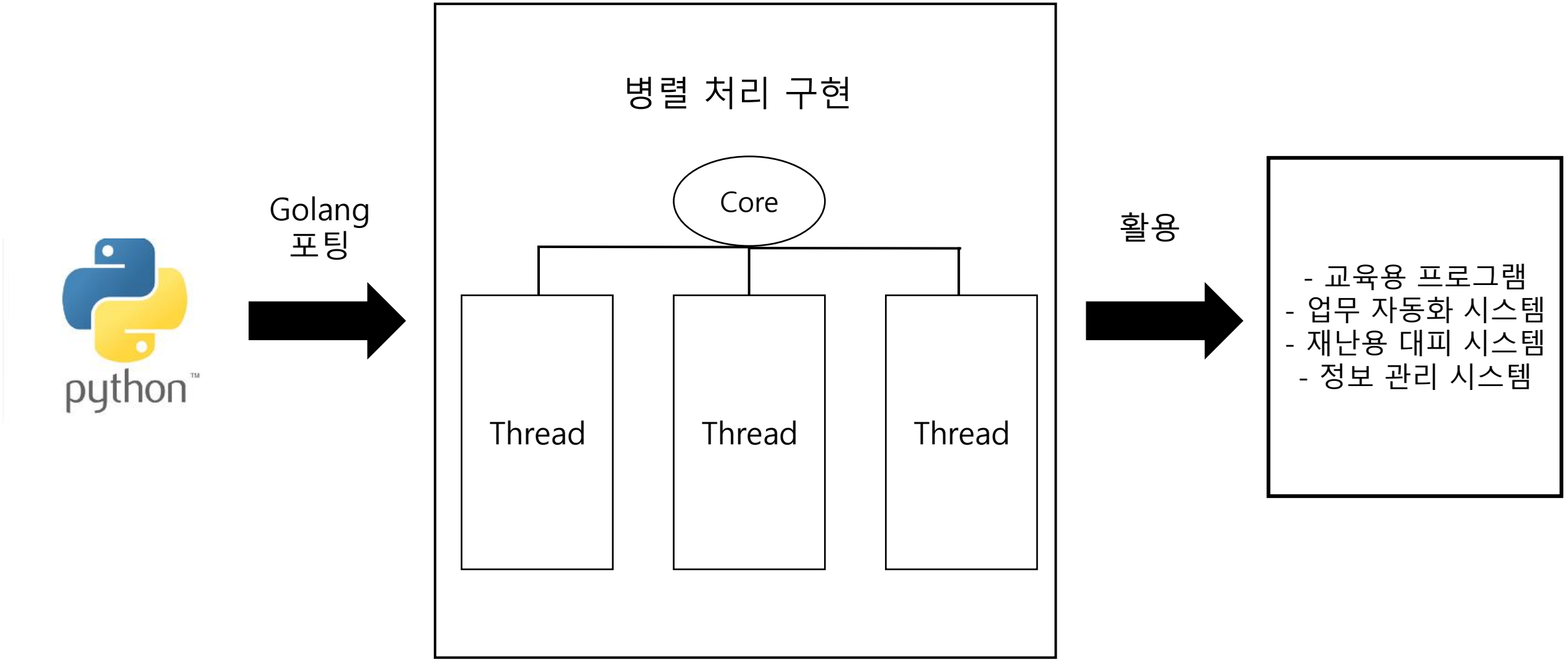
GO 기반 Simulation game engine 개발 및 활용

한밭대학교 컴퓨터공학과
김범수 도용주 이제혁

목차
INDEX

1 프로젝트 개요

프로젝트 목표



2 진행 상황

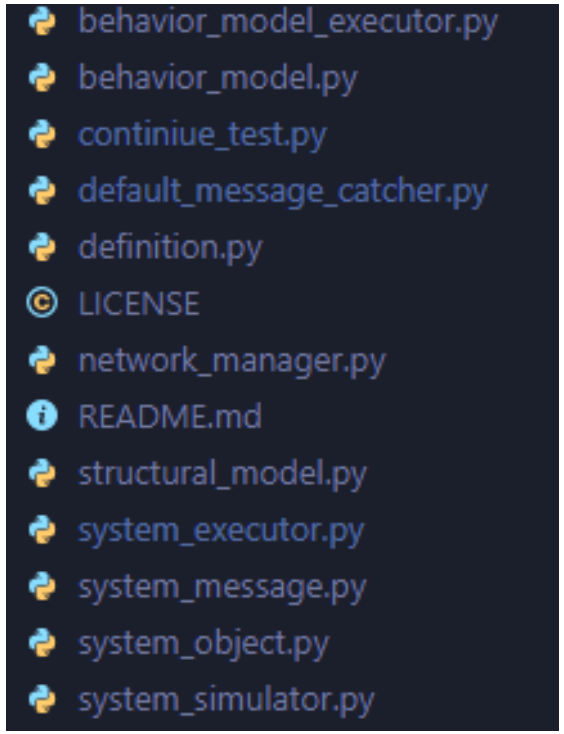
미로 게임

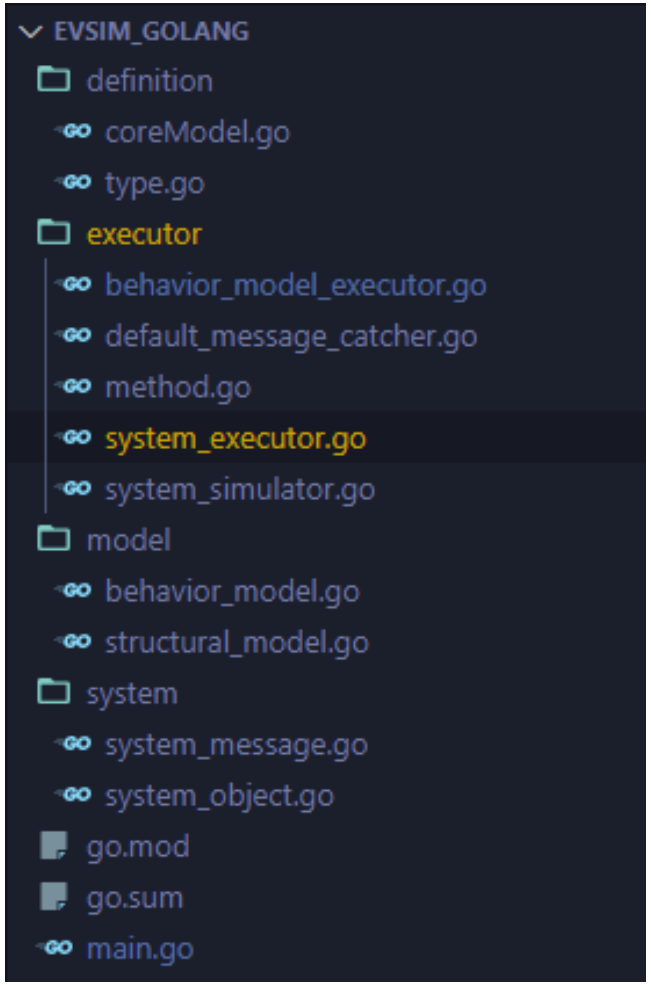
```
DD
exttrans IDLE
[agent][start]
[agent][in] cm_list :['F', 'F', 'F', 'F', 'F', 'F', 'F', 'F', 'F', 'F']
output SEND
[agent][current] : [1, 1]
[Gm][in] received
[Gm] aX:1 aY:1
int trans SEND
[Gm][out]{'R': 0, 'L': 1, 'F': 0, 'B': 1}
exttrans IDLE
[agent][in]
output MOVE
[agent] [cm] = F, [rest cmlist] = ['F', 'F', 'F', 'F', 'F', 'F', 'F', 'F', 'F']
[agent] move X:1,Y:2

int trans MOVE
output SEND
[agent][current] : [1, 2]
[Gm][in] received
[Gm] aX:1 aY:2
int trans SEND
[Gm][out]{'R': 1, 'L': 1, 'F': 0, 'B': 0}
exttrans IDLE
[agent][in]
output MOVE
[agent] [cm] = F, [rest cmlist] = ['F', 'F', 'F', 'F', 'F', 'F', 'F', 'F']
