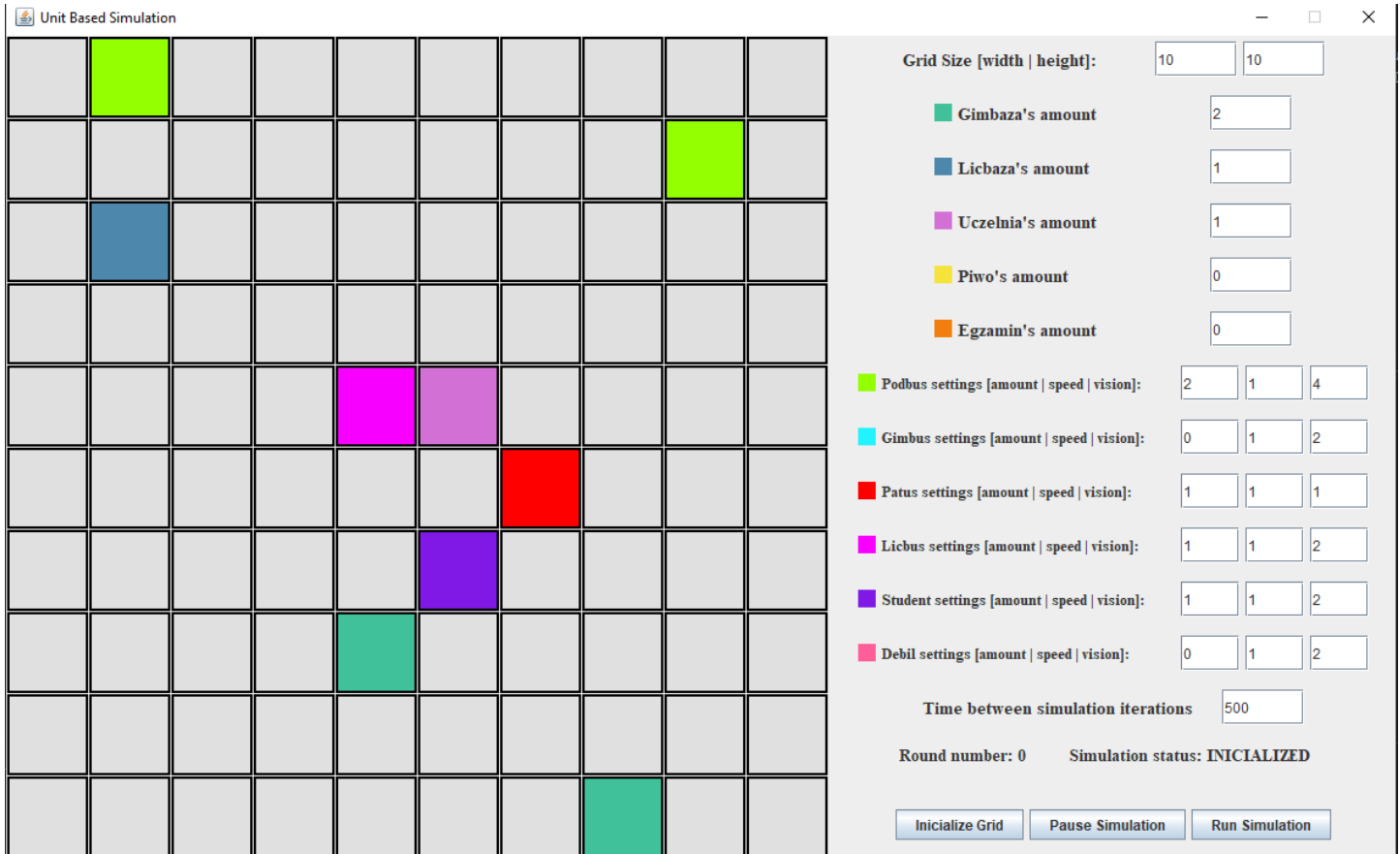


Widać bardzo duże rozrzuty wartości końcowych tych parametrów, jednak przy większości przypadków da się zauważyć pewne zależności:

1. Entity ruchome końcowe (Student i Debil) występują wręcz w przytłaczająco niskich ilościach w porównaniu do pozostałych Entitów.
2. Liczba rund na początku rośnie, a po przekroczeniu około 50 Podbusów, zaczyna stopniowo i powoli, maleć
3. Patusy, Licbusy i Egzaminy zdają się mieć coraz większy rozstrzał wyników wraz z postępem ilości początkowej Podbusów. Warto wspomnieć, że te jednostki mają pewną zależność. Licbusy upuszczają Egzaminy, które przyciągają Patusy. Być może wpływa to jakoś na taki rozrzut wyników wraz z postępującą ilością Podbusów.
4. Podczas, gdy liczba bytów ruchomych zdaje się rosnać wraz z ilością Podbusów, po czym ustabilnia się, to liczba bytów nieruchomych (Piwo i Egzamin) zdaje się rosnać do pewnej ilości Podbusów (około 300) po czym zaczyna stopniowo maleć.