[agent] move X:1,Y:3
```

```
int trans MOVE
output SEND
[agent][current] : [1, 3]
[Gm][in] received
[Gm] aX:1 aY:3
int trans SEND
[Gm][out]{'R': 1, 'L': 1, 'F': 0, 'B': 0}
exttrans IDLE
[agent][in]
output MOVE
[agent] [cm] = F, [rest cmlist] = ['F', 'F', 'F', 'F', 'F', 'F', 'F']
[agent] move X:1,Y:4

int trans MOVE
output SEND
[agent][current] : [1, 4]
[Gm][in] received
[Gm] aX:1 aY:4
int trans SEND
[Gm][out]{'R': 1, 'L': 1, 'F': 1, 'B': 0}
exttrans IDLE
[agent][in]
output MOVE
[agent] [cm] = F, [rest cmlist] = ['F', 'F', 'F', 'F', 'F', 'F']
[agent] can't go
[agent] if move
int trans MOVE
output MOVE
[agent] [cm] = R, [rest cmlist] = ['F', 'F', 'F', 'F', 'F', 'F']
[agent] can't go
[agent] if move
int trans MOVE
output MOVE
[agent] [cm] = L, [rest cmlist] = ['F', 'F', 'F', 'F', 'F', 'F']
[agent] can't go
[agent] if move
int trans END
GAME END
```

2 진행 상황

1:1 포팅


behavior_model_executor.py
behavior_model.py
continiue_test.py
default_message_catcher.py
definition.py
LICENSE
network_manager.py
README.md
structural_model.py
system_executor.py
system_message.py
system_object.py
system_simulator.py


EVSIM_GOLANG
definition
coreModel.go
type.go
executor
behavior_model_executor.go
default_message_catcher.go
method.go
system_executor.go
system_simulator.go
model
behavior_model.go
structural_model.go
system
system_message.go
system_object.go
go.mod
go.sum
main.go

상속 문제

coreModel.go

```go
type CoreModel struct {
    _type         int
    Name          string
    Intput_ports  []string
    _output_ports []string
}

func (c *CoreModel) Set_name(name string) {
    c.Name = name
}

func (c *CoreModel) Get_name() string {
    return c.Name
}
func (c *CoreModel) Insert_input_port(port string) {
    c.Intput_ports = append(c.Intput_ports, port)
}
func (c *CoreModel) RetrieveInput_port() []string {
    return c.Intput_ports
}

func (c *CoreModel) Insert_output_port(port string) {
    c._output_ports = append(c._output_ports, port)
}

func (c *CoreModel) Retrieve_output_port() []string {
    return c._output_ports
}

func (c *CoreModel) Get_type() int {
    return c._type
}
```

behavior_model.go

```go
import (
    "evsim_golang/definition"
)

type Behaviormodel struct {
    States    map[string]float64
    CoreModel *definition.CoreModel
}

func (b *Behaviormodel) Insert_state(name string, deadline float64) { //deadline 디폴트값 = 0
    // if deadline == 0 {
    //  deadline = math.Inf(1)
    // }
    b.States[name] = deadline
}

func (b *Behaviormodel) Update_state(name string, deadline float64) { //deadline 디폴트값 = 0
    // if deadline == 0 {
    //  deadline = math.Inf(1)
    // }
    b.States[name] = deadline
}

func NewBehaviorModel(name string) *Behaviormodel {
    b := Behaviormodel{}
    b.States = make(map[string]float64)
    b.CoreModel = definition.NewCoreModel(name, definition.BEHAVIORAL)
    return &b
}
```