• Zdjęcia aplikacji

Unit Based Simulation



Grid Size [width | height]: 10 | 10

Gimbaza's amount: 2

Licbaza's amount: 1

Uczelnia's amount: 1

Piwo's amount: 0

Egzamin's amount: 0

Podbus settings [amount | speed | vision]: 2 | 1 | 4

Gimbus settings [amount | speed | vision]: 0 | 1 | 2

Patus settings [amount | speed | vision]: 1 | 1 | 1

Licbus settings [amount | speed | vision]: 1 | 1 | 2

Student settings [amount | speed | vision]: 1 | 1 | 2

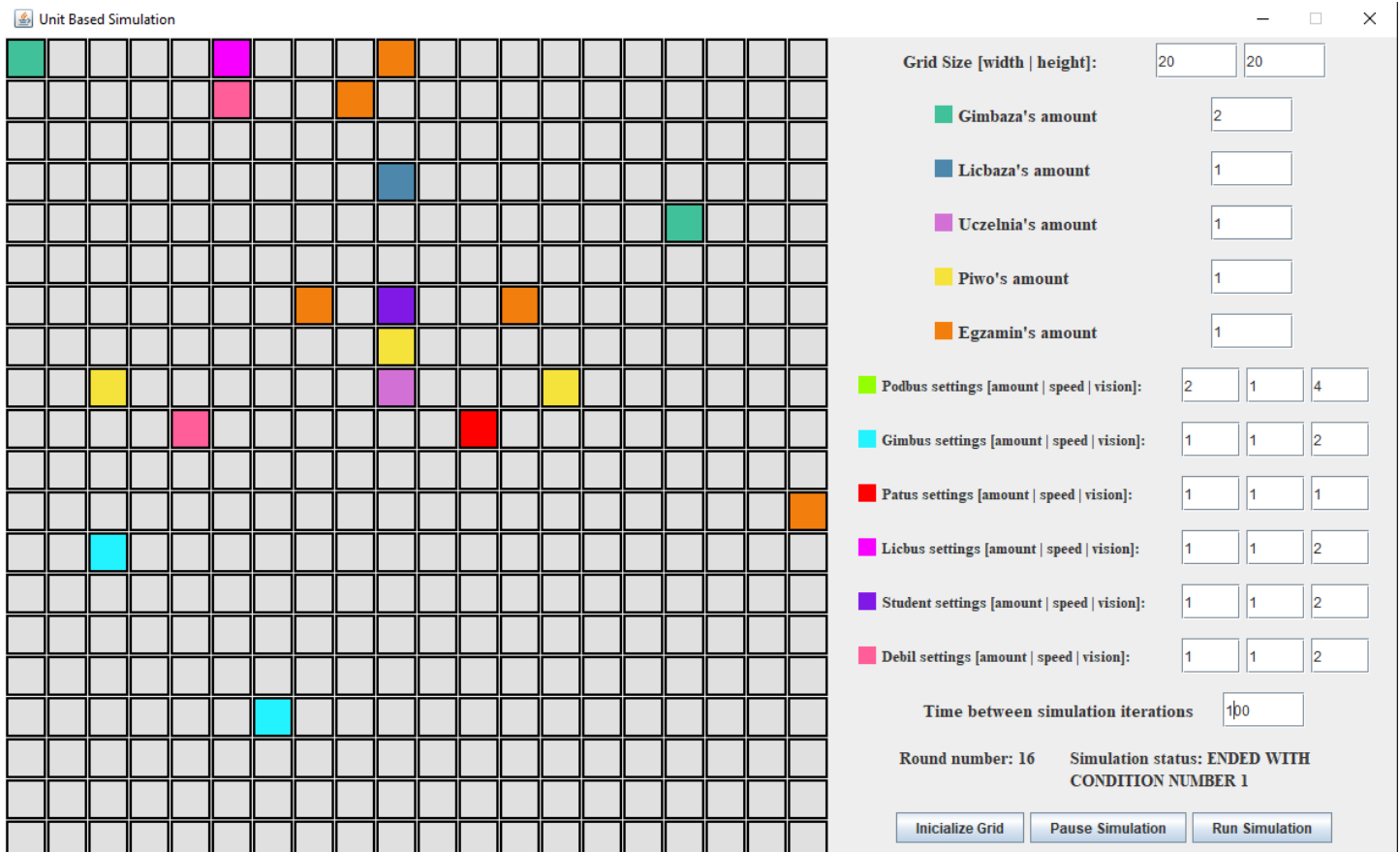
Debil settings [amount | speed | vision]: 0 | 1 | 2

Time between simulation iterations: 500

Round number: 0 Simulation status: INICIALIZED

Initialize Grid Pause Simulation Run Simulation

Unit Based Simulation



Grid Size [width | height]: 20 | 20

Gimbaza's amount: 2

Licbaza's amount: 1

Uczelnia's amount: 1

Piwo's amount: 1

Egzamin's amount: 1

Podbus settings [amount | speed | vision]: 2 | 1 | 4

Gimbus settings [amount | speed | vision]: 1 | 1 | 2

Patus settings [amount | speed | vision]: 1 | 1 | 1

Licbus settings [amount | speed | vision]: 1 | 1 | 2

Student settings [amount | speed | vision]: 1 | 1 | 2

Debil settings [amount | speed | vision]: 1 | 1 | 2

Time between simulation iterations: 100

Round number: 16 Simulation status: ENDED WITH CONDITION NUMBER 1

Initialize Grid Pause Simulation Run Simulation