Abstract Method 문제

```go
type AbstractModel interface {
    Int_trans()
    Ext_trans(port string, msg *system.SysMessage)
    Output() *system.SysMessage
}

type BehaviorModelExecutor struct {
    sysobject      *system.SysObject
    Behaviormodel *model.Behaviormodel
    AbstractModel

    _cancel_reshedule_f bool
    engine_name         string
    Cur_state           string
    Instance_t          float64
    Destruct_t          float64
    Next_event_t        float64
    requestedTime       float64
}
```

```go
func NewGenerator() *Generator {
    gen := Generator{}
    gen.executor = executor.NewExecutor(0, definition.Infinite, "Gen", "sname")
    gen.executor.AbstractModel = &gen
    gen.executor.Init_state("IDLE")
    gen.executor.Behaviormodel.Insert_state("IDLE", definition.Infinite)
    gen.executor.Behaviormodel.Insert_state("SEND", 1)
    gen.executor.Behaviormodel.Insert_state("MOVE", 1)
    gen.executor.Behaviormodel.CoreModel.Insert_input_port("start")
    gen.executor.Behaviormodel.CoreModel.Insert_output_port("process")
    gen.msg_list = append(gen.msg_list, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
    return &gen
}
```

디버깅 및 Continue_test 동작

```go
func main() {
    se := executor.NewSysSimulator()
    se.Register_engine("sname", "REAL_TIME", 1)
    sim := se.Get_engine("sname")
    sim.Behaviormodel.CoreModel.Insert_input_port("start")

    gen := NewGenerator()
    pro := NewProcessor()

    sim.Register_entity(gen.executor)

    sim.Register_entity(pro.executor)

    sim.Coupling_relation(nil, "start", gen.executor, "start")
    sim.Coupling_relation(gen.executor, "process", pro.executor, "process")
    sim.Insert_external_event("start", nil, 0)
    sim.Simulate(definition.Infinite)

}
```

```
PS C:\Users\J.H.Lee\OneDrive - 한밭대학교\공부하자\캡스톤 디자인\1학기\evsim_golang> go run main.go
[gen][in]: 2022-05-03 21:15:32.4022158 +0900 KST m=+0.002131201
[gen][out]: 2022-05-03 21:15:32.4229981 +0900 KST m=+0.022913501
[proc][in] 2022-05-03 21:15:32.4235156 +0900 KST m=+0.023431001
[proc][out] 2022-05-03 21:15:32.4235156 +0900 KST m=+0.023431001
1
[gen][out]: 2022-05-03 21:15:32.4235156 +0900 KST m=+0.023431001
[proc][in] 2022-05-03 21:15:32.4240428 +0900 KST m=+0.023958201
[proc][out] 2022-05-03 21:15:32.4240428 +0900 KST m=+0.023958201
1 2
[gen][out]: 2022-05-03 21:15:32.4240428 +0900 KST m=+0.023958201
[proc][in] 2022-05-03 21:15:32.4245742 +0900 KST m=+0.024489601
[proc][out] 2022-05-03 21:15:32.4245742 +0900 KST m=+0.024489601
1 2 3
[gen][out]: 2022-05-03 21:15:32.4245742 +0900 KST m=+0.024489601
[proc][in] 2022-05-03 21:15:32.4245742 +0900 KST m=+0.024489601
[proc][out] 2022-05-03 21:15:32.4251049 +0900 KST m=+0.025020301
1 2 3 4
[gen][out]: 2022-05-03 21:15:32.4251049 +0900 KST m=+0.025020301
[proc][in] 2022-05-03 21:15:32.4251049 +0900 KST m=+0.025020301
[proc][out] 2022-05-03 21:15:32.4256439 +0900 KST m=+0.025559301
1 2 3 4 5
[gen][out]: 2022-05-03 21:15:32.4256439 +0900 KST m=+0.025559301
[proc][in] 2022-05-03 21:15:32.4256439 +0900 KST m=+0.025559301
[proc][out] 2022-05-03 21:15:32.4261625 +0900 KST m=+0.026077901
1 2 3 4 5 6
[gen][out]: 2022-05-03 21:15:32.4261783 +0900 KST m=+0.026093701
[proc][in] 2022-05-03 21:15:32.4261783 +0900 KST m=+0.026093701
[proc][out] 2022-05-03 21:15:32.4261783 +0900 KST m=+0.026093701
1 2 3 4 5 6 7
[gen][out]: 2022-05-03 21:15:32.4266981 +0900 KST m=+0.026613501
[proc][in] 2022-05-03 21:15:32.4266981 +0900 KST m=+0.026613501
[proc][out] 2022-05-03 21:15:32.4266981 +0900 KST m=+0.026613501
1 2 3 4 5 6 7 8
[gen][out]: 2022-05-03 21:15:32.4272289 +0900 KST m=+0.027144301
[proc][in] 2022-05-03 21:15:32.4272289 +0900 KST m=+0.027144301
[proc][out] 2022-05-03 21:15:32.4272289 +0900 KST m=+0.027144301
1 2 3 4 5 6 7 8 9
[gen][out]: 2022-05-03 21:15:32.427756 +0900 KST m=+0.027671401
[proc][in] 2022-05-03 21:15:32.427756 +0900 KST m=+0.027671401
[proc][out] 2022-05-03 21:15:32.427756 +0900 KST m=+0.027671401
1 2 3 4 5 6 7 8 9 10
[gen][out]: 2022-05-03 21:15:32.4282911 +0900 KST m=+0.028206501
```

3 향후 계획

향후 진행 계획

1. Golang 엔진 이 실제 시간(REAL_TIME)으로 동작하게 끔 디버깅

2. 1대1 포팅 된 코드를 Golang의 특성에 맞게끔 코드 리팩토링

3. Go 루틴을 이용하여 병렬처리 알고리즘을 구현

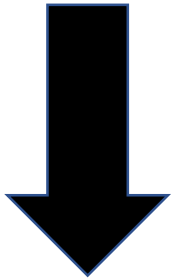
4. 기존 엔진과 포팅된 엔진의 성능 비교

3 향후 계획

향후 진행 계획

```
[Gen][IN]: Gen0 : 2022-05-03 22:46:14.509314
[Gen][IN]: Gen1 : 2022-05-03 22:46:14.509389
[Gen][IN]: Gen2 : 2022-05-03 22:46:14.509389
[Gen][IN]: Gen3 : 2022-05-03 22:46:14.509389
[Gen][IN]: Gen4 : 2022-05-03 22:46:14.509389
[Gen][IN]: Gen5 : 2022-05-03 22:46:14.509389
[Gen][IN]: Gen6 : 2022-05-03 22:46:14.509389
[Gen][OUT]: Gen0 : 2022-05-03 22:46:15.514642
Global time : 1
[Gen][OUT]: Gen1 : 2022-05-03 22:46:16.529338
Global time : 2
[Gen][OUT]: Gen2 : 2022-05-03 22:46:17.535380
Global time : 3
[Gen][OUT]: Gen3 : 2022-05-03 22:46:18.541822
Global time : 4
[Gen][OUT]: Gen4 : 2022-05-03 22:46:19.550110
Global time : 5
[Gen][OUT]: Gen5 : 2022-05-03 22:46:20.556048
Global time : 6
[Gen][OUT]: Gen6 : 2022-05-03 22:46:21.563164
Global time : 7
```

각 오브젝트 동작까지의 미세한 시간오차



Golang의 병렬 처리

미세 한 수준의 오차를 더 개선

Q & A

감사합니